SIEMENS

SIMATIC

S7
Controlador programable S7-1200

Manual de sistema

Prólogo

Sinopsis del producto 1

Nuevas funciones 2

Software de programación STEP 7 3

Montaje 4

Principios básicos del PLC 5

Configuración de dispositivos 6

Principios básicos de programación 7

Instrucciones básicas 8

Instrucciones avanzadas 9

Instrucciones tecnológicas 10

Comunicación 11

Servidor web 12

Procesador de comunicaciones y Modbus TCP 13

Comunicación TeleService (correo electrónico SMTP) 14

Herramientas online y diagnóstico 15

Datos técnicos A

Calcular la corriente necesaria B

Información de pedido C

Cambio de dispositivo y compatibilidad de los repuestos D

V4.5 05/2021
A5E02486683-AO

Notas jurídicas

Filosofía en la señalización de advertencias y peligros

Este manual contiene las informaciones necesarias para la seguridad personal así como para la prevención de daños materiales. Las informaciones para su seguridad personal están resaltadas con un triángulo de advertencia; las informaciones para evitar únicamente daños materiales no llevan dicho triángulo. De acuerdo al grado de peligro las consignas se representan, de mayor a menor peligro, como sigue.

⚠ **PELIGRO**
Significa que si no se adoptan las medidas preventivas adecuadas **se producirá** la muerte o bien lesiones corporales graves.

⚠ **ADVERTENCIA**
Significa que si no se adoptan las medidas preventivas adecuadas **puede producirse** la muerte o bien lesiones corporales graves.

⚠ **PRECAUCIÓN**
Significa que si no se adoptan las medidas preventivas adecuadas pueden producirse lesiones corporales.

ATENCIÓN
Significa que si no se adoptan las medidas preventivas adecuadas pueden producirse daños materiales.

Si se dan varios niveles de peligro se usa siempre la consigna de seguridad más estricta en cada caso. Si en una consigna de seguridad con triángulo de advertencia de alarma de posibles daños personales, la misma consigna puede contener también una advertencia sobre posibles daños materiales.

Personal cualificado

El producto/sistema tratado en esta documentación sólo deberá ser manejado o manipulado por **personal cualificado** para la tarea encomendada y observando lo indicado en la documentación correspondiente a la misma, particularmente las consignas de seguridad y advertencias en ella incluidas. Debido a su formación y experiencia, el personal cualificado está en condiciones de reconocer riesgos resultantes del manejo o manipulación de dichos productos/sistemas y de evitar posibles peligros.

Uso previsto de los productos de Siemens

Considere lo siguiente:

⚠ **ADVERTENCIA**
Los productos de Siemens sólo deberán usarse para los casos de aplicación previstos en el catálogo y la documentación técnica asociada. De usarse productos y componentes de terceros, éstos deberán haber sido recomendados u homologados por Siemens. El funcionamiento correcto y seguro de los productos exige que su transporte, almacenamiento, instalación, montaje, manejo y mantenimiento hayan sido realizados de forma correcta. Es preciso respetar las condiciones ambientales permitidas. También deberán seguirse las indicaciones y advertencias que figuran en la documentación asociada.

Marcas registradas

Todos los nombres marcados con ® son marcas registradas de Siemens AG. Los restantes nombres y designaciones contenidos en el presente documento pueden ser marcas registradas cuya utilización por terceros para sus propios fines puede violar los derechos de sus titulares.

Exención de responsabilidad

Hemos comprobado la concordancia del contenido de esta publicación con el hardware y el software descritos. Sin embargo, como es imposible excluir desviaciones, no podemos hacernos responsable de la plena concordancia. El contenido de esta publicación se revisa periódicamente; si es necesario, las posibles correcciones se incluyen en la siguiente edición.

Siemens AG
Digital Industries
Postfach 48 48
90026 NÜRNBERG
ALEMANIA

A5E02486683-AO
Ⓟ 04/2021 Sujeto a cambios sin previo aviso

Copyright © Siemens AG 2021.
Reservados todos los derechos

Prólogo

Objeto del manual

La gama S7-1200 abarca distintos controladores lógicos programables (PLC) que pueden utilizarse para numerosas tareas. Gracias a su diseño compacto, bajo costo y amplio juego de instrucciones, los PLC S7-1200 son idóneos para controlar una gran variedad de aplicaciones. Los modelos S7-1200 y el software de programación STEP 7 (Página 37) basado en Windows ofrecen la flexibilidad necesaria para solucionar las tareas de automatización.

Este manual contiene información sobre cómo montar y programar los PLC S7-1200 y está dirigido a ingenieros, programadores, técnicos de instalación y electricistas que dispongan de conocimientos básicos sobre los controladores lógicos programables.

Nociones básicas

Para comprender este manual se requieren conocimientos básicos en el campo de la automatización y de los controladores lógicos programables.

Alcance del manual

Este manual describe los productos siguientes:

- STEP 7 Basic y Professional (Página 37)
- CPU S7-1200 con versión de firmware V4.5

En los datos técnicos (Página 1245) encontrará una lista completa de los productos S7-1200 descritos en el manual.

Certificaciones, marcado CE, C-Tick y otras homologaciones

Para más información, consulte los datos técnicos (Página 1245).

Service & Support

Además de la documentación, Siemens ofrece conocimientos técnicos en Internet, en la página web del Customer Support (https://support.industry.siemens.com/cs/ww/es/).

Contacte con el representante de Siemens más próximo si tiene consultas de carácter técnico, así como para obtener información sobre los cursillos de formación o para pedir productos S7. Puesto que los representantes de Siemens han sido debidamente aleccionados y tienen conocimientos detallados sobre las operaciones, los procesos y la industria, así como sobre los distintos productos de Siemens empleados, pueden solucionar cualquier problema de forma rápida y eficiente.

Documentación e información

S7-1200 y STEP 7 ofrecen una documentación variada y otros recursos para encontrar la información técnica requerida.

- El manual de sistema del controlador programable S7-1200 ofrece información específica sobre el funcionamiento, la programación y las especificaciones de toda la familia de productos S7-1200.
  El manual de sistema está disponible en formato electrónico (PDF). Este y otros manuales electrónicos pueden descargarse o visualizarse en la página web de Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/). El manual de sistema también está disponible en el disco de documentos que se suministra con toda CPU S7-1200.
- El sistema de información online de STEP 7 ofrece un acceso inmediato a la información conceptual y a instrucciones específicas que describen el funcionamiento y las funciones del paquete de programación, así como el funcionamiento básico de las CPU SIMATIC.
- La página web de Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/) proporciona acceso a las versiones en formato electrónico (PDF) del conjunto de documentos de SIMATIC, incluidos el manual de sistema y el sistema de información de STEP 7. Los documentos existentes están disponibles en el enlace Product Support. Este acceso a la documentación online permite arrastrar y soltar temas de varios documentos para crear un manual propio personalizado. Las actualizaciones de manuales de sistema publicados anteriormente también están disponibles en Siemens Industry Online Support.
  Para acceder a la documentación online, haga clic en "mySupport" en la parte izquierda de la página y seleccione "Documentación" en las opciones de navegación. Para emplear las funciones de la documentación mySupport debe iniciar sesión como usuario registrado.
- La página web de Siemens Industry Online Support también cuenta con preguntas frecuentes y otros documentos que pueden ser de ayuda para S7-1200 y STEP 7.
- También puede seguir discusiones sobre productos y participar en ellas a través del foro técnico de Service & Support (https://support.industry.siemens.com/tf/ww/en/?Language=en&siteid=csius&treeLang=en&groupid=4000002&extranet=standard&viewreg=WW&nodeid0=34612486). Los foros permiten interactuar con diferentes expertos sobre productos.
  - Foro de S7-1200 (https://support.industry.siemens.com/tf/ww/en/threads/237?title=simatic-s7-1200&skip=0&take=10&orderBy=LastPostDate+desc)
  - Foro de 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)

Información de seguridad

Siemens ofrece productos y soluciones con funciones de seguridad industrial con el objetivo de hacer más seguro el funcionamiento de instalaciones, sistemas, máquinas y redes.

Para proteger las instalaciones, los sistemas, las máquinas y las redes contra de amenazas cibernéticas, es necesario implementar (y mantener continuamente) un concepto de seguridad industrial integral que esté conforme al estado del arte. Los productos y las soluciones de Siemens constituyen una parte de este concepto.

Los clientes son responsables de impedir el acceso no autorizado a sus instalaciones, sistemas, máquinas y redes. Dichos sistemas, máquinas y componentes solo deben estar conectados a la

red corporativa o a Internet cuando y en la medida que sea necesario y siempre que se hayan tomado las medidas de protección adecuadas (p. ej. cortafuegos y segmentación de la red).

Para obtener información adicional sobre las medidas de seguridad industrial que podrían ser implementadas, por favor visite (https://www.siemens.com/industrialsecurity).

Los productos y las soluciones de Siemens están sometidos a un desarrollo constante con el fin de hacerlos más seguros. Siemens recomienda expresamente realizar actualizaciones en cuanto estén disponibles y utilizar únicamente las últimas versiones de los productos. El uso de versiones de los productos anteriores o que ya no sean soportadas y la falta de aplicación de las nuevas actualizaciones, puede aumentar el riesgo de amenazas cibernéticas.

Para mantenerse informado de las actualizaciones de productos, recomendamos que se suscriba al Siemens Industrial Security RSS Feed en (https://www.siemens.com/industrialsecurity).

Mantener la seguridad operacional de la planta

Mantener la seguridad operacional de la planta

Nota

Nota importante para mantener la seguridad operacional de la planta

Las plantas con funciones relacionadas con la seguridad están sujetas a requisitos especiales de seguridad operacional por parte del operador. Incluso los proveedores están obligados a observar medidas especiales durante la monitorización del producto. Por esta razón, le informamos mediante notificaciones especiales de funciones y desarrollos de producto que son (o podrían ser) relevantes para el funcionamiento de sistemas desde el punto de vista de la seguridad.

Suscribiéndose a las notificaciones adecuadas podrá estar seguro de estar siempre al día y ser capaz de hacer cambios en su sistema, en caso necesario.

Visite el Industry Online Support. Siga los siguientes enlaces y, a su lado, haga clic con el botón derecho del ratón en "Correo electrónico con actualización":

- SIMATIC S7-300/S7-300F (https://support.industry.siemens.com/cs/ww/es/ps/13751)
- SIMATIC S7-400/S7-400H/S7-400F/FH (https://support.industry.siemens.com/cs/ww/es/ps/13828)
- SIMATIC WinAC RTX (F) (https://support.industry.siemens.com/cs/ww/es/ps/13915)
- SIMATIC S7-1500/SIMATIC S7-1500F (https://support.industry.siemens.com/cs/ww/es/ps/13716)
- SIMATIC S7-1200/SIMATIC S7-1200F (https://support.industry.siemens.com/cs/ww/es/ps/13683)
- Periferia descentralizada (https://support.industry.siemens.com/cs/ww/es/ps/14029)
- STEP 7 TIA Portal (https://support.industry.siemens.com/cs/ww/es/ps/14667)

Índice

Prólogo ... **3**

1 Sinopsis del producto ... **27**

1.1 Introducción al PLC S7-1200 ... 27

1.2 Capacidad de expansión de la CPU ... 31

1.3 Paneles HMI Basic ... 32

2 Nuevas funciones ... **35**

3 Software de programación STEP 7 ... **37**

3.1 Requisitos del sistema ... 37

3.2 Diferentes vistas que facilitan el trabajo ... 39

3.3 Compatibilidad entre STEP 7 y S7-1200 ... 40

4 Montaje ... **43**

4.1 Directrices para montar dispositivos S7-1200 ... 43

4.2 Corriente necesaria ... 44

4.3 Procedimientos de montaje y desmontaje ... 46
4.3.1 Dimensiones de montaje de los dispositivos S7-1200 ... 46
4.3.2 Montaje y desmontaje de la CPU ... 49
4.3.3 Montaje y desmontaje de SB, CB o BB ... 52
4.3.4 Instalación y desmontaje de un SM ... 54
4.3.5 Montaje y desmontaje de un CM o CP ... 55
4.3.6 Extraer y reinsertar el conector del bloque de terminales del S7-1200 ... 56
4.3.7 Instalación y desmontaje de un cable de ampliación ... 57

4.4 Directrices de cableado ... 59

5 Principios básicos del PLC ... **67**

5.1 Ejecución del programa de usuario ... 67
5.1.1 Estados operativos de la CPU ... 71
5.1.2 Procesamiento del ciclo en estado operativo RUN ... 74
5.1.3 Bloques de organización (OB) ... 75
5.1.3.1 OB de ciclo ... 75
5.1.3.2 OB de arranque ... 76
5.1.3.3 OB de alarma de retardo ... 76
5.1.3.4 OB de alarma cíclica ... 77
5.1.3.5 OB de alarma de proceso ... 78
5.1.3.6 OB de alarma de error de tiempo ... 79
5.1.3.7 OB de alarma de error de diagnóstico ... 80
5.1.3.8 OB de presencia de módulo ... 82
5.1.3.9 OB de fallo del rack o estación ... 83
5.1.3.10 OB de hora ... 83
5.1.3.11 OB de estado ... 84

5.1.3.12 OB de actualización ... 84
5.1.3.13 OB de perfil ... 85
5.1.3.14 OB MC-Servo y MC-Interpolator ... 85
5.1.3.15 MC-PreServo ... 85
5.1.3.16 MC-PostServo ... 86
5.1.3.17 Prioridades y colas de espera para la ejecución de eventos ... 87
5.1.4 Vigilancia y configuración del tiempo de ciclo ... 91
5.1.5 Memoria de la CPU ... 93
5.1.5.1 Marcas de sistema y de ciclo ... 95
5.1.6 Búfer de diagnóstico ... 97
5.1.7 Reloj en tiempo real ... 98
5.1.8 Configurar las salidas en una transición de RUN a STOP ... 98

5.2 Almacenamiento de datos, áreas de memoria, E/S y direccionamiento ... 99
5.2.1 Acceder a los datos del S7-1200 ... 99

5.3 Procesamiento de valores analógicos ... 105

5.4 Tipos de datos ... 106
5.4.1 Tipos de datos Bool, Byte, Word y DWord ... 107
5.4.2 Tipos de datos de entero ... 108
5.4.3 Tipos de datos de real en coma flotante ... 109
5.4.4 Tipos de datos de fecha y hora ... 109
5.4.5 Tipos de datos Carácter y Cadena ... 112
5.4.6 Tipo de datos ARRAY ... 114
5.4.7 Tipo de datos Estructura de datos ... 115
5.4.8 Tipo de datos PLC ... 115
5.4.9 Tipo de datos de puntero Variant ... 116
5.4.10 Acceder a un "segmento" de un tipo de datos de variable ... 116
5.4.11 Acceso a una variable con una superposición de AT ... 117

5.5 Utilizar una Memory Card ... 119
5.5.1 Insertar una Memory Card en la CPU ... 120
5.5.2 Configurar los parámetros de arranque de la CPU antes de copiar el proyecto en la Memory Card ... 122
5.5.3 Utilizar la Memory Card como tarjeta de "Transferencia" ... 123
5.5.4 Utilizar la Memory Card como tarjeta de "Programa" ... 125
5.5.5 Uso de una SIMATIC Memory Card para proteger datos de configuración confidenciales del PLC ... 130
5.5.6 Actualización de firmware ... 132

5.6 Recuperación si se olvida la contraseña ... 135

6 Configuración de dispositivos ... 137

6.1 Insertar una CPU ... 138

6.2 Cargar la configuración de una CPU conectada ... 139

6.3 Agregar módulos a la configuración ... 141

6.4 Control de configuración ... 142
6.4.1 Ventajas y aplicaciones del control de configuración ... 142
6.4.2 Configuración de la instalación central y módulos opcionales ... 143
6.4.3 Ejemplo de control de configuración ... 149

6.5 Cambiar un dispositivo ... 152

6.6 Configurar el funcionamiento de la CPU ... 153
6.6.1 Vista general ... 153
6.6.2 Configuración de tiempos de filtro de entradas digitales ... 155
6.6.3 Capturar impulsos ... 156
6.7 Configurar el soporte multilingüe ... 158
6.8 Protección y seguridad ... 160
6.8.1 Uso del asistente de seguridad para establecer los ajustes de seguridad del PLC ... 160
6.8.2 Protección de datos de configuración confidenciales del PLC ... 161
6.8.3 Protección por nivel de acceso para la CPU ... 163
6.8.4 Configurar mecanismos de conexión ... 165
6.8.4.1 Configurar el mecanismo de acceso para interlocutores remotos ... 165
6.8.4.2 Activar la comunicación PG/PC y HMI segura y crear certificados ... 166
6.8.5 Memoria de carga externa ... 167
6.8.6 Protección de know-how ... 167
6.8.7 Protección anticopia ... 168
6.9 Configurar los parámetros de los módulos ... 169
6.10 Configurar la CPU para la comunicación ... 171
6.11 Sincronización horaria ... 172

7 Principios básicos de programación ... 175

7.1 Directrices para diseñar un sistema PLC ... 175
7.2 Estructurar el programa de usuario ... 176
7.3 Utilizar bloques para estructurar el programa ... 177
7.3.1 Bloque de organización (OB) ... 178
7.3.2 Función (FC) ... 180
7.3.3 Bloque de función (FB) ... 181
7.3.4 Bloque de datos (DB) ... 182
7.3.5 Crear bloques lógicos reutilizables ... 184
7.3.6 Transferir parámetros a bloques ... 184
7.4 Principios básicos de la coherencia de datos ... 187
7.5 Lenguaje de programación ... 188
7.5.1 Esquema de contactos (KOP) ... 188
7.5.2 Diagrama de funciones (FUP) ... 189
7.5.3 SCL ... 189
7.5.3.1 Editor de programas SCL ... 190
7.5.3.2 Expresiones y operaciones de SCL ... 191
7.5.3.3 Direccionamiento indexado con instrucciones PEEK y POKE ... 195
7.5.4 EN y ENO para KOP, FUP y SCL ... 197
7.6 Cargar los elementos del programa en la CPU ... 199
7.7 Sincronizar la CPU online y el proyecto offline ... 202
7.8 Cargar desde la CPU online ... 204
7.8.1 Comparar la CPU online con la CPU offline ... 204
7.9 Depurar y comprobar el programa ... 204
7.9.1 Vigilar y modificar datos de la CPU ... 204
7.9.2 Tablas de observación y tablas de forzado ... 205

7.9.3 Referencia cruzada para mostrar la utilización ... 206
7.9.4 Estructura de llamadas para ver la jerarquía de llamadas ... 207

8 Instrucciones básicas ... 209

8.1 Operaciones lógicas con bits ... 209
8.1.1 Operaciones lógicas con bits ... 209
8.1.2 Instrucciones "Activar salida" y "Desactivar salida" ... 212
8.1.3 Instrucciones de flanco ascendente y descendente ... 214

8.2 Temporizadores ... 217

8.3 Contadores ... 226

8.4 Comparación ... 231
8.4.1 Instrucciones de comparación ... 231
8.4.2 IN_Range (valor dentro del rango) y OUT_Range (valor fuera del rango) ... 232
8.4.3 OK (comprobar validez) y NOT_OK (comprobar invalidez) ... 233
8.4.4 Instrucciones de comparación Variant y Array ... 233
8.4.4.1 Instrucciones de comparación de igualdad y desigualdad ... 233
8.4.4.2 Instrucciones de comparación con cero ... 234
8.4.4.3 IS_ARRAY (Consultar si es un ARRAY) ... 235

8.5 Funciones matemáticas ... 236
8.5.1 CALCULATE (calcular) ... 236
8.5.2 Instrucciones "Sumar", "Restar", "Multiplicar" y "Dividir" ... 237
8.5.3 MOD (obtener resto de división) ... 238
8.5.4 NEG (generar complemento a dos) ... 239
8.5.5 INC (incrementar) y DEC (decrementar) ... 239
8.5.6 ABS (calcular valor absoluto) ... 240
8.5.7 MIN (determinar mínimo) y MAX (determinar máximo) ... 241
8.5.8 LIMIT (ajustar valor límite) ... 242
8.5.9 Instrucciones para exponentes, logaritmos y trigonometría ... 242

8.6 Transferencia ... 244
8.6.1 MOVE (copiar valor), MOVE_BLK (copiar área), UMOVE_BLK (copiar área sin interrupciones) y MOVE_BLK_VARIANT (copiar área) ... 244
8.6.2 Deserializar ... 248
8.6.3 Serialize ... 250
8.6.4 FILL_BLK (rellenar área) y UFILL_BLK (rellenar área sin interrupciones) ... 253
8.6.5 SWAP (cambiar disposición) ... 254
8.6.6 LOWER_BOUND: (leer límite inferior del ARRAY) ... 255
8.6.7 UPPER_BOUND: (leer límite superior de ARRAY) ... 256
8.6.8 Instrucciones de leer / escribir memoria ... 258
8.6.8.1 PEEK y POKE (solo SCL) ... 258
8.6.8.2 Instrucciones Leer y escribir en formato Big y Little Endian (SCL) ... 259
8.6.9 Instrucciones Variant ... 261
8.6.9.1 VariantGet (Leer valor de una variable VARIANT) ... 261
8.6.9.2 VariantPut (escribir valor en una variable VARIANT) ... 262
8.6.9.3 CountOfElements (consultar número de elementos de ARRAY) ... 263
8.6.10 Instrucciones Legacy ... 264
8.6.10.1 Instrucciones FieldRead (Leer campo) y FieldWrite (Escribir campo) ... 264
8.6.11 SCATTER ... 266
8.6.12 SCATTER_BLK ... 270
8.6.13 GATHER ... 275
8.6.14 GATHER_BLK ... 279

8.7 Conversión ... 284
8.7.1 CONV (convertir valor) ... 284
8.7.2 Instrucciones de conversión de SCL ... 285
8.7.3 ROUND (redondear número) y TRUNC (truncar a entero) ... 288
8.7.4 CEIL y FLOOR (redondear un número en coma flotante al siguiente entero superior o inferior) ... 289
8.7.5 SCALE_X (escalar) y NORM_X (normalizar) ... 290
8.7.6 Instrucciones de conversión Variant ... 293
8.7.6.1 VARIANT_TO_DB_ANY (convertir VARIANT en DB_ANY) ... 293
8.7.6.2 DB_ANY_TO_VARIANT (convertir DB_ANY en VARIANT) ... 294

8.8 Control del programa ... 295
8.8.1 Instrucciones JMP (Saltar si RLO = 1), JMPN (Saltar si RLO = 0) y LABEL (Etiqueta) ... 295
8.8.2 JMP_LIST (definir lista de saltos) ... 296
8.8.3 SWITCH (distribuidor de saltos) ... 297
8.8.4 RET (retroceder) ... 299
8.8.5 ENDIS_PW (limitar y habilitar legitimación de la contraseña) ... 300
8.8.6 RE_TRIGR (reiniciar tiempo de vigilancia del ciclo) ... 303
8.8.7 STP (finalizar programa) ... 304
8.8.8 Instrucciones GET_ERROR y GET_ERROR_ID (consultar error e ID de error localmente) ... 304
8.8.9 RUNTIME (medir tiempo de ejecución) ... 307
8.8.10 Instrucciones de control del programa de SCL ... 309
8.8.10.1 Vista general de las instrucciones de control del programa de SCL ... 309
8.8.10.2 Instrucción IF-THEN ... 310
8.8.10.3 Instrucción CASE ... 311
8.8.10.4 Instrucción FOR ... 312
8.8.10.5 Instrucción WHILE-DO ... 313
8.8.10.6 Instrucción REPEAT-UNTIL ... 314
8.8.10.7 Instrucción CONTINUE ... 315
8.8.10.8 Instrucción EXIT ... 315
8.8.10.9 Instrucción GOTO ... 316
8.8.10.10 Instrucción RETURN ... 316

8.9 Operaciones lógicas con palabras ... 317
8.9.1 Instrucciones de operaciones lógicas AND, OR y XOR ... 317
8.9.2 INV (complemento a 1) ... 318
8.9.3 Instrucciones DECO (Descodificar) y ENCO (Codificar) ... 318
8.9.4 Instrucciones SEL (Seleccionar), MUX (Multiplexar) y DEMUX (Desmultiplexar) ... 319

8.10 Desplazamiento y rotación ... 322
8.10.1 Instrucciones SHR (Desplazar a la derecha) y SHL (Desplazar a la izquierda) ... 322
8.10.2 Instrucciones ROR (Rotar a la derecha) y ROL (Rotar a la izquierda) ... 323

9 Instrucciones avanzadas ... 325

9.1 Funciones de fecha, hora y reloj ... 325
9.1.1 Instrucciones de fecha y hora ... 325
9.1.2 Funciones de reloj ... 328
9.1.3 Estructura de datos TimeTransformationRule ... 330
9.1.4 SET_TIMEZONE (ajustar zona horaria) ... 331
9.1.5 RTM (contador de horas de funcionamiento) ... 332

9.2 Cadena y carácter ... 334
9.2.1 Sinopsis del tipo de datos String ... 334
9.2.2 S_MOVE (desplazar cadena de caracteres) ... 335

9.2.3 Instrucciones de conversión de cadenas ... 335
9.2.3.1 Instrucciones S_CONV, STRG_VAL y VAL_STRG (Convertir a/de cadena de caracteres y número) ... 335
9.2.3.2 Instrucciones Strg_TO_Chars y Chars_TO_Strg (Convertir a/de cadena de caracteres y Array of CHAR) ... 344
9.2.3.3 Instrucciones ATH y HTA (Convertir a/de cadena de caracteres ASCII y número hexadecimal) ... 346
9.2.4 Instrucciones con cadenas ... 348
9.2.4.1 MAX_LEN (longitud máxima de una cadena de caracteres) ... 349
9.2.4.2 LEN (determinar la longitud de una cadena de caracteres) ... 349
9.2.4.3 CONCAT (agrupar cadenas de caracteres) ... 350
9.2.4.4 Instrucciones LEFT, RIGHT y MID (Leer los caracteres izquierdos, derechos o centrales de una cadena) ... 351
9.2.4.5 DELETE (borrar caracteres de una cadena) ... 352
9.2.4.6 INSERT (insertar caracteres en una cadena) ... 353
9.2.4.7 REPLACE (reemplazar caracteres de una cadena) ... 354
9.2.4.8 FIND (buscar caracteres en una cadena) ... 355
9.2.5 Información de tiempo de ejecución ... 356
9.2.5.1 GetSymbolName (leer nombre de una variable en el parámetro de entrada) ... 356
9.2.5.2 GetSymbolPath (consultar el nombre global compuesto de la alimentación del parámetro de entrada) ... 359
9.2.5.3 GetInstanceName (leer el nombre de la instancia de bloque) ... 362
9.2.5.4 GetInstancePath (consultar el nombre global compuesto de la instancia de bloque) ... 365
9.2.5.5 GetBlockName (leer el nombre del bloque) ... 367

9.3 E/S distribuidas (PROFINET, PROFIBUS o AS-i) ... 369
9.3.1 Instrucciones de E/S descentralizadas ... 369
9.3.2 RDREC y WRREC (leer/escribir registro) ... 370
9.3.3 GETIO (leer memoria imagen de proceso) ... 373
9.3.4 SETIO (transferir memoria imagen de proceso) ... 374
9.3.5 GETIO_PART (leer área de memoria imagen de proceso) ... 375
9.3.6 SETIO_PART (transferir área de memoria imagen de proceso) ... 376
9.3.7 RALRM (recibir alarma) ... 377
9.3.8 D_ACT_DP (activar/desactivar dispositivos PROFINET IO) ... 380
9.3.9 Parámetro STATUS para RDREC, WRREC y RALRM ... 385
9.3.10 Otras ... 389
9.3.10.1 DPRD_DAT y DPWR_DAT (leer/escribir datos coherentes) ... 389
9.3.10.2 RCVREC (recibir registro con I-device/esclavo I) ... 392
9.3.10.3 PRVREC (proveer registro con I-device/esclavo I) ... 394
9.3.10.4 DPNRM_DG (leer datos de diagnóstico de un esclavo DP) ... 396

9.4 PROFIenergy ... 398

9.5 Alarmas ... 400
9.5.1 Instrucciones ATTACH y DETACH (Asignar OB a evento de alarma/deshacer asignación) ... 400
9.5.2 Alarmas cíclicas ... 403
9.5.2.1 SET_CINT (ajustar parámetros de alarma cíclica) ... 403
9.5.2.2 QRY_CINT (consultar parámetros de alarma cíclica) ... 405
9.5.3 Alarmas horarias: ... 406
9.5.3.1 SET_TINTL (Ajustar alarma horaria) ... 407
9.5.3.2 CAN_TINT (Cancelar alarma horaria) ... 408
9.5.3.3 ACT_TINT (Activar alarma horaria) ... 409
9.5.3.4 QRY_TINT (Consultar estado de alarma horaria) ... 409
9.5.4 Alarmas de retardo ... 410

9.5.5 Instrucciones DIS_AIRT y EN_AIRT (Retardar/habilitar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad) ... 413

9.6 Alarmas ... 414
9.6.1 Gen_UsrMsg (crear avisos de diagnóstico de usuario) ... 414

9.7 Diagnóstico (PROFINET o PROFIBUS) ... 416
9.7.1 Instrucciones de diagnóstico ... 416
9.7.2 RD_SINFO (leer información de arranque del OB actual) ... 417
9.7.3 LED (leer estado del LED) ... 427
9.7.4 Get_IM_Data (leer datos de identificación y mantenimiento) ... 428
9.7.5 Get_Name (leer el nombre de un dispositivo PROFINET IO) ... 429
9.7.6 GetStationInfo (leer la dirección IP o MAC de un dispositivo PROFINET IO) ... 435
9.7.7 Instrucción DeviceStates ... 445
9.7.7.1 Configuraciones de ejemplo de DeviceStates ... 446
9.7.8 Instrucción ModuleStates ... 450
9.7.8.1 Configuraciones de ejemplo de ModuleStates ... 452
9.7.9 GET_DIAG (leer información de diagnóstico) ... 455
9.7.10 GetSMCInfo (leer información sobre la Memory Card) ... 462
9.7.11 Eventos de diagnóstico de E/S descentralizadas ... 465

9.8 Impulso ... 466
9.8.1 CTRL_PWM (modulación del ancho de impulso) ... 466
9.8.2 CTRL_PTO (tren de impulsos) ... 468
9.8.3 Manejo de las salidas de impulso ... 471
9.8.4 Configurar un canal de impulso para PWM o PTO ... 473

9.9 Recetas y registros de datos ... 478
9.9.1 Recetas ... 478
9.9.1.1 Vista general de la receta ... 478
9.9.1.2 Ejemplo de receta ... 479
9.9.1.3 Instrucciones del programa que transfieren datos de receta ... 483
9.9.1.4 Programa de ejemplo de receta ... 487
9.9.2 Registros ... 489
9.9.2.1 Estructura del registro ... 490
9.9.2.2 Instrucciones de programa que controlan registros ... 491
9.9.2.3 Trabajar con registros de datos ... 505
9.9.2.4 Límite del tamaño de los archivos de registro ... 506
9.9.2.5 Programa de ejemplo de registros de datos ... 509

9.10 Control de bloques de datos ... 514
9.10.1 CREATE_DB (crear bloque de datos) ... 514
9.10.2 Instrucciones READ_DBL y WRIT_DBL (Leer de/escribir en un bloque de datos de la memoria de carga) ... 518
9.10.3 ATTR_DB (leer atributos de un bloque de datos) ... 521
9.10.4 DELETE_DB (borrar bloque de datos) ... 523

9.11 Procesamiento de direcciones ... 524
9.11.1 GEO2LOG (determinar la ID de hardware a partir del slot) ... 524
9.11.2 LOG2GEO (determinar el slot a partir de la ID de hardware) ... 526
9.11.3 IO2MOD (determinar la ID de hardware a partir de una dirección ES) ... 527
9.11.4 RD_ADDR (determinar las direcciones ES a partir de la ID de hardware) ... 529
9.11.5 Tipo de datos del sistema GEOADDR ... 530

9.12 Códigos de error comunes para las instrucciones avanzadas ... 531

9.13 Tratamiento de archivos ... 532
9.13.1 FileReadC: leer archivo de la SIMATIC Memory Card ... 532
9.13.2 FileWriteC: escribir archivo en la Memory Card ... 534
9.13.3 FileDelete: Borrar archivo de la Memory Card ... 537

10 Instrucciones tecnológicas ... 539

10.1 Contaje (contadores rápidos) ... 539
10.1.1 Instrucción CTRL_HSC_EXT (controlar contador rápido) ... 540
10.1.1.1 Descripción general de la instrucción ... 540
10.1.1.2 Ejemplo ... 541
10.1.1.3 Tipos de datos de sistema (SDT) de la instrucción CTRL_HSC_EXT ... 544
10.1.2 Funcionamiento del contador rápido ... 549
10.1.2.1 Función de sincronización ... 549
10.1.2.2 Función de puerta ... 550
10.1.2.3 Función de captura ... 552
10.1.2.4 Función de comparación ... 553
10.1.2.5 Aplicaciones ... 553
10.1.3 Configurar un contador rápido ... 554
10.1.3.1 Tipo de contaje ... 556
10.1.3.2 Fase de operación ... 557
10.1.3.3 Valores iniciales ... 560
10.1.3.4 Funciones de entrada ... 561
10.1.3.5 Función de salida ... 562
10.1.3.6 Eventos de alarma ... 562
10.1.3.7 Asignación de los pines de entrada de hardware ... 563
10.1.3.8 Asignación de los pines de salida de hardware ... 565
10.1.3.9 Direcciones de la memoria de entrada del HSC ... 565
10.1.3.10 Identificador de hardware ... 566
10.1.4 Instrucción CTRL_HSC (controlar contadores rápidos) antigua ... 566
10.1.4.1 Descripción general de la instrucción ... 566
10.1.4.2 Uso de CTRL_HSC ... 568
10.1.4.3 Valor de contaje actual del HSC ... 568

10.2 Motion Control ... 569
10.2.1 Sinopsis de Motion Control ... 569
10.2.2 Componentes de hardware para Motion Control ... 569
10.2.3 Instrucciones de Motion Control ... 571
10.2.3.1 Vista general de instrucciones de MC ... 571
10.2.3.2 MC_Power (habilitar/bloquear eje) ... 572
10.2.3.3 MC_Reset (confirmar error) ... 573
10.2.3.4 MC_Home (referenciar eje) ... 573
10.2.3.5 MC_Halt (detener eje) ... 574
10.2.3.6 MC_MoveAbsolute (posicionamiento absoluto del eje) ... 574
10.2.3.7 MC_MoveRelative (posicionamiento relativo del eje) ... 575
10.2.3.8 MC_MoveVelocity (mover el eje a la velocidad predefinida) ... 575
10.2.3.9 MC_MoveJog (desplazamiento del eje en modo Jog) ... 576
10.2.3.10 MC_CommandTable (ejecutar comandos de eje como secuencia de movimiento) ... 576
10.2.3.11 MC_WriteParam (escribir parámetros de un objeto tecnológico) ... 577
10.2.3.12 Instrucción MC_ReadParam (leer parámetros de un objeto tecnológico) ... 577
10.2.3.13 MC_ChangeDynamic (cambiar la configuración de la dinámica del eje) ... 578
10.2.4 Información adicional sobre S7-1200 Motion Control ... 578

10.3 PID Control ... 579

10.3.1 Funciones PID ... 579
10.3.2 Instrucciones PID ... 580
10.3.2.1 Funciones PID ... 580
10.3.2.2 Instrucción PID_Compact ... 580
10.3.2.3 Instrucción PID_3Step ... 581
10.3.2.4 Instrucción PID_Temp ... 582
10.3.3 Información adicional sobre la regulación PID de S7-1200 ... 582

11 Comunicación ... 585

11.1 Sinopsis ... 585
11.2 Comunicación segura (secure) y comunicación antigua (legacy) ... 586
11.3 Protocolos de comunicación y puertos utilizados por la comunicación Ethernet ... 589
11.4 Conexiones de comunicación asíncronas ... 591
11.5 PROFINET ... 594
11.5.1 Crear una conexión de red ... 596
11.5.2 Configurar la vía de conexión local/de interlocutor ... 596
11.5.3 Asignar direcciones IP (Internet Protocol) ... 599
11.5.3.1 Asignar direcciones IP a los dispositivos de programación y red ... 599
11.5.3.2 Comprobar la dirección IP de la programadora ... 601
11.5.3.3 Asignar una dirección IP a una CPU online ... 602
11.5.3.4 Configurar una dirección IP para una CPU en el proyecto ... 603
11.5.4 Comprobar la red PROFINET ... 607
11.5.5 Localizar la dirección Ethernet (MAC) en la CPU ... 608
11.5.6 Configurar la sincronización del Network Time Protocol (NTP) ... 610
11.5.7 Tiempo de arranque de dispositivo, asignación de nombre y de dirección en PROFINET ... 612
11.5.8 Open User Communication ... 613
11.5.8.1 Protocolos ... 613
11.5.8.2 TCP y ISO on TCP ... 613
11.5.8.3 Servicios de comunicación y números de puerto utilizados ... 614
11.5.8.4 Modo ad hoc ... 615
11.5.8.5 ID de conexión para las instrucciones de Open User Communication ... 616
11.5.8.6 Parámetros de la conexión PROFINET ... 618
11.5.8.7 Configuración de DNS ... 626
11.5.8.8 Configurar una conexión OUC en TIA Portal V17 ... 627
11.5.8.9 Instrucciones TSEND_C y TRCV_C ... 630
11.5.8.10 Instrucciones Legacy TSEND_C y TRCV_C ... 644
11.5.8.11 Instrucciones TCON, TDISCON, TSEND y TRCV ... 651
11.5.8.12 TCONSettings ... 661
11.5.8.13 Instrucciones Legacy TCON, TDISCON, TSEND y TRCV ... 667
11.5.8.14 Instrucción T_RESET (Finalizar y restablecer una conexión existente) ... 676
11.5.8.15 Instrucción T_DIAG (Comprueba el estado de la conexión y lee información) ... 678
11.5.8.16 Instrucción TMAIL_C (Envía un correo electrónico utilizando la interfaz Ethernet de la CPU) ... 682
11.5.8.17 UDP ... 706
11.5.8.18 TUSEND y TURCV ... 706
11.5.8.19 T_CONFIG ... 712
11.5.8.20 Parámetros comunes para instrucciones ... 723
11.5.9 Comunicación con una programadora ... 724
11.5.9.1 Establecer la conexión de hardware ... 725
11.5.9.2 Configurar los dispositivos ... 725
11.5.9.3 Asignar direcciones IP (Internet Protocol) ... 726

11.5.9.4 Comprobar la red PROFINET ... 726
11.5.10 Comunicación entre dispositivos HMI y el PLC ... 727
11.5.10.1 Configurar las conexiones de red lógicas entre dos dispositivos ... 728
11.5.11 Comunicación entre PLCs ... 728
11.5.11.1 Configurar las conexiones de red lógicas entre dos dispositivos ... 729
11.5.11.2 Configurar la vía de conexión local/interlocutor entre dos dispositivos ... 730
11.5.11.3 Configurar los parámetros de transmisión y recepción ... 730
11.5.12 Configurar una CPU y un dispositivo PROFINET IO ... 733
11.5.12.1 Agregar un dispositivo PROFINET IO ... 733
11.5.12.2 Asignar CPUs y nombres de dispositivo ... 734
11.5.12.3 Asignar direcciones IP (Internet Protocol) ... 734
11.5.12.4 Configurar el tiempo de ciclo IO ... 735
11.5.13 Configurar una CPU y un I-device PROFINET ... 736
11.5.13.1 Funcionalidad de I-device ... 736
11.5.13.2 Propiedades y ventajas del I-device ... 737
11.5.13.3 Características de un I-device ... 737
11.5.13.4 Intercambio de datos entre un sistema IO de nivel superior y otro subordinado ... 740
11.5.13.5 Configuración del I-device ... 743
11.5.14 Dispositivos compartidos ... 745
11.5.14.1 Funcionalidad de Shared Device ... 745
11.5.14.2 Ejemplo: configurar un Shared Device (configuración GSD) ... 748
11.5.14.3 Ejemplo: Configuración de un I-device como un Shared Device ... 753
11.5.15 Media Redundancy Protocol (MRP) ... 762
11.5.15.1 Redundancia de medios con topologías en anillo ... 765
11.5.15.2 Uso de Media Redundancy Protocol (MRP) ... 766
11.5.15.3 Configurar la redundancia de medios ... 769
11.5.16 Routing S7 ... 772
11.5.16.1 Routing S7 entre interfaces de CPU y CP ... 773
11.5.16.2 Routing S7 entre dos interfaces de CP ... 774
11.5.17 Desactivar SNMP ... 774
11.5.17.1 Desactivar SNMP ... 775
11.5.18 Diagnóstico ... 777
11.5.19 Instrucciones E/S descentralizadas ... 777
11.5.20 Instrucciones de diagnóstico ... 777
11.5.21 Eventos de diagnóstico de E/S descentralizadas ... 777

11.6 PROFIBUS ... 777
11.6.1 Servicios de comunicaciones de CM PROFIBUS ... 779
11.6.2 Referencia a los manuales de usuario de CM PROFIBUS ... 780
11.6.3 Configurar un maestro DP y un dispositivo esclavo ... 781
11.6.3.1 Agregar el módulo CM 1243-5 (maestro DP) y un esclavo DP ... 781
11.6.3.2 Configurar las conexiones de red lógicas entre dos dispositivos PROFIBUS ... 781
11.6.3.3 Asignar direcciones PROFIBUS al módulo CM 1243-5 y al esclavo DP ... 782
11.6.4 Instrucciones E/S descentralizadas ... 783
11.6.5 Instrucciones de diagnóstico ... 783
11.6.6 Eventos de diagnóstico de E/S descentralizadas ... 784

11.7 AS-i ... 784
11.7.1 Configuración de un maestro AS-i y un dispositivo esclavo ... 785
11.7.1.1 Agregar el módulo maestro AS-i CM 1243-2 y esclavo AS-i ... 785
11.7.1.2 Configuración de las conexiones de red lógicas entre dos dispositivos AS-i ... 786
11.7.1.3 Configurar las propiedades del maestro AS-i CM1243-2 ... 786
11.7.1.4 Asignar una dirección AS-i a un esclavo AS-i ... 787

11.7.2 Intercambio de datos entre el programa de usuario y los esclavos AS-i ... 789
11.7.2.1 Configuración básica de STEP 7 ... 789
11.7.2.2 Configurar esclavos con STEP 7 ... 791
11.7.3 Instrucciones E/S descentralizadas ... 793
11.7.4 Trabajar con herramientas online AS-i ... 793

11.8 Comunicación S7 ... 795
11.8.1 GET y PUT (leer y escribir datos de una CPU remota) ... 795
11.8.2 Crear una conexión S7 ... 799
11.8.3 Configurar la vía de conexión local/interlocutor entre dos dispositivos ... 800
11.8.4 Asignación de parámetros de conexión GET/PUT ... 800
11.8.4.1 Parámetros de la conexión ... 801
11.8.4.2 Configurar una conexión S7 de CPU a CPU ... 803

11.9 Qué hacer cuando no se puede acceder a la CPU a través de la dirección IP ... 808

11.10 Servidor OPC UA ... 809
11.10.1 Configuración del servidor OPC UA ... 809
11.10.1.1 Activación del servidor OPC UA ... 810
11.10.1.2 Comportamiento del servidor OPC UA durante el funcionamiento ... 810
11.10.1.3 Configuración del servidor OPC UA ... 812
11.10.1.4 Uso del S7-1200 como servidor OPC UA ... 813
11.10.2 Seguridad del servidor OPC UA ... 815
11.10.2.1 Directivas de seguridad soportadas ... 816
11.10.2.2 Clientes de confianza ... 818
11.10.2.3 Autenticación de usuario ... 818
11.10.3 Interfaz de servidor OPC UA ... 820
11.10.3.1 Tipos de datos soportados ... 820
11.10.3.2 Representación del PLC ... 821
11.10.3.3 Interfaces de servidor descargables ... 823
11.10.4 Búfer de diagnóstico OPC UA ... 825
11.10.4.1 Límites de OPC UA alcanzados ... 826
11.10.4.2 Lectura remota del búfer de diagnóstico en OPC UA ... 828
11.10.4.3 Eventos de seguridad OPC UA ... 829
11.10.4.4 Uso incorrecto de OPC UA ... 835
11.10.4.5 Sinopsis de avisos de OPC UA ... 837
11.10.5 Llamada de métodos OPC UA ... 837
11.10.5.1 Información importante sobre métodos de servidor ... 837
11.10.5.2 Condiciones específicas del uso de métodos de servidor ... 840

12 Servidor web ... 843

12.1 Habilitar el servidor web ... 846

12.2 Configuración de usuarios de servidor web ... 848

12.3 Acceso a las páginas web desde un PC ... 851

12.4 Acceso a las páginas web desde un dispositivo móvil ... 852

12.5 Usar un módulo CP para acceder a páginas web ... 853

12.6 Descargar e instalar un certificado de seguridad ... 854

12.7 Páginas web estándar ... 857
12.7.1 Representación de las páginas web estándar ... 857
12.7.2 Páginas básicas ... 858
12.7.3 Inicio de sesión y privilegios de usuario ... 859

12.7.4 Introducción ... 862
12.7.5 Arranque ... 863
12.7.6 Diagnóstico ... 864
12.7.7 Búfer de diagnóstico ... 866
12.7.8 Información del módulo ... 867
12.7.9 Comunicación ... 872
12.7.10 Estado de variable ... 875
12.7.11 Tablas de observación ... 877
12.7.12 Copia de seguridad online ... 879
12.7.13 Página Registros de datos ... 882
12.7.14 Archivos de usuario ... 884
12.7.15 Data Log UserFiles API ... 889
12.7.16 File Browser ... 889

12.8 Páginas web definidas por el usuario ... 891
12.8.1 Creación de páginas HTML ... 892
12.8.2 Comandos AWP soportados por el servidor web del S7-1200 ... 893
12.8.2.1 Leer variables ... 894
12.8.2.2 Escribir variables ... 895
12.8.2.3 Leer variables especiales ... 897
12.8.2.4 Escribir variables especiales ... 899
12.8.2.5 Utilizar un alias para una referencia de variable ... 900
12.8.2.6 Definir tipos de enumeración ... 901
12.8.2.7 Referenciar variables de CPU con un tipo de enumeración ... 901
12.8.2.8 Crear fragmentos ... 903
12.8.2.9 Importar fragmentos ... 904
12.8.2.10 Combinar definiciones ... 905
12.8.2.11 Procesamiento de nombres de variable que contienen caracteres especiales ... 905
12.8.3 Configurar el uso de las páginas web definidas por el usuario ... 907
12.8.4 Configurar la página de entrada ... 908
12.8.5 Programar la instrucción WWW para páginas web definidas por el usuario ... 909
12.8.6 Cargar los bloques de programa en la CPU ... 910
12.8.7 Acceso a las páginas web definidas por el usuario ... 911
12.8.8 Limitaciones específicas de las páginas web definidas por el usuario ... 911
12.8.9 Ejemplo de una página web definida por el usuario ... 913
12.8.9.1 Página web para vigilar y controlar una turbina de viento ... 913
12.8.9.2 Leer y visualizar datos del controlador ... 915
12.8.9.3 Utilizar un tipo de enumeración ... 916
12.8.9.4 Escribir una entrada del usuario en el controlador ... 917
12.8.9.5 Escribir una variable especial ... 918
12.8.9.6 Referencia: Lista HTML de la página web para la vigilancia remota de la turbina de viento ... 918
12.8.9.7 Configuración en STEP 7 de la página web de ejemplo ... 923
12.8.10 Configurar páginas web definidas por el usuario en varios idiomas ... 924
12.8.10.1 Crear la estructura de carpetas ... 924
12.8.10.2 Programar el cambio de idioma ... 924
12.8.10.3 Configurar STEP 7 para utilizar una estructura de página multilingüe ... 927
12.8.11 Control avanzado de páginas web definidas por el usuario ... 927
12.8.12 Web API ... 931
12.8.12.1 Web API ... 931
12.8.12.2 Métodos Web API soportados ... 932

12.9 Limitaciones ... 932
12.9.1 Uso de JavaScript ... 934

12.9.2 Restricciones de funciones cuando las opciones de Internet no permiten cookies 934
12.9.3 Reglas para introducir nombres de variables y valores 934
12.9.4 Importar registros de datos en formato CSV a versiones de Microsoft Excel que no son estadounidenses ni británicas 935

13 Procesador de comunicaciones y Modbus TCP 937

13.1 Utilizar las interfaces de comunicación serie 937

13.2 Polarizar y terminar un conector de red RS485 938

13.3 Comunicación punto a punto (PtP) 939
13.3.1 Comunicación PtP, Freeport 940
13.3.2 Comunicación 3964(R) 942
13.3.3 Configurar la comunicación PtP Freeport 943
13.3.3.1 Gestionar el control de flujo 945
13.3.3.2 Configurar los parámetros de transmisión 946
13.3.3.3 Configurar los parámetros de recepción 947
13.3.4 Configurar la comunicación 3964(R) 956
13.3.4.1 Configurar los puertos de comunicación 3964(R) 956
13.3.4.2 Configurar la prioridad y los parámetros de protocolo para 3964(R) 957
13.3.5 Instrucciones de comunicación punto a punto 959
13.3.5.1 Parámetros comunes de las instrucciones de comunicación punto a punto 959
13.3.5.2 Port_Config (configurar los parámetros de comunicación dinámicamente) 961
13.3.5.3 Send_Config (configurar los parámetros de transmisión en serie dinámicamente) 964
13.3.5.4 Receive_Config (configurar los parámetros de recepción en serie dinámicamente) 966
13.3.5.5 P3964_Config (configurar el protocolo 3964(R)) 971
13.3.5.6 Send_P2P (transmitir datos del búfer de transmisión) 973
13.3.5.7 Receive_P2P (habilitar recepción de mensajes) 976
13.3.5.8 Receive_Reset (borrar el búfer de recepción) 978
13.3.5.9 Signal_Get (consultar señales RS-232) 979
13.3.5.10 Signal_Set (activar señales RS-232) 980
13.3.5.11 Get_Features 982
13.3.5.12 Set_Features 983
13.3.6 Programar la comunicación PtP 984
13.3.6.1 Arquitectura de sondeo 985
13.3.7 Ejemplo: Comunicación punto a punto 986
13.3.7.1 Configurar los módulos de comunicación 987
13.3.7.2 Estados operativos de RS422 y RS485 989
13.3.7.3 Programar el programa de STEP 7 992
13.3.7.4 Configurar el emulador de terminal 993
13.3.7.5 Ejecutar el programa de ejemplo 994

13.4 Comunicación de interfaz serie universal (USS) 994
13.4.1 Seleccionar la versión de las instrucciones USS 997
13.4.2 Requisitos para utilizar el protocolo USS 998
13.4.3 Instrucciones USS 1000
13.4.3.1 USS_Port_Scan (procesar comunicación vía red USS) 1000
13.4.3.2 USS_Drive_Control (intercambio de datos con el accionamiento) 1002
13.4.3.3 USS_Read_Param (leer datos del accionamiento) 1005
13.4.3.4 USS_Write_Param (modificar datos del accionamiento) 1006
13.4.4 Códigos de estado USS 1008
13.4.5 Requisitos generales para la configuración del accionamiento USS 1010
13.4.6 Ejemplo: conexión e instalación general del accionamiento USS 1010

13.5 Comunicación Modbus ... 1013
13.5.1 Sinopsis de la comunicación Modbus RTU y Modbus TCP ... 1013
13.5.2 Modbus TCP ... 1016
13.5.2.1 Vista general ... 1016
13.5.2.2 Seleccionar la versión de las instrucciones Modbus TCP ... 1016
13.5.2.3 Instrucciones Modbus TCP ... 1017
13.5.2.4 Ejemplos de Modbus TCP ... 1079
13.5.3 Modbus RTU ... 1084
13.5.3.1 Vista general ... 1084
13.5.3.2 Seleccionar la versión de las instrucciones Modbus RTU ... 1086
13.5.3.3 Número máximo de esclavos Modbus soportados ... 1086
13.5.3.4 Instrucciones Modbus RTU ... 1087
13.5.3.5 Ejemplos de Modbus RTU ... 1108

13.6 Comunicación Legacy PtP (solo CM/CB 1241) ... 1112
13.6.1 Instrucciones Legacy punto a punto ... 1112
13.6.1.1 PORT_CFG (configurar los parámetros de comunicación dinámicamente) ... 1112
13.6.1.2 SEND_CFG (configurar los parámetros de transmisión en serie dinámicamente) ... 1114
13.6.1.3 RCV_CFG (configurar parámetros de recepción serie dinámicamente) ... 1115
13.6.1.4 SEND_PTP (transmitir datos del búfer de transmisión) ... 1120
13.6.1.5 RCV_PTP (habilitar recepción de mensajes) ... 1122
13.6.1.6 RCV_RST (borrar búfer de recepción) ... 1124
13.6.1.7 SGN_GET (consultar señales RS-232) ... 1125
13.6.1.8 SGN_SET (activar señales RS-232) ... 1126

13.7 Comunicación Legacy USS (solo CM/CB 1241) ... 1127
13.7.1 Seleccionar la versión de las instrucciones USS ... 1128
13.7.2 Requisitos para utilizar el protocolo USS ... 1129
13.7.3 Instrucciones Legacy USS ... 1131
13.7.3.1 Instrucción USS_PORT (Procesar comunicación vía red USS) ... 1131
13.7.3.2 Instrucción USS_DRV (Intercambiar datos con el accionamiento) ... 1133
13.7.3.3 Instrucción USS_RPM (Leer parámetros del accionamiento) ... 1136
13.7.3.4 Instrucción USS_WPM (Modificar parámetros en el accionamiento) ... 1137
13.7.4 Códigos de estado Legacy USS ... 1139
13.7.5 Requisitos generales para la configuración del accionamiento Legacy USS ... 1141

13.8 Comunicación Legacy Modbus TCP ... 1141
13.8.1 Vista general ... 1141
13.8.2 Seleccionar la versión de las instrucciones Modbus TCP ... 1141
13.8.3 Instrucciones Legacy Modbus TCP ... 1142
13.8.3.1 MB_CLIENT (comunicar como cliente TCP Modbus vía PROFINET) ... 1142
13.8.3.2 MB_SERVER (comunicar como servidor TCP Modbus vía PROFINET) ... 1148
13.8.4 Ejemplos de Legacy Modbus TCP ... 1154
13.8.4.1 Ejemplo: Legacy MB_SERVER conexiones TCP múltiples ... 1154
13.8.4.2 Ejemplo: Legacy MB_CLIENT 1: Peticiones múltiples con conexión TCP común ... 1154
13.8.4.3 Ejemplo: Legacy MB_CLIENT 2: Peticiones múltiples con varias conexiones TCP ... 1155
13.8.4.4 Ejemplo: Legacy MB_CLIENT 3: Petición de escritura de imagen de salida ... 1156
13.8.4.5 Ejemplo: Legacy MB_CLIENT 4: Coordinación de varias peticiones ... 1157

13.9 Comunicación Legacy Modbus RTU (solo CM/CB 1241) ... 1158
13.9.1 Vista general ... 1158
13.9.2 Seleccionar la versión de las instrucciones Modbus RTU ... 1158
13.9.3 Instrucciones Legacy Modbus RTU ... 1159
13.9.3.1 MB_COMM_LOAD (configurar puerto en módulo PtP para Modbus RTU) ... 1159

13.9.3.2 MB_MASTER (comunicar como maestro Modbus vía puerto PtP) ... 1162
13.9.3.3 MB_SLAVE (comunicar como esclavo Modbus vía puerto PtP) ... 1168
13.9.4 Ejemplos de Modbus RTU antiguo ... 1174
13.9.4.1 Ejemplo: programa de maestro Legacy Modbus RTU ... 1174
13.9.4.2 Ejemplo: programa de esclavo Legacy Modbus RTU ... 1175

13.10 Industrial Remote Communication (IRC) ... 1177
13.10.1 Sinopsis de los CP de Telecontrol ... 1177
13.10.2 Conexión a una red GSM ... 1180
13.10.3 Aplicaciones del CP 1242-7 ... 1181
13.10.4 Otras propiedades del CP 1242-7 ... 1182
13.10.5 Más información ... 1182
13.10.6 Accesorios ... 1183
13.10.7 Ejemplos de configuración de telecontrol ... 1184

14 Comunicación TeleService (correo electrónico SMTP) ... 1189

14.1 Instrucción TM_Mail (Transmitir e-mail) ... 1189

15 Herramientas online y diagnóstico ... 1197

15.1 LEDs de estado ... 1197

15.2 Establecer una conexión online con una CPU ... 1201

15.3 Asignar un nombre a un dispositivo PROFINET IO online ... 1202

15.4 Ajustar la dirección IP y la hora ... 1204

15.5 Actualización del firmware ... 1204

15.6 Establecer o borrar la contraseña de protección de datos de configuración confidenciales del PLC ... 1205

15.7 Restablecer la configuración de fábrica ... 1206

15.8 Comprobar si un módulo tiene algún defecto (guardar datos de servicio) ... 1207

15.9 Formatear una SIMATIC Memory Card desde STEP 7 ... 1209

15.10 Panel de control de la CPU online ... 1210

15.11 Vigilar el tiempo de ciclo y la carga de la memoria ... 1211

15.12 Visualizar los eventos de diagnóstico de la CPU ... 1211

15.13 Comparar CPUs online y offline ... 1213

15.14 Realizar una comparación online/offline de la topología ... 1213

15.15 Vigilar y modificar valores en la CPU ... 1215
15.15.1 Conexión online para observar los valores en la CPU ... 1215
15.15.2 Visualización del estado en el editor de programas ... 1216
15.15.3 Obtener una instantánea de los valores online de un DB para restaurar valores ... 1217
15.15.4 Utilizar una tabla de observación para observar y modificar valores en la CPU ... 1218
15.15.4.1 Utilizar un disparador para observar o forzar variables PLC ... 1220
15.15.4.2 Desbloquear las salidas en modo STOP ... 1221
15.15.5 Forzar valores permanentemente en la CPU ... 1221
15.15.5.1 Utilizar la tabla de forzado permanente ... 1221
15.15.5.2 Funcionamiento de la función de forzado permanente ... 1222

15.16 Carga en estado operativo RUN ... 1224

15.16.1 Requisitos para la "Carga en estado operativo RUN" ... 1225
15.16.2 Modificar el programa en el estado operativo RUN ... 1226
15.16.3 Descargar bloques seleccionados ... 1227
15.16.4 Descargar un único bloque seleccionado con un error de compilación en otro bloque ... 1228
15.16.5 Modificación y carga de los bloques existentes en estado operativo RUN ... 1229
15.16.6 Reacción del sistema si falla la descarga ... 1232
15.16.7 Factores que se deben tener en cuenta al cargar en estado operativo RUN ... 1232

15.17 Trazas y registros de datos de la CPU en condiciones de disparo ... 1234

15.18 Determinar el tipo de condición de rotura de hilo en un módulo SM 1231 ... 1236

15.19 Realizar una copia de seguridad de la CPU y restaurarla ... 1238
15.19.1 Opciones de copia de seguridad y restauración ... 1238
15.19.2 Realizar una copia de seguridad de una CPU online ... 1240
15.19.3 Restaurar una CPU ... 1242

A Datos técnicos ... 1245

A.1 Página web de Siemens Online Support ... 1245

A.2 Datos técnicos generales ... 1245

A.3 Asignación de pines del puerto X1 de la interfaz PROFINET ... 1254

A.4 CPU 1211C ... 1255
A.4.1 Especificaciones generales y propiedades ... 1255
A.4.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1211C ... 1257
A.4.3 Entradas y salidas digitales ... 1261
A.4.4 Entradas analógicas ... 1263
A.4.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU ... 1263
A.4.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU ... 1264
A.4.4.3 Rangos de medida de entradas analógicas de tensión (CPU) ... 1264
A.4.5 Diagramas de cableado de la CPU 1211C ... 1265

A.5 CPU 1212C ... 1268
A.5.1 Especificaciones generales y propiedades ... 1268
A.5.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1212C ... 1270
A.5.3 Entradas y salidas digitales ... 1273
A.5.4 Entradas analógicas ... 1275
A.5.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU ... 1276
A.5.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU ... 1276
A.5.4.3 Rangos de medida de entradas analógicas de tensión (CPU) ... 1276
A.5.5 Diagramas de cableado de la CPU 1212C ... 1277

A.6 CPU 1214C ... 1280
A.6.1 Especificaciones generales y propiedades ... 1280
A.6.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1214C ... 1282
A.6.3 Entradas y salidas digitales ... 1285
A.6.4 Entradas analógicas ... 1287
A.6.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU ... 1288
A.6.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU ... 1288
A.6.4.3 Rangos de medida de entradas analógicas de tensión (CPU) ... 1288
A.6.5 Diagramas de cableado de la CPU 1214C ... 1289

A.7 CPU 1215C ... 1292
A.7.1 Especificaciones generales y propiedades ... 1292
A.7.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1215C ... 1294

A.7.3 Entradas y salidas digitales ... 1298
A.7.4 Entradas y salidas analógicas ... 1300
A.7.4.1 Respuesta a escalón de las entradas analógicas integradas en la CPU ... 1300
A.7.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU ... 1301
A.7.4.3 Rangos de medida de entradas analógicas de tensión (CPU) ... 1301
A.7.4.4 Especificaciones de salidas analógicas ... 1301
A.7.5 Diagramas de cableado de la CPU 1215C ... 1303

A.8 CPU 1217C ... 1307
A.8.1 Especificaciones generales y propiedades ... 1307
A.8.2 Temporizadores, contadores y bloques lógicos admitidos por la CPU 1217C ... 1309
A.8.3 Entradas y salidas digitales ... 1313
A.8.4 Entradas y salidas analógicas ... 1316
A.8.4.1 Datos técnicos de la entrada analógica ... 1316
A.8.4.2 Respuesta a escalón de las entradas analógicas integradas en la CPU ... 1317
A.8.4.3 Tiempo de muestreo para los puertos analógicos integrados en la CPU ... 1317
A.8.4.4 Rangos de medida de entradas analógicas de tensión (CPU) ... 1318
A.8.4.5 Especificaciones de salidas analógicas ... 1318
A.8.5 Diagramas de cableado de la CPU 1217C ... 1319
A.8.6 CPU 1217C Detalle y ejemplo de aplicación de entrada diferencial (DI) ... 1321
A.8.7 CPU 1217C Detalle y ejemplo de aplicación de salida diferencial (DQ) ... 1322

A.9 Módulos de señales digitales (SMs) ... 1323
A.9.1 Datos técnicos del módulo de entradas digitales SM 1221 ... 1323
A.9.2 Datos técnicos del módulo de salidas digitales SM 1222 de 8 salidas ... 1325
A.9.3 Datos técnicos del módulo de salidas digitales SM 1222 de 16 salidas ... 1326
A.9.4 Especificaciones de la SM 1223 entrada/salida digital V DC ... 1332
A.9.5 Especificaciones de la SM 1223 entrada/salida digital V AC ... 1339

A.10 Módulos de señales analógicos (SMs) ... 1341
A.10.1 Datos técnicos del módulo de entradas analógicas SM 1231 ... 1341
A.10.2 Datos técnicos del módulo de salidas analógicas SM 1232 ... 1346
A.10.3 Datos técnicos del módulo de entradas/salidas analógicas SM 1234 ... 1348
A.10.4 Respuesta a un escalón de las entradas analógicas ... 1351
A.10.5 Tiempo de muestreo y tiempos de actualización para entradas analógicas ... 1351
A.10.6 Rangos de medida de entradas analógicas de tensión e intensidad (SB y SM) ... 1352
A.10.7 Rangos de medida de salidas analógicas de tensión e intensidad (SB y SM) ... 1353

A.11 Módulos de señales RTD y de termopar (SMs) ... 1354
A.11.1 SM 1231 de termopar ... 1354
A.11.1.1 Funcionamiento básico de un termopar ... 1356
A.11.1.2 Tablas de selección de termopares para el SM 1231 ... 1357
A.11.2 SM 1231 RTD ... 1360
A.11.2.1 Tablas de selección para el SM 1231 RTD ... 1363

A.12 Módulos tecnológicos ... 1366
A.12.1 Maestro SM 1278 4xIO-Link ... 1366
A.12.1.1 Vista general del maestro SM 1278 4xIO-Link ... 1369
A.12.1.2 Conexión ... 1371
A.12.1.3 Parámetros/espacio de direcciones ... 1374
A.12.1.4 Alarmas, alarmas de error y de sistema ... 1377

A.13 Signal Boards digitales (SBs) ... 1380
A.13.1 Datos técnicos de la SB 1221 de entradas digitales 200 kHz ... 1380
A.13.2 Datos técnicos de la SB 1222 de salidas digitales 200 kHz ... 1382

A.13.3 Datos técnicos de la SB 1223 de entradas/salidas digitales 200 kHz 1385
A.13.4 Especificaciones de la SB 1223 2 entradas de 24 V DC / 2 salidas de 24 V DC 1388

A.14 Signal Boards analógicas (SBs) 1391
A.14.1 Datos técnicos de la SB 1231 de 1 entrada analógica 1391
A.14.2 Datos técnicos de la SB 1232 de 1 salida analógica 1393
A.14.3 Rangos de medida para entradas y salidas analógicas 1395
A.14.3.1 Respuesta a un escalón de las entradas analógicas 1395
A.14.3.2 Tiempo de muestreo y tiempos de actualización para entradas analógicas 1395
A.14.3.3 Rangos de medida de entradas analógicas de tensión e intensidad (SB y SM) 1395
A.14.3.4 Rangos de medida de salidas analógicas de tensión e intensidad (SB y SM) 1396
A.14.4 Signal Boards de termopar (SB) 1398
A.14.4.1 Datos técnicos de la SB 1231 de termopar de 1 entrada analógica 1398
A.14.4.2 Funcionamiento básico de un termopar 1399
A.14.5 Signal Boards RTD (SB) 1402
A.14.5.1 Datos técnicos de la SB 1231 de 1 entrada analógica RTD 1402
A.14.5.2 Tablas de selección para la SB 1231 RTD 1405

A.15 BB 1297 Battery Board 1407

A.16 Interfaces de comunicación 1408
A.16.1 PROFIBUS 1408
A.16.1.1 CM 1242-5 ESCLAVO PROFIBUS DP 1408
A.16.1.2 Asignación de pines del conector hembra Sub D del CM 1242-5 1410
A.16.1.3 CM 1243-5 maestro PROFIBUS DP 1410
A.16.1.4 Asignación de pines del conector sub-D del CM 1243-5 1412
A.16.2 CP 1242-7 1412
A.16.2.1 CP 1242-7 GPRS 1413
A.16.2.2 Antena GSM/GPRS ANT794-4MR 1414
A.16.2.3 Antena plana ANT794-3M 1415
A.16.3 CM 1243-2 AS-i Master 1416
A.16.3.1 Datos técnicos del maestro AS-i CM 1243-2 1416
A.16.3.2 Conexiones eléctricas del maestro AS-i 1417
A.16.4 RS232, RS422 y RS485 1418
A.16.4.1 Datos técnicos de CB 1241 RS485 1418
A.16.4.2 Datos técnicos del CM 1241 RS232 1421
A.16.4.3 Datos técnicos del CM 1241 RS422/485 1422

A.17 TeleService (TS Adapter y TS Adapter modular) 1423

A.18 SIMATIC Memory Cards 1424

A.19 Simuladores de entradas 1424

A.20 Módulo del potenciómetro de S7-1200 1426

A.21 Cable para módulos de ampliación 1427

A.22 Productos adicionales 1428
A.22.1 Módulo de alimentación PM 1207 1428
A.22.2 Módulo de conmutación compacto CSM 1277 1428
A.22.3 Módulo CM CANopen 1428
A.22.4 Módulo de comunicación RF120C 1429
A.22.5 Módulo SM 1238 Energy Meter 1429
A.22.6 Sistemas de pesaje electrónico SIWAREX 1430

B Calcular la corriente necesaria 1431
C Información de pedido 1435
C.1 Módulos CPU 1435
C.2 Módulos de señales (SM), Signal Boards (SB) y Battery Boards (BB) 1435
C.3 Comunicación 1437
C.4 CPUs y módulos de señales de seguridad positiva 1438
C.5 Otros módulos 1438
C.6 Memory Cards 1439
C.7 Dispositivos HMI Basic 1439
C.8 Repuestos y hardware adicional 1439
C.9 Software de programación 1446
C.10 Licencias OPC UA 1446
D Cambio de dispositivo y compatibilidad de los repuestos 1447
D.1 Sustituir una CPU con protección de datos de configuración confidenciales del PLC 1447
D.2 Sustitución de una CPU V3.0 por una CPU V4.x 1448
D.3 Kits de repuesto de bloques de terminales para S7-1200 V3.0 e inferiores 1453
Índice alfabético 1455

Sinopsis del producto

1

1.1 Introducción al PLC S7-1200

El controlador S7-1200 ofrece la flexibilidad y potencia necesarias para controlar una gran variedad de dispositivos para las distintas necesidades de automatización. Gracias a su estructura compacta, configuración flexible y amplio juego de instrucciones, el S7-1200 es idóneo para controlar una gran variedad de aplicaciones.

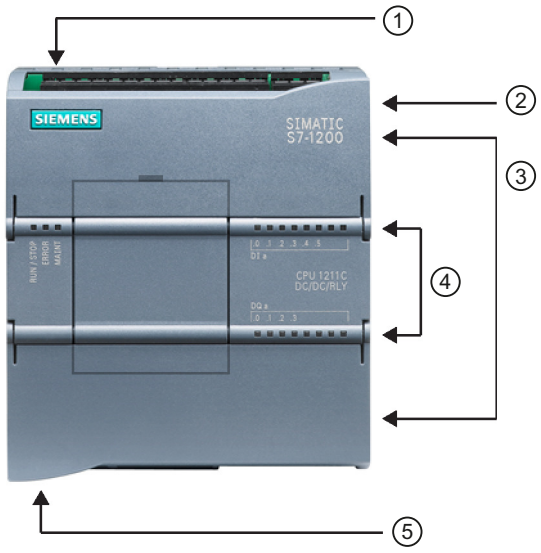
La CPU combina los siguientes elementos, además de otros, en una caja compacta para crear un controlador potente:

- un microprocesador
- una fuente de alimentación integrada
- circuitos de entrada y salida
- PROFINET incorporado
- E/S rápidas de Motion Control

Una vez cargado el programa en la CPU, esta contiene la lógica necesaria para vigilar y controlar los dispositivos de la aplicación. La CPU vigila las entradas y cambia el estado de las salidas según la lógica del programa de usuario, que puede incluir lógica booleana, instrucciones de contaje y temporización, funciones matemáticas complejas, Motion Control, así como comunicación con otros dispositivos inteligentes.

La CPU incorpora un puerto PROFINET para la comunicación en una red PROFINET. Hay disponibles módulos adicionales para la comunicación a través de redes y protocolos como los siguientes:

- PROFIBUS
- GPRS
- LTE
- WAN con funciones integradas de seguridad (firewall, VPN)
- RS485
- RS232
- RS422
- IEC 60870
- DNP3
- USS
- MODBUS



① Conector de corriente
② Ranura para Memory Card (debajo de la tapa superior)
③ Conectores extraíbles para el cableado de usuario (detrás de las tapas)
④ LED de estado para las E/S integradas
⑤ Conector PROFINET (en el lado inferior de la CPU)

Numerosas funciones de seguridad protegen el acceso tanto a la CPU como al programa de control:

- La protección por contraseña (Página 163) permite configurar el acceso a las funciones de la CPU.
- La "protección de know-how" (Página 167) oculta el código dentro de un bloque específico.
- La protección anticopia (Página 168) vincula el programa a una SIMATIC Memory Card o CPU específica.
- Protección de datos de configuración confidenciales del PLC (Página 161)
- Comunicación PG/PC y HMI segura (Página 166)

Tabla 1-1 Comparación de los modelos de CPU

Función		**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
Dimensiones físicas (mm)		90 x 100 x 75		110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Memoria de usuario	Trabajo	50 kB	75 kB	100 kB	125 kB	150 kB
	Carga	1 MB	2 MB	4 MB		
	Remanente	14 KB				
E/S locales integradas	Digitales	6 entradas/ 4 salidas	8 entradas/ 6 salidas	14 entradas/ 10 salidas		
	Analógicas	2 entradas			2 entradas/2 salidas	
Tamaño de la memoria imagen de proceso	Entradas (I)	1024 bytes				
	Salidas (Q)	1024 bytes				
Área de marcas (M)		4096 bytes		8192 bytes		
Ampliación con módulo de señales (SM)		Ninguna	2	8		
Signal board (SB), Battery Board (BB) o Communication Board (CB)		1				

Función		**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
Módulo de comunicación (CM) (ampliación en el lado izquierdo)		3				
Contadores rápidos	Total	Se han configurado un máximo de 6 para usar cualquier entrada integrada o de SB.				
	1 MHz	-				De Ib.2 a Ib.5
	100/[1]80 kHz	De Ia.0 a Ia.5				
	30/[1]20 kHz	--	De Ia.6 a Ia.7	De Ia.6 a Ib.5		De Ia.6 a Ib.1
	200 kHz[3]					
Salidas pulsadas[2]	Total	Se han configurado un máximo de 4 para usar cualquier salida integrada o de SB				
	1 MHz	--				De Qa.0 a Qa.3
	100 kHz	De Qa.0 a Qa.3				De Qa.4 a Qb.1
	20 kHz	--	De Qa.4 a Qa.5	De Qa.4 a Qb.1		--
SIMATIC Memory Card		SIMATIC Memory Card (opcional)				
Registros (Data logs)	Número	Como máximo 8 abiertos a la vez				
	Tamaño	500 MB por registro o limitado por la memoria de carga máxima disponible				
Tiempo de respaldo del reloj de tiempo real		20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)				
Puerto de comunicación Ethernet PROFINET		1			2	
Velocidad de ejecución de funciones matemáticas con números reales		2,3 µs/instrucción				
Velocidad de ejecución booleana		0,08 µs/instrucción				

[1] La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

[3] Se dispone de hasta 200 kHz con la SB 1221 DI x 24 V DC 200 kHz y la SB 1221 DI 4 x 5 V DC 200 kHz.

Los diferentes modelos de CPU ofrecen una gran variedad de funciones y prestaciones que permiten crear soluciones efectivas destinadas a numerosas aplicaciones. Para más información sobre una CPU en particular, consulte los datos técnicos (Página 1245).

Tabla 1-2 Bloques, temporizadores y contadores que soporta el S7-1200

Elemento			**Descripción**				
Bloques	Tipo		OB, FB, FC, DB				
	Tamaño	**Modelo de CPU**	**CPU 1211C**	**CPU 1212C**	**CPU 1214C**	**CPU 1215C**	**CPU 1217C**
		Bloques lógicos	50KB	64KB	64KB	64KB	64KB
		Bloques de datos enlazados[1]	50KB	75KB	100KB	125KB	150KB
		Bloques de datos no enlazados[2]	256KB	256KB	256KB	256KB	256KB
	Cantidad		Un total de hasta 1024 bloques (OB + FB + FC + DB)				
	Profundidad de anidamiento		16 desde el OB de ciclo o arranque; 6 desde cualquier OB de evento de alarma[3]				
	Observar		Se puede observar a la vez el estado de 2 bloques lógicos				

Elemento		Descripción
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarmas de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

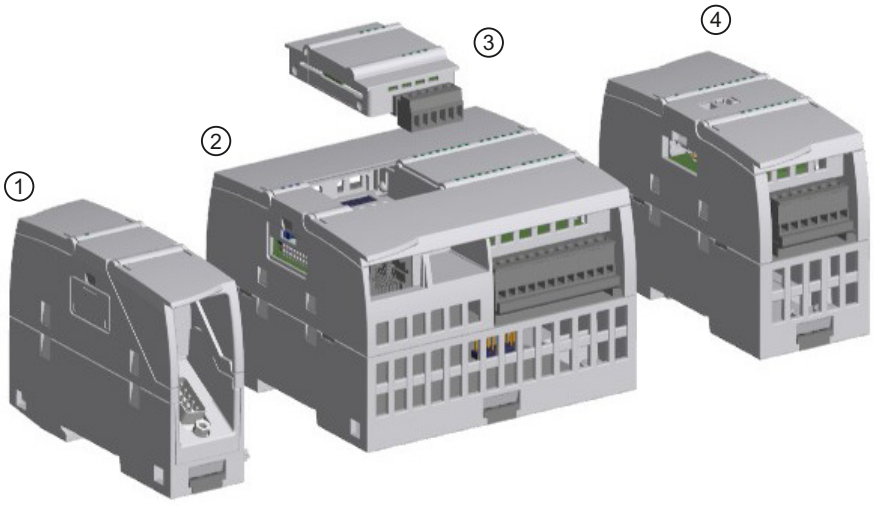
[1] Guardado en la memoria de trabajo y la memoria de carga. No puede exceder el tamaño de la memoria de trabajo o carga restante.

[2] Guardado solo en la memoria de carga

[3] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

1.2 Capacidad de expansión de la CPU

La familia S7-1200 ofrece diversos módulos y placas de conexión para ampliar las capacidades de la CPU con E/S adicionales y otros protocolos de comunicación. Para más información sobre un módulo en particular, consulte los datos técnicos (Página 1245).



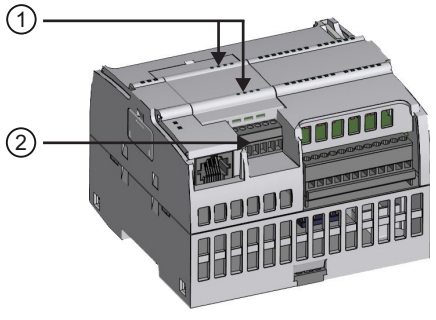
① Módulo de comunicación (CM) o procesador de comunicaciones (CP) (Página 1408)

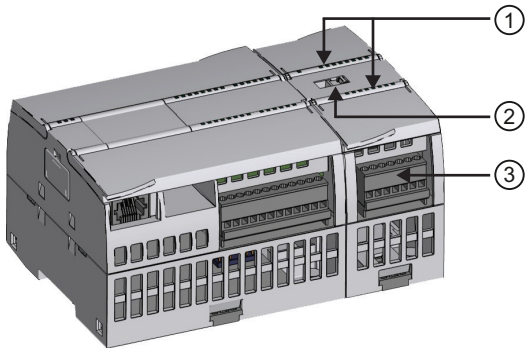
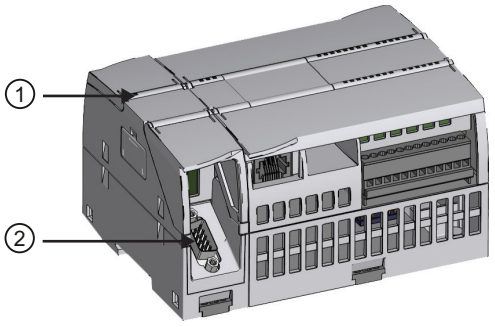
② CPU (CPU 1211C (Página 1255), CPU 1212C (Página 1268), CPU 1214C (Página 1280), CPU 1215C (Página 1292), CPU 1217C (Página 1307))

③ Signal Board (SB) (SB digital (Página 1380), SB analógica (Página 1391)), Communication Board (CB) (Página 1418) o Battery Board (BB)CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C) (Página 1407)

④ Módulo de señales (SM) (SM digital (Página 1323), SM analógico (Página 1341), SM termopar (Página 1354), SM RTD (Página 1360), SM tecnológico) (Página 1366)

Tabla 1-3 Módulos de ampliación S7-1200

Tipo de módulo	Descripción
La CPU soporta una placa de ampliación tipo plug-in: • Una Signal Board (SB) proporciona E/S adicionales a la CPU. La SB se conecta en la parte frontal de la CPU. • Una placa de comunicación (CB) permite agregar un puerto de comunicación adicional a la CPU. • Una placa de batería (BB) ofrece respaldo a largo plazo del reloj en tiempo real.	 ① LEDs de estado en la SB ② Conector extraíble para el cableado de usuario

Tipo de módulo	Descripción
Los módulos de señales (SM) agregan funciones a la CPU. Los SM se conectan en el lado derecho de la CPU. • E/S digitales • E/S analógicas • RTD y termopar • Maestro SM 1278 IO-Link • SM 1238 Energy Meter (https://support.industry.siemens.com/cs/ww/en/view/109483435)	 ① LEDs de estado ② Lengüeta de desplazamiento del conector de bus ③ Conector extraíble para el cableado de usuario
Los módulos de comunicación (CM) y los procesadores de comunicaciones (CP) agregan opciones de comunicación a la CPU, p. ej. para la conectividad de PROFIBUS o RS232/RS485 (para PtP, Modbus o USS) o el maestro AS-i. Un CP ofrece funciones para otros tipos de comunicación, como la conexión a la CPU a través de una red GPRS, LTE, IEC, DNP3, o WDC. • La CPU soporta hasta tres CMs o CPs • Cada CM o CP se conecta en el lado izquierdo de la CPU (o en el lado izquierdo de otro CM o CP)	 ① LEDs de estado ② Conector de comunicación

1.3 Paneles HMI Basic

Los SIMATIC HMI Basic Panels incorporan pantalla táctil para el control básico por parte del operador y tareas de control. Todos los paneles ofrecen el grado de protección IP65 y certificación CE, UL, cULus y NEMA 4x.

Los paneles HMI Basic (Página 1439)disponibles se describen a continuación:

- KTP400 Basic: pantalla táctil de 4 pulgadas con 4 teclas configurables, resolución de 480 x 272 y 800 variables
- KTP700 Basic: pantalla táctil de 7 pulgadas con 8 teclas configurables, resolución de 800 x 480 y 800 variables
- KTP700 Basic DP: pantalla táctil de 7 pulgadas con 8 teclas configurables, resolución de 800 x 480 y 800 variables
- KTP900 Basic: pantalla táctil de 9 pulgadas con 8 teclas configurables, resolución de 800 x 480 y 800 variables

- KTP1200 Basic: pantalla táctil de 12 pulgadas con 10 teclas configurables, resolución de 800 x 480 y 800 variables
- KTP 1200 Basic DP: pantalla táctil de 12 pulgadas con 10 teclas configurables, resolución de 800 x 400 y 800 variables

Nuevas funciones

2

La versión V4.5 ofrece las siguientes funciones nuevas:

- Mejoras del S7-1200 relativas a OPC UA (Página 809):
  - Llamadas a método de servidor (llamadas a procedimiento remoto)
  - Tipos de datos estructurados y de array
  - Diagnóstico mejorado
- Instrucciones nuevas:
  - La instrucción GetSMCInfo recoge información sobre la SIMATIC Memory Card enchufada
  - Instrucciones de lectura/escritura de archivo compactas: FileReadC (Página 532), FileWriteC (Página 534) y FileDelete (Página 537)
- Tipos de conexión OUC TCP, Iso-on-TCP y UDP
- La Open User Communication ahora soporta TCON_Settings (Página 661)
- Servidor web: soporta la API moderna, además de la administración de certificados
- Soporte PROFINET de la función Media Redundancy Protocol (MRP) (Página 762) como "Client" y como "Manager" (CPU 1215C y CPU 1217C)
- Función DataLog mejorada, con campo de sello de tiempo de sincronización con S7-1500
- Nueva vista general de puertos de comunicación y protocolos (Página 589) utilizados en la comunicación Ethernet
- Seguridad mejorada:
  - Uso de certificados X.509 y TLS (Transport Layer Security) para habilitar la comunicación PG/PC y HMI segura (Página 166)
  - Protección de datos de configuración confidenciales del PLC (Página 161)
  - Asistente de seguridad (Página 160) en TIA Portal para facilitar la configuración de la comunicación segura y con funciones de protección.
  - Soporte continuo de la comunicación antigua (legacy) además de la comunicación segura (Página 586)
  - Cifrado mejorado de las contraseñas de nivel de acceso de la CPU (Página 163) con un ajuste predeterminado de protección completa de la CPU
  - Capacidad para el uso de una SIMATIC Memory Card para establecer o modificar la contraseña para la Protección de datos de configuración confidenciales del PLC (Página 130)
- Memoria remanente ampliada para CPU S7-1200 de 10 KB a 14 KB

- Homologación CCC para productos S7-1200
  No todos los modelos del S7-1200 pueden certificarse conforme a estas normas (Página 1245) y el estado de certificación puede cambiar sin previo aviso. Es responsabilidad del usuario determinar las certificaciones aplicables consultando las clasificaciones marcadas en el producto.
- Información importante sobre cómo mantener la seguridad operacional de la planta (Página 5)

Sustituir una CPU V3.0 por una CPU V4.x.x

A la hora de sustituir una CPU S7-1200 V3.0 por una CPU S7-1200 V4.x.x, tenga en cuenta las diferencias (Página 1448) documentadas entre ambas versiones y las acciones que debe realizar el usuario.

Software de programación STEP 7

3

STEP 7 ofrece un entorno confortable que permite desarrollar, editar y observar la lógica del programa necesaria para controlar la aplicación, incluyendo herramientas para gestionar y configurar todos los dispositivos del proyecto, tales como controladores y dispositivos HMI. Para poder encontrar la información necesaria, STEP 7 ofrece un completo sistema de ayuda en pantalla.

STEP 7 proporciona lenguajes de programación estándar, que permiten desarrollar de forma cómoda y eficiente el programa de control.

- KOP (esquema de contactos) (Página 188) es un lenguaje de programación gráfico. Su representación es similar a los esquemas de circuitos.
- FUP (diagrama de funciones) (Página 189) es un lenguaje de programación que se basa en los símbolos lógicos gráficos empleados en el álgebra booleana.
- SCL (Structured Control Language) (Página 189) es un lenguaje de programación de alto nivel basado en texto.

Al crear un bloque lógico, se debe seleccionar el lenguaje de programación que empleará dicho bloque. El programa de usuario puede emplear bloques lógicos creados con cualquiera de los lenguajes de programación.

Nota

STEP 7 es el componente de software para programación y configuración de TIA Portal. TIA Portal, además de STEP 7, también contiene WinCC para el diseño y la ejecución de la visualización de procesos en runtime, con ayuda en pantalla para WinCC y STEP 7.

Las nuevas funciones de S7-1200 V4.5 requieren STEP 7 Professional V17.

3.1 Requisitos del sistema

Hay que instalar STEP 7 con derechos de administrador.

Tabla 3-1 Requisitos del sistema

Hardware/software	Requisitos
Tipo de procesador	Intel® Core™ i3-6100U, 2.30 GHz o mejor
RAM	8 GB
Espacio disponible en el disco duro	20 GB en la unidad de disco C:\

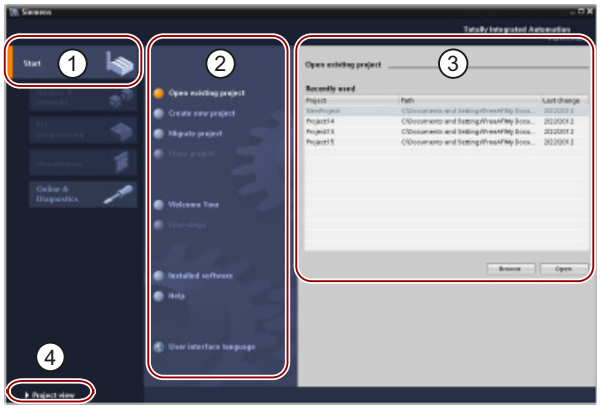
Hardware/software	Requisitos
Sistemas operativos	STEP 7 puede utilizarse con los siguientes sistemas operativos: • Windows 7 (64 bits): – Windows 7 Home Premium SP1 ** – Windows 7 Professional SP1 – Windows 7 Enterprise SP1 – Windows 7 Ultimate SP1 • Windows 10 (64 bits): – Windows 10 Home Version 1709 ** – 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 bits) – Windows Server 2012 R2 StdE (instalación completa) – Windows Server 2016 Standard (instalación completa)
Tarjeta gráfica	32 MB RAM Intensidad de color de 24 bits
Resolución de la pantalla	1024 x 768
Red	Ethernet de 100 Mbits/s o más rápido, para la comunicación entre STEP 7 y la CPU

* Incluidas todas las actualizaciones de seguridad relevantes. Para más información sobre los sistemas operativos, consulte la ayuda de Microsoft Windows o el sitio web de Microsoft.

** Solo para ediciones Basic

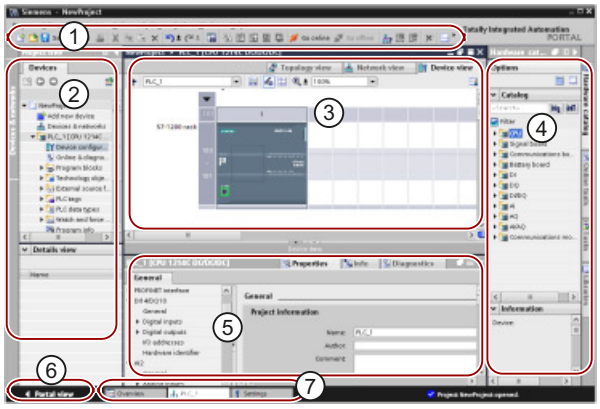
3.2 Diferentes vistas que facilitan el trabajo

STEP 7 proporciona un entorno de fácil manejo para programar la lógica del controlador, configurar la visualización de HMI y definir la comunicación por red. Para aumentar la productividad, STEP 7 ofrece dos vistas diferentes del proyecto, a saber: Distintos portales orientados a tareas y organizados según las funciones de las herramientas (vista del portal) o una vista orientada a los elementos del proyecto (vista del proyecto). El usuario puede seleccionar la vista que considere más apropiada para trabajar eficientemente. Con un solo clic es posible cambiar entre la vista del portal y la vista del proyecto.



Vista del portal

① Portales para las diferentes tareas
② Tareas del portal seleccionado
③ Panel de selección para la acción seleccionada
④ Cambia a la vista del proyecto



Vista del proyecto

① Menús y barra de herramientas
② Árbol del proyecto
③ Área de trabajo
④ Task Cards
⑤ Ventana de inspección
⑥ Cambia a la vista del portal
⑦ Barra del editor

Puesto que todos estos componentes se encuentran en un solo lugar, es posible acceder fácilmente a cualquier área del proyecto. El área de trabajo consta de tres pestañas:

- Vista de dispositivos: muestra el dispositivo que se ha agregado o seleccionado y los módulos que tiene conectados
- Vista de redes: muestra las CPU y las conexiones de la red
- Vista topológica: muestra la topología PROFINET de la red, incluidos los dispositivos, componentes pasivos, puertos, interconexiones y diagnósticos de puerto

Todas las vistas permiten realizar también tareas de configuración. La ventana de inspección muestra las propiedades e información acerca del objeto seleccionado en el área de trabajo. Si se seleccionan varios objetos, la ventana de inspección muestras las propiedades que pueden

configurarse. La ventana de inspección incluye fichas que permiten ver información de diagnóstico y otros mensajes.

La barra de editores agiliza el trabajo y mejora la eficiencia, ya que muestra todos los editores que están abiertos. Para cambiar entre los editores abiertos, basta con hacer clic sobre el editor en cuestión. También es posible visualizar dos editores simultáneamente, ya sea en mosaico vertical u horizontal. Esta función permite mover elementos entre los editores mediante Drag & Drop.

El sistema de información de STEP 7 dispone de una amplia Ayuda en pantalla para todas las herramientas de configuración, programación y vigilancia de STEP 7. Ofrece información más detallada en relación a lo explicado en el presente manual.

3.3 Compatibilidad entre STEP 7 y S7-1200

STEP 7 V17 soporta la configuración y programación de una CPU S7-1200 V4.5.

Es posible descargar proyectos para versiones anteriores de las CPU S7-1200 V4.x de STEP 7 V13 o posterior en una CPU S7-1200 V4.5. La configuración y el programa estarán limitados al conjunto de funciones e instrucciones compatibles con la versión anterior de la CPU S7-1200 y su versión de STEP 7.

Esta compatibilidad con versiones anteriores permite ejecutar en las CPU S7-1200 V4.5 programas que se habían diseñado y programado previamente para versiones anteriores.

⚠ ADVERTENCIA

Riesgos al copiar y pegar la lógica del programa de versiones anteriores de STEP 7

El hecho de copiar la lógica del programa de una versión anterior de STEP 7 puede causar un comportamiento impredecible en la ejecución del programa o errores de compilación. Las diferentes versiones de STEP 7 implementan los elementos del programa de forma distinta. El compilador no siempre detecta las diferencias si los cambios se realizan mediante copiar y pegar desde una versión anterior a STEP 7 V15. La ejecución de una lógica del programa impredecible puede causar la muerte o lesiones graves si no se corrige el programa.

Si emplea una lógica de programa de una versión anterior de STEP 7, actualice siempre todo el proyecto a la última versión de STEP 7. Después podrá copiar, cortar y pegar la lógica del programa según sea necesario. En la versión STEP 7 V16 y posteriores es posible abrir un proyecto de STEP 7 V 13 o posterior. Seguidamente, STEP 7 ejecuta las conversiones de compatibilidad necesarias y actualiza el programa de forma correcta. Estas actualizaciones y correcciones son necesarias para una ejecución y compilación debidas del programa. Si el proyecto es anterior a STEP 7 V13, deberá actualizarse progresivamente a STEP 7 V17 (Página 1447).

No es posible descargar proyectos para CPU S7-1200 V1.0, V2.0 o V3.0 en una CPU S7-1200 V4.x. Consulte el Sustitución de una CPU V3.0 por una CPU V4.x (Página 1448) tema sobre las directrices de actualización de proyectos antiguos a un proyecto que se pueda descargar.

Nota

Proyectos con versiones de CPU S7-1200 V1.x

En STEP 7 V15.1 no es posible abrir un proyecto STEP 7 que contenga CPU S7-1200 V1.x. Para utilizar el proyecto existente debe abrirlo con STEP 7 V13 (con cualquier actualización) y convertir las CPU S7-1200 V1.x a V2.0 o posteriores. Seguidamente, podrá emplear STEP 7 V15.1 para abrir el proyecto guardado con las CPU convertidas.

Compatibilidad con respecto a la comunicación segura

Las funciones de comunicación segura (Página 160) a partir de V4.5 de la CPU S7-1200 requieren STEP 7 V17 o posterior.

Puede darse el caso que utilice varias CPU S7-1200 V4.5 pero que no utilice STEP 7 V17.

Para más detalles sobre la comunicación entre una CPU S7-1200 V4.5, clientes y HMI, consulte el tema Activar la comunicación PG/PC y HMI segura y crear certificados (Página 166).

Consulte también

Nuevas funciones (Página 35)

Montaje

4

4.1 Directrices para montar dispositivos S7-1200

Los equipos S7-1200 son fáciles de montar. El S7-1200 puede montarse en un panel o en un raíl DIN, bien sea horizontal o verticalmente. El tamaño pequeño del S7-1200 permite ahorrar espacio.

Los estándares de equipos eléctricos clasifican el sistema SIMATIC S7-1200 como Equipo abierto. El S7-1200 debe montarse en una carcasa, un armario eléctrico o una sala de control. El acceso a la carcasa, el armario eléctrico o la sala de control debe limitarse a personal autorizado.

El S7-1200 debe instalarse en un entorno seco. Los circuitos SELV/PELV están diseñados para proporcionar protección contra choque eléctrico en entornos secos.

La instalación debe proporcionar la resistencia mecánica, la protección contra las llamas y la estabilidad homologada adecuadas para equipos abiertos en la categoría específica del lugar de instalación conforme a los reglamentos eléctricos y de edificaciones vigentes.

La contaminación conductora debida a polvo, humedad y polución aérea puede causar fallos funcionales y eléctricos en el PLC.

Si el PLC se ubica en una zona en la que puede haber contaminación conductora, habrá que proteger el PLC con una envolvente con grado de protección adecuado. IP54 es un grado utilizado generalmente para envolventes de equipos eléctricos en entornos sucios, y puede ser adecuado en su caso.

⚠ ADVERTENCIA

El montaje incorrecto del S7-1200 puede provocar fallos eléctricos o un funcionamiento inesperado de las máquinas.

Los fallos eléctricos y el funcionamiento inesperado pueden derivar en muerte, lesiones corporales graves y/o daños materiales.

Para garantizar que el equipo funciona de forma segura hay que seguir todas las instrucciones de montaje, así como del mantenimiento de un entorno de funcionamiento correcto.

Alejar los dispositivos S71200 de fuentes de calor, alta tensión e interferencias

Como regla general para la disposición de los dispositivos del sistema, los aparatos que generan altas tensiones e interferencias deben mantenerse siempre alejados de los equipos de baja tensión y de tipo lógico, tales como el S71200.

Al configurar la disposición del S7-1200 en el panel, se deben tener en cuenta los aparatos que generan calor y disponer los equipos electrónicos en las zonas más frías del armario eléctrico. Si se reduce la exposición a entornos de alta temperatura, aumentará la vida útil de cualquier dispositivo electrónico.

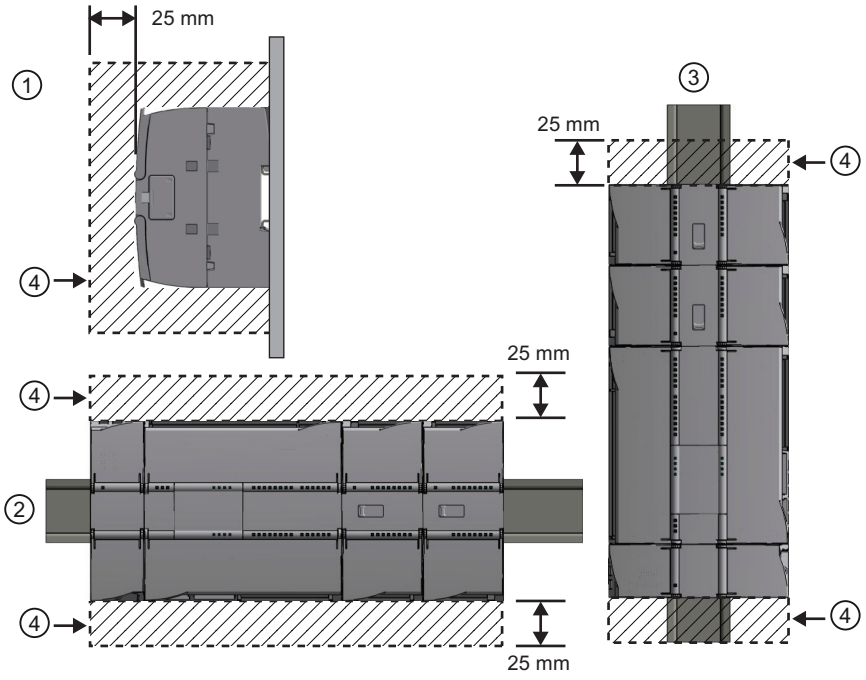
También se debe considerar la ruta del cableado de los dispositivos montados en el panel. Evite tender las líneas de señales de baja tensión y los cables de comunicación en un mismo canal junto con los cables AC y DC de alta energía y conmutación rápida.

Prever espacio suficiente para la refrigeración y el cableado

La refrigeración de los dispositivos S71200 se realiza por convección natural. Para la refrigeración correcta es preciso dejar un espacio mínimo de 25 mm por encima y por debajo de los dispositivos. Asimismo, se deben prever como mínimo 25 mm de profundidad entre el frente de los módulos y el interior de la carcasa.

⚠ **PRECAUCIÓN**
En el montaje vertical, la temperatura ambiente máxima admisible se reduce en 10 grados Celsius.
Las S7-1200 montadas en vertical deben orientarse de la forma mostrada en la figura.
Asegúrese de que el sistema S7-1200 se ha montado correctamente.

Al planificar la disposición del sistema S71200, prevea espacio suficiente para el cableado y la conexión de los cables de comunicación.



1 Vista lateral
2 Montaje horizontal
3 Montaje vertical
4 Espacio libre

4.2 Corriente necesaria

La CPU dispone de una fuente de alimentación interna que suministra energía eléctrica a la CPU, los módulos de señales, la Signal Board y los módulos de comunicación, así como otros consumidores de 24 V DC.

En los datos técnicos (Página 1245) encontrará más información sobre la corriente de 5 V DC que suministra la CPU y la corriente de 5 V DC que requieren los módulos de señales, las Signal Boards

y los módulos de comunicación. En "Cálculo de la corriente necesaria" (Página 1431) encontrará más información sobre cómo determinar cuánta energía (o corriente) puede suministrar la CPU a la configuración.

La CPU ofrece una alimentación de sensores de 24 V DC que suministra 24 V DC a las entradas y bobinas de relé de los módulos de señales, así como a otros consumidores. Si los requisitos de corriente de 24 V DC exceden la capacidad de alimentación de los sensores, es preciso añadir una fuente de alimentación externa de 24 V DC al sistema. En los datos técnicos (Página 1245) se indica la corriente necesaria para la alimentación de sensores de 24 V DC de las distintas CPU.

Si se requiere una fuente de alimentación externa de 24 V DC, vigile que no se conecte en paralelo con la alimentación de sensores de la CPU. Para aumentar la protección contra interferencias, se recomienda conectar los cables neutros (M) de las distintas fuentes de alimentación.

⚠ ADVERTENCIA

Si se conecta una fuente de alimentación externa de 24 V DC en paralelo con la fuente de alimentación 24 V DC de sensores, podría surgir un conflicto entre ambas fuentes, ya que cada una intenta establecer su propio nivel de tensión de salida.

Este conflicto puede reducir la vida útil u ocasionar la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema PLC. El funcionamiento imprevisible puede producir la muerte, lesiones corporales graves y/o daños materiales.

La fuente de alimentación DC de sensores y cualquier fuente de alimentación externa deben alimentar diferentes puntos.

Algunos puertos de entrada de la alimentación de 24 V DC del sistema S7-1200 están interconectados, teniendo un circuito lógico común que conecta varios bornes M. Por ejemplo, los circuitos siguientes están interconectados si no tienen aislamiento galvánico según las hojas de datos técnicos: la fuente de alimentación de 24 V DC de la CPU, la entrada de alimentación de la bobina de relé de un SM, o bien la fuente de alimentación de una entrada analógica no aislada. Todos los bornes M sin aislamiento galvánico deben conectarse al mismo potencial de referencia externo.

⚠ ADVERTENCIA

Si los bornes M sin aislamiento galvánico se conectan a diferentes potenciales de referencia, circularán corrientes indeseadas que podrían averiar o causar reacciones inesperadas en el PLC y los equipos conectados.

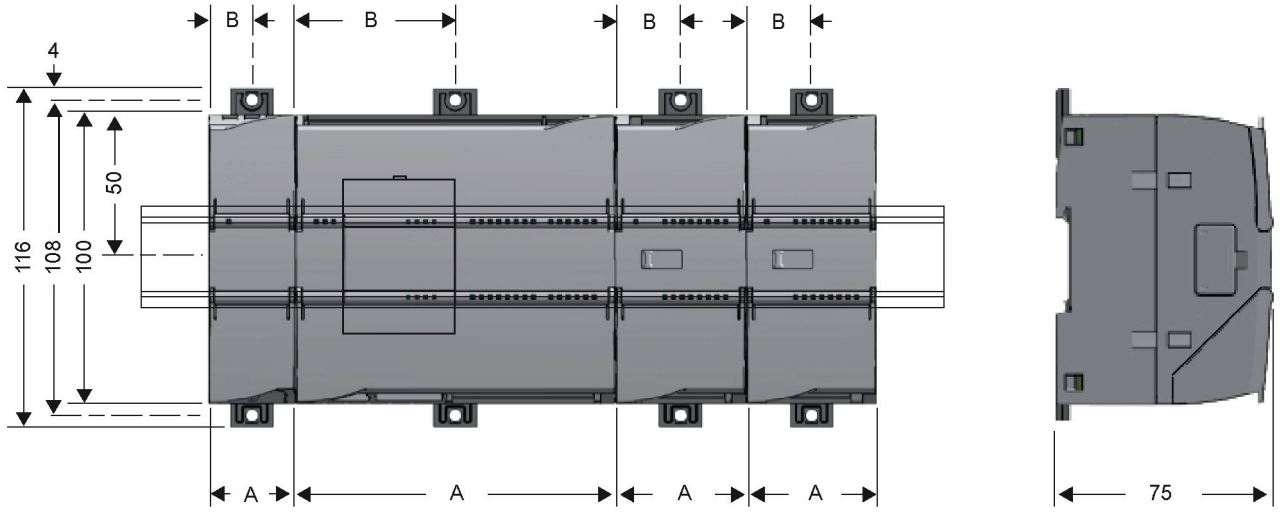
Si no se cumplen estas directrices, es posible que se produzcan averías o reacciones inesperadas que podrían causar la muerte, lesiones corporales graves y/o daños materiales.

Asegúrese que todos los bornes M sin aislamiento galvánico de un sistema S7-1200 están conectados al mismo potencial de referencia.

4.3 Procedimientos de montaje y desmontaje

4.3.1 Dimensiones de montaje de los dispositivos S7-1200

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



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

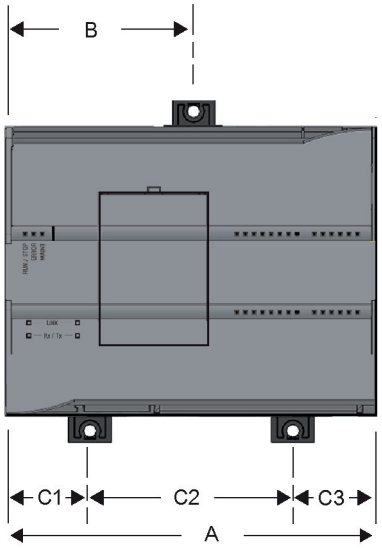


Tabla 4-1 Dimensiones de montaje (mm)

Dispositivos S7-1200		**Ancho A (mm)**	**Ancho B (mm)**	**Ancho C (mm)**
CPU	CPU 1211C y CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (parte superior)	Parte inferior: C1: 32,5 C2: 65 C3: 32,5
	CPU 1217C	150	75	Parte inferior: C1: 37,5 C2: 75 C3: 37,5
Módulos de señales	Digitales 8 y 16 puntos Analógicas 2, 4 y 8 puntos Termopar 4 y 8 puntos RTD 4 puntos Maestro SM 1278 IO-Link	45	22,5	--
	Digital DQ 8 x Relay (inversor)	70	35	--
	Analógico 16 puntos RTD 8 puntos	70	35	--
	SM 1238 Energy Meter module	45	22,5	--
Interfaces de comunicación	CM 1241 RS232 y CM 1241 RS422/485 CM 1243-5 PROFIBUS maestro y CM 1242-5 PROFIBUS esclavo 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) Adapter IE Advanced [1] TS (TeleService) Adapter IE Basic [1] TS Adapter Módulo TS	 30 30	 15 15	 -- --

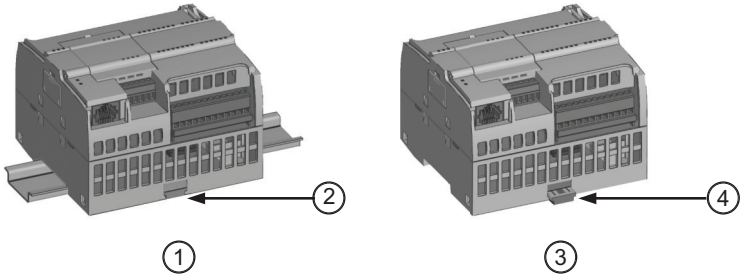
[1] Antes de instalar el TS (TeleService) Adapter IE Advanced o IE Basic hay que conectar el TS Adapter y un módulo TS. El ancho total ("ancho A") es de 60 mm.

Cada CPU, SM, CM y CP admite el montaje en un perfil DIN o en un panel. Utilice los clips del módulo previstos para el perfil DIN para fijar el dispositivo al perfil. Estos clips también pueden extenderse a otra posición para poder montar la unidad directamente en un panel. La dimensión interior del orificio para los clips de fijación en el dispositivo es 4,3 mm.

Es preciso prever una zona de disipación de 25 mm por encima y por debajo de la unidad para que el aire pueda circular libremente.

Montaje y desmontaje de dispositivos S7-1200

La CPU se puede montar fácilmente en un perfil estándar o en un panel. Los clips de fijación permiten fijar el dispositivo al perfil DIN. Estos clips también encajan en una posición extendida para proveer orificios de montaje que permiten montar el dispositivo directamente en un panel.



① Montaje en perfil DIN
② Clip de fijación al perfil enclavado
③ Montaje en panel
④ Clip de fijación en posición extendida para el montaje en panel

Antes de montar o desmontar cualquier dispositivo eléctrico, asegúrese que se ha desconectado la alimentación. Asegúrese también que está desconectada la alimentación eléctrica de todos los equipos conectados.

⚠ ADVERTENCIA

Si el S7-1200 o los dispositivos conectados se montan o desmontan estando conectada la alimentación, puede producirse un choque eléctrico o un funcionamiento inesperado de los dispositivos.

Si la alimentación del S7-1200 y de los dispositivos conectados no se desconecta por completo antes del montaje o desmontaje, podrían producirse la muerte, lesiones corporales graves y/o daños materiales debidos a choques eléctricos o al funcionamiento inesperado de los equipos.

Respete siempre las medidas de seguridad necesarias y asegúrese que la alimentación del S7-1200 está desconectada antes de montar o desmontar las CPU S7-1200 o los equipos conectados.

Al sustituir o montar un dispositivo S7-1200, vigile que se utilice siempre el módulo correcto o un dispositivo equivalente.

⚠ ADVERTENCIA

El montaje incorrecto de un módulo S7-1200 puede ocasionar el funcionamiento impredecible del programa del S7-1200.

Si un dispositivo S7-1200 no se sustituye por el mismo modelo o si no se monta con la orientación correcta y en el orden previsto, podrían producirse la muerte, lesiones corporales graves y/o daños materiales debido al funcionamiento inesperado del equipo.

Sustituya un dispositivo S7-1200 por el mismo modelo y móntelo con la orientación y posición correctas.

⚠ ADVERTENCIA

No desconecte ningún dispositivo en presencia de una atmósfera inflamable o combustible.

La desconexión de dispositivos en presencia de atmósferas inflamables o combustibles puede provocar un incendio o una explosión lo que puede producir la muerte, lesiones corporales graves y/o daños materiales.

Respete siempre las medidas de seguridad necesarias en presencia de una atmósfera inflamable o combustible.

Nota

Las descargas electroestáticas pueden deteriorar el dispositivo o la ranura para tarjetas en la CPU.

Cuando utilice el dispositivo, deberá estar en contacto con una superficie conductiva puesta a tierra o llevar puesta una pulsera puesta a tierra.

4.3.2 Montaje y desmontaje de la CPU

La CPU se puede montar en un panel o en un perfil DIN.

Nota

Conecte los módulos de comunicación necesarios a la CPU y monte el conjunto en forma de unidad. Los módulos de señales se montan por separado una vez montada la CPU.

Al montar las unidades en un perfil DIN o panel deben considerarse los siguientes puntos:

- Para el montaje en un raíl DIN, asegúrese de que el clip de fijación superior está en la posición enclavada (interior) y que el clip de fijación inferior está extendido, tanto en la CPU como en los CMs acoplados.
- Una vez montados los dispositivos en el perfil DIN, enclave los clips de sujeción para sujetar los dispositivos al raíl.
- Para el montaje en un panel, asegúrese de que los clips de fijación al raíl DIN están en posición extendida.

Para montar la CPU en un panel, proceda del siguiente modo:

1. Posicione y taladre los orificios de montaje (M4), según las dimensiones indicadas en la tabla Dimensiones de montaje (mm) (Página 46).
2. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica.
3. Extienda los clips de fijación del módulo. Asegúrese que los clips de fijación al perfil DIN en los lados superior e inferior de la CPU están en posición extendida.
4. Atornille el módulo al panel utilizando un tornillo M4 de cabeza alomada con una arandela elástica y otra plana. No utilice un tornillo de cabeza avellanada.

Nota

El tipo de tornillo viene determinado por el material en el que se monta. Aplique el par adecuado hasta que la arandela elástica quede plana. No aplique un par excesivo a los tornillos de montaje. No utilice un tornillo de cabeza avellanada.

Nota

Si la CPU se monta en un perfil normalizado en entornos donde se presenten vibraciones fuertes, o bien con orientación vertical, puede resultar necesario asegurarla con topes. Utilice una escuadra final (8WA1808 o 8WA1805) en el perfil DIN para que los módulos permanezcan conectados. Si el sistema se encuentra en un entorno donde se presenten vibraciones fuertes, es recomendable montar la CPU en un panel, puesto que este ofrece una mejor protección contra vibraciones.

Tabla 4-2 Instalar la CPU en un perfil DIN

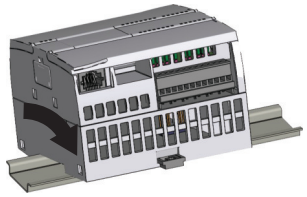
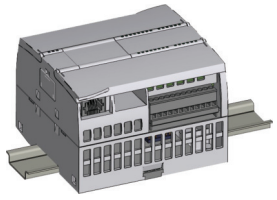
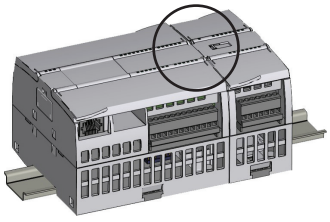
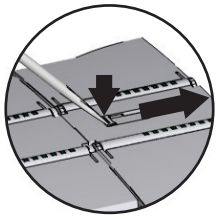
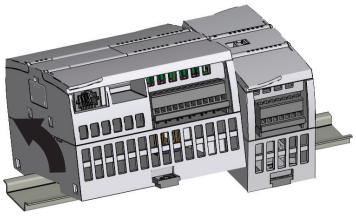
Tarea	Procedimiento
 	1. Monte el perfil DIN. Atornille el perfil al panel de montaje dejando un espacio de 75 mm entre tornillo y tornillo. 2. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 3. Enganche la CPU por el lado superior del perfil. 4. Extraiga el clip de fijación en el lado inferior de la CPU de manera que asome por encima del perfil. 5. Gire la CPU hacia abajo para posicionarla correctamente en el perfil. 6. Oprima los clips hasta que la CPU encaje en el perfil.

Tabla 4-3 Retirar la CPU de un perfil DIN

Tarea	Procedimiento
  	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Desconecte los conectores de E/S, el cableado y los cables restantes de la CPU (Página 56). 3. Desmonte la CPU y los módulos de comunicación conectados en forma de conjunto. Todos los módulos de señales deben permanecer montados. 4. Si hay un SM conectado a la CPU, retire el conector de bus: – Coloque un destornillador junto a la lengüeta en el lado superior del módulo de señales. – Oprima hacia abajo para desenclavar el conector de la CPU. – Desplace la lengüeta por completo hacia la derecha. 5. Desmonte la CPU: – Extraiga el clip de fijación para desenclavar la CPU del perfil DIN. – Gire la CPU hacia arriba, extráigala del perfil y retírela del sistema.

4.3.3 Montaje y desmontaje de SB, CB o BB

Tabla 4-4 Montaje de SB, CB o BB 1297

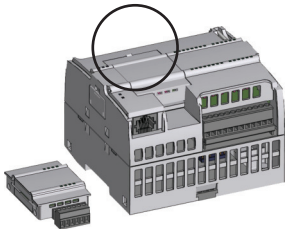
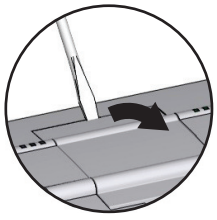
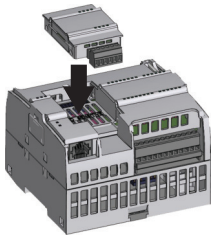
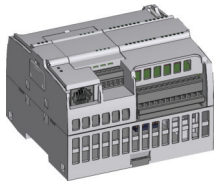
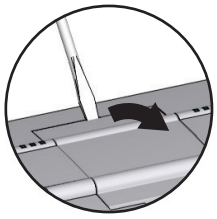
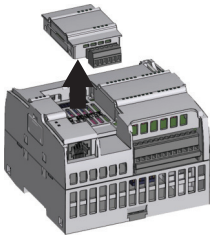
Tarea	Procedimiento
  	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Retire las cubiertas de bloque de terminales superior e inferior de la CPU. 3. Inserte un destornillador en la ranura arriba de la CPU en el lado posterior de la tapa. 4. Haga palanca suavemente para levantar la tapa y retírela de la CPU. 5. Coloque el módulo recto en su posición de montaje en el lado superior de la CPU. 6. Presione firmemente el módulo hasta que encaje en su posición. 7. Coloque nuevamente las tapas de los bloques de terminales.

Tabla 4-5 Desmontaje de SB, CB o BB 1297

Tarea	Procedimiento
  	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Retire las cubiertas de bloque de terminales superior e inferior de la CPU. 3. Retire cuidadosamente el conector de la Signal Board (si está instalado) soltándolo con un destornillador. 4. Inserte un destornillador en la ranura en el lado superior del módulo. 5. Haga palanca suavemente para desacoplar el módulo de la CPU. 6. Sin usar un destornillador, retire el módulo hacia arriba desde su posición montada en el lado superior de la CPU. 7. Vuelva a colocar la cubierta en la CPU. 8. Coloque nuevamente las tapas de los bloques de terminales.

Instalación o sustitución de la batería en la BB 1297

La BB 1297 requiere una batería de tipo CR1025. La batería no se suministra con la BB 1297 y debe adquirirse. Para instalar o sustituir la batería, proceda del siguiente modo:

1. En la BB 1297, instale una batería nueva con el lado positivo de la batería hacia arriba y el lado negativo junto a la tarjeta de circuito impreso.
2. La BB 1297 está lista para su instalación en la CPU. Asegúrese de que la CPU y todo el equipamiento del S7-1200 están desconectados de la tensión eléctrica y siga las instrucciones de instalación anteriores para instalar la BB 1297.

Para sustituir la batería de la BB 1297:

1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. Desenchufe la BB 1297 de la CPU según las instrucciones de desenchufe indicadas anteriormente.
2. Retire con cuidado la batería vieja con la ayuda de un destornillador pequeño. Presiones la batería para que salga de debajo del clip.
3. Instale una nueva batería de repuesto CR1025 con el lado positivo de la batería hacia arriba y el lado negativo junto a la tarjeta de circuito impreso.
4. Vuelva a instalar la BB 1297 según las instrucciones de instalación indicadas anteriormente.

⚠ ADVERTENCIA

La instalación de una batería sin especificar en la BB 1297 o la conexión de cualquier otro modo de una batería sin especificar en el circuito puede provocar un incendio o causar daños materiales y un comportamiento imprevisto de la maquinaria.

Un incendio o el comportamiento imprevisto de la maquinaria podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Utilice solo la batería CR1025 especificada para el respaldo del reloj en tiempo real.

4.3.4 Instalación y desmontaje de un SM

Tabla 4-6 Instalación de un SM

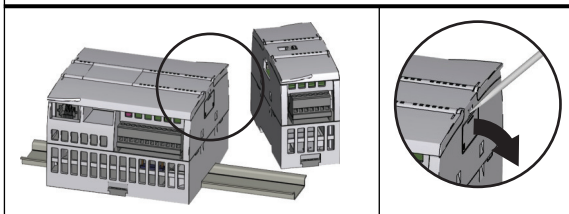
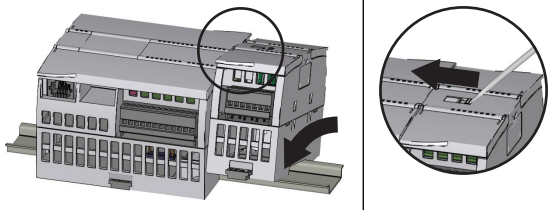
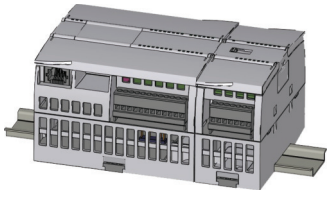
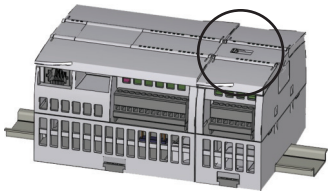
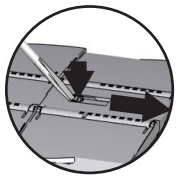
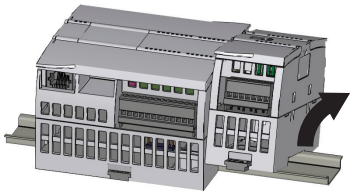
Tarea	Procedimiento
	El SM se monta una vez montada la CPU. 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Retire la tapa del conector en el lado derecho de la CPU: – Inserte un destornillador en la ranura arriba de la tapa. – Haga palanca suavemente en el lado superior de la tapa y retírela. 3. Guarde la tapa para poder reutilizarla.
	Conecte el SM a la CPU: 1. Coloque el SM junto a la CPU. 2. Enganche el SM por el lado superior del perfil DIN. 3. Extraiga el clip de fijación inferior para colocar el SM sobre el perfil. 4. Gire el SM hacia abajo hasta su posición junto a la CPU y oprima el clip de fijación inferior para enclavar el SM en el perfil.
	Al extender el conector de bus se crean las conexiones mecánicas y eléctricas para el SM. 1. Coloque un destornillador junto a la lengüeta en el lado superior del SM. 2. Desplace la lengüeta por completo hacia la izquierda para extender el conector de bus hacia la CPU. Siga el mismo procedimiento para montar un módulo de señales en otro módulo de señales.

Tabla 4-7 Desmontaje de un SM

Tarea	Procedimiento
	Cualquier SM se puede desmontar sin necesidad de desmontar la CPU u otros SMs. 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Desconecte los conectores de E/S y el cableado del SM (Página 56). 3. Retraiga el conector de bus. – Coloque un destornillador junto a la lengüeta en el lado superior del SM. – Oprima hacia abajo para desenclavar el conector de la CPU. – Desplace la lengüeta por completo hacia la derecha. Si hay otro SM en el lado derecho, repita este procedimiento para ese SM.
	
	Desmonte el SM: 1. Extraiga el clip de fijación inferior para desenclavar el SM del perfil DIN. 2. Gire el SM hacia arriba y extráigalo del perfil. Retire el SM del sistema. 3. En caso necesario, cubra el conector de bus de la CPU para impedir que se ensucie. Siga el mismo procedimiento para desmontar un módulo de señales de otro módulo de señales.

4.3.5 Montaje y desmontaje de un CM o CP

Conecte los módulos de comunicación necesarios a la CPU y monte el conjunto como una unidad, tal y como se muestra en Instalación y desmontaje de una CPU (Página 49).

Tabla 4-8 Instalación de un CM o CP

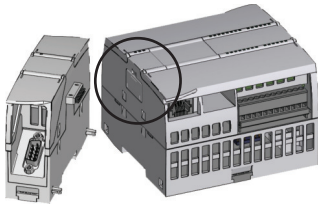
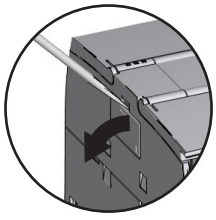
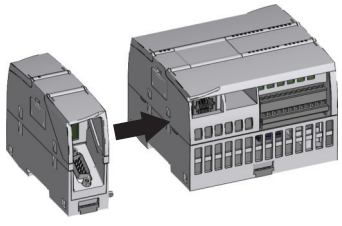
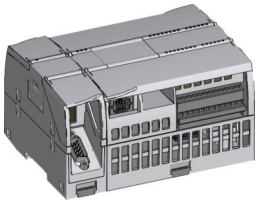
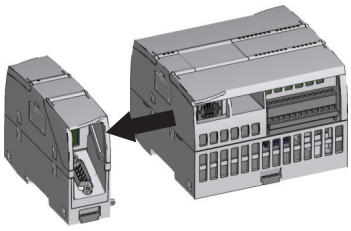
Tarea	Procedimiento
  	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Acople el CM a la CPU antes de montar el conjunto en forma de unidad en el perfil DIN o panel. 3. Retire la tapa de bus en el lado izquierdo de la CPU: – Inserte un destornillador en la ranura arriba de la tapa de bus. – Haga palanca suavemente en el lado superior de la tapa. 4. Retire la tapa de bus. Guarde la tapa para poder reutilizarla. 5. Conecte el CM o CP a la CPU: – Alinee el conector de bus y las clavijas del CM con los orificios de la CPU. – Empuje firmemente una unidad contra la otra hasta que encajen las clavijas. 6. Instale la CPU y el CP en un perfil DIN o panel.

Tabla 4-9 Desmontaje de un CM o CP

Tarea	Procedimiento
 	Desmonte la CPU y el CM en forma de unidad del raíl DIN o panel. 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Desconecte los conectores de E/S y retire el cableado y demás cables de la CPU y los CMs. 3. Para el montaje en un raíl DIN, extienda los clips de sujeción inferiores de la CPU y los CMs. 4. Desmonte la CPU y los CMs del raíl DIN o panel. 5. Sujete la CPU y los CMs firmemente con las manos y sepárelos.

ATENCIÓN
Separe los módulos sin usar una herramienta. No utilice herramientas para separar los módulos, puesto que podrían deteriorarse.

4.3.6 Extraer y reinsertar el conector del bloque de terminales del S7-1200

La CPU, la SB y los módulos SM incorporan conectores extraíbles que facilitan la conexión del cableado.

Tabla 4-10 Extracción del conector

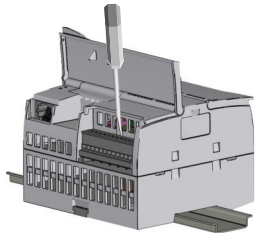
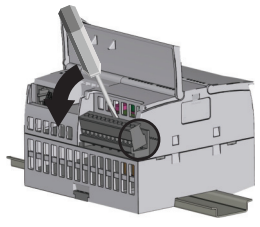
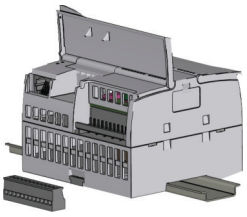
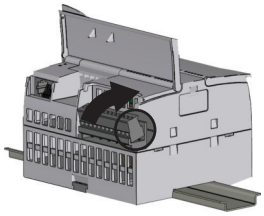
Tarea	Procedimiento
 	Prepare el sistema para la extracción del bloque de terminales desconectando la alimentación de la CPU y abriendo la tapa situada sobre el bloque de terminales. 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Busque la ranura para insertar la punta del destornillador en el lado superior del conector. 3. Inserte un destornillador en la ranura. 4. Haga palanca suavemente en el lado superior del conector para extraerlo de la CPU. El conector se desenclava audiblemente. 5. Sujete el conector con las manos y extráigalo de la CPU.

Tabla 4-11 Instalación del conector

Tarea	Procedimiento
 	Prepare los componentes para el montaje del bloque de terminales desconectando la alimentación de la CPU y abriendo la tapa para el conector. 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Alinee el conector a los pines del dispositivo. 3. Alinee el lado de cableado del conector en el zócalo. 4. Con un movimiento giratorio, empuje firmemente el conector hacia abajo hasta que encaje. Compruebe si el conector está bien alineado y encajado correctamente.

4.3.7 Instalación y desmontaje de un cable de ampliación

El cable de ampliación S7-1200 proporciona una flexibilidad adicional a la hora de configurar la estructura del sistema S7-1200. Solo se permite un cable de ampliación por sistema de CPU. El cable de ampliación se instala o bien entre la CPU y el primer SM, o bien entre dos SMs.

Tabla 4-12 Instalación y desmontaje del conector macho del cable de ampliación

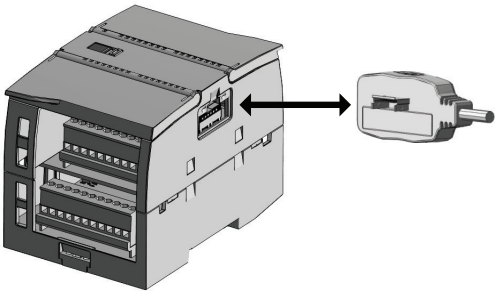
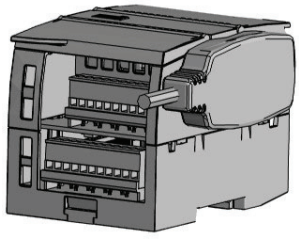
Tarea	Procedimiento
 	Para montar el conector macho: 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Presione el conector en el conector de bus del lado derecho del módulo de señales o la CPU. Para retirar el conector macho: 1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Tire del conector hembra para soltarlo del módulo de señales o la CPU.

Tabla 4-13 Montaje del conector hembra del cable de ampliación

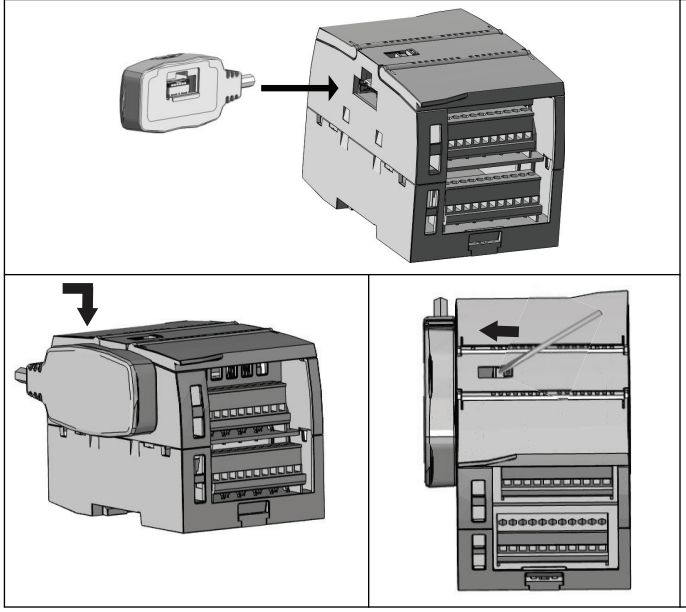
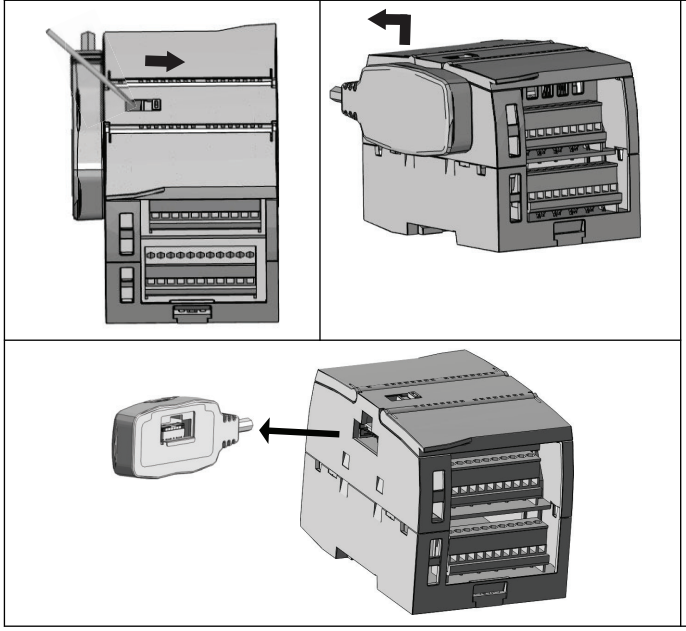
Tarea	Procedimiento
	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Coloque el conector hembra en el conector de bus del lado izquierdo del módulo de señales. 3. Deslice la extensión con gancho del conector hembra en el interior de la carcasa en el conector de bus y presione suavemente hacia abajo para encastrar el gancho. 4. Bloquee el conector en posición: – Coloque un destornillador junto a la lengüeta en el lado superior del módulo de señales. – Desplace la lengüeta por completo hacia la izquierda. Para encastrar el conector, deslice la lengüeta al máximo hacia la izquierda. La lengüeta debe quedar bloqueada en posición.

Tabla 4-14 Desmontaje del conector hembra del cable de ampliación

Tarea	Procedimiento
	1. Asegúrese de que la CPU y todo el equipamiento S7-1200 están desconectados de la tensión eléctrica. 2. Desbloquee el conector: – Coloque un destornillador junto a la lengüeta en el lado superior del módulo de señales. – Presione hacia abajo suavemente y deslice la lengüeta por completo hacia la derecha. 3. Levante el conector ligeramente para desenclavar la ampliación de gancho. 4. Retire el conector hembra.

Nota

Instalación del cable de ampliación en un entorno con vibraciones

Si el cable de ampliación está conectado a módulos en movimiento o que no están fijados firmemente, el extremo macho del cable puede aflojarse gradualmente.

Utilice una brida para fijar el cable del extremo macho en el perfil DIN (o cualquier otro lugar) para aliviar más la tensión.

Intente no forzar demasiado cuando tire del cable durante la instalación. Asegúrese de que la conexión entre cable y módulo esté en la posición correcta después de finalizar la instalación.

4.4 Directrices de cableado

La puesta a tierra y el cableado correctos de todos los equipos eléctricos es importante para garantizar el funcionamiento óptimo del sistema y aumentar la protección contra interferencias de la aplicación y del S7-1200. Encontrará los diagramas de cableado del S7-1200 en los datos técnicos (Página 1245).

Requisitos

Antes de poner a tierra o cablear cualquier dispositivo eléctrico, asegúrese que la alimentación está desconectada. Asegúrese también que está desconectada la alimentación eléctrica de todos los equipos conectados.

Vigile que se respeten todos los reglamentos eléctricos vinculantes al cablear el S7-1200 y los equipos conectados. El equipo se debe montar y operar conforme a todas las normas nacionales y locales vigentes. Contacte con las autoridades locales para determinar qué reglamentos y normas rigen en su caso específico.

⚠ ADVERTENCIA

Si el S7-1200 o los equipos conectados se montan o cablean estando conectada la alimentación, puede producirse un choque eléctrico o un funcionamiento inesperado de los equipos.

Si la alimentación del S7-1200 y de los equipos conectados no se desconecta por completo antes del montaje o desmontaje, pueden producirse la muerte, lesiones corporales graves y/o daños debidos a choques eléctricos o al funcionamiento inesperado de los equipos.

Respete siempre las medidas de seguridad necesarias y asegúrese que la alimentación eléctrica del S7-1200 está desconectada antes de montar o desmontar el S7-1200 o los equipos conectados.

Considere siempre los aspectos de seguridad al configurar la puesta a tierra y el cableado del sistema S7-1200. Los dispositivos de control electrónicos, tales como el S7-1200, pueden fallar y causar reacciones inesperadas de los equipos que se están controlando o vigilando. Por este

motivo, se recomienda prever medidas de seguridad independientes del S7-1200 para evitar lesiones corporales y/o daños materiales.

⚠ ADVERTENCIA

Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados.

Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Prevea dispositivos de parada de emergencia, dispositivos de protección electromecánicos y otras medidas redundantes de seguridad que sean independientes del S7-1200.

Directrices de aislamiento galvánico

Los límites de la alimentación AC del S7-1200 y de las E/S a los circuitos AC se han diseñado y aprobado para proveer un aislamiento galvánico seguro entre las tensiones de línea AC y los circuitos de baja tensión. Estos límites incluyen un aislamiento doble o reforzado, o bien un aislamiento básico más uno adicional, según las distintas normas. Los componentes que cruzan estos límites, tales como optoacopladores, condensadores, transformadores y relés se han aprobado, ya que proveen un aislamiento galvánico seguro. Solo los circuitos con una tensión nominal AC incluyen un aislamiento de seguridad frente a otros circuitos. Las barreras de aislamiento entre circuitos de 24 V DC son solamente funcionales y no deben tomarse para definir la seguridad.

La salida de la fuente de alimentación de sensores, los circuitos de comunicación y los circuitos lógicos internos de un S7-1200 con fuente de alimentación AC incluida tienen una fuente SELV (pequeña tensión de seguridad) conforme a EN 61131-2.

Para conservar el carácter seguro de los circuitos de baja tensión del S7-1200, las conexiones externas a puertos de comunicación, circuitos analógicos y todas las fuentes de alimentación nominales de 24 V DC y circuitos E/S deben ser alimentados por fuentes homologadas, que cumplan los requisitos de SELV, PELV, clase 2, tensión limitada o intensidad limitada, según distintas normas.

⚠ ADVERTENCIA

La utilización de fuentes de alimentación no aisladas o con aislamiento simple para abastecer los circuitos de baja tensión desde un conductor AC pueden causar tensiones peligrosas en circuitos considerados no peligrosos (seguros al tacto), tales como los circuitos de comunicación y el cableado de sensores de baja tensión.

Las altas tensiones inesperadas podrían causar choques eléctricos que pueden producir la muerte, lesiones corporales graves y/o daños materiales.

Utilice solo convertidores de alta a baja tensión aprobados como fuentes de circuitos de tensión limitada seguros al tacto.

Directrices de puesta a tierra del S7-1200

La mejor forma de poner a tierra la aplicación es garantizar que todos los conductores neutros y de masa del S7-1200 y de los equipos conectados se pongan a tierra en un mismo punto. Este punto debería conectarse directamente a la toma de tierra del sistema.

Todos los cables de puesta a tierra deberían tener la menor longitud posible y una sección grande, p. ej. 2 mm^2 (14 AWG).

Al definir físicamente las tierras es necesario considerar los requisitos de puesta a tierra de protección y el funcionamiento correcto de los dispositivos protectores.

Directrices de cableado del S7-1200

Al diseñar el cableado del S7-1200, prevea un interruptor unipolar para cortar simultáneamente la alimentación de la CPU S7-1200, de todos los circuitos de entrada y de todos los circuitos de salida. Prevea dispositivos de protección contra sobreintensidad (p. ej. fusibles o cortacircuitos) para limitar las corrientes de fallo en el cableado de alimentación. Para mayor protección es posible disponer un fusible u otro limitador de sobrecorriente en todos los circuitos de salida.

Utilice dispositivos de supresión de sobretensiones apropiados en el cableado sujeto a perturbaciones por descargas atmosféricas. Encontrará más información en Inmunidad a sobretensiones transitorias (Página 1245), en el apartado de datos técnicos generales.

Evite tender las líneas de señales de baja tensión y los cables de comunicación en una misma canaleta junto con los cables AC y los cables DC de alta energía y conmutación rápida. El tendido deberá efectuarse por pares; con el hilo de neutro o común combinado con el hilo de alimentación o de señal.

Utilice el cable más corto posible y vigile que tenga una sección suficiente para conducir la corriente necesaria.

Los hilos y los cables deben tener una temperatura nominal 30 °C superior a la temperatura ambiente alrededor del S7-1200 (por ejemplo, conductores de 85 °C como mínimo para una temperatura ambiente de 55 °C). Deben determinarse requisitos distintos para el material y el tipo de cable a partir de los valores nominales específicos del circuito eléctrico y del entorno de la instalación.

Utilice cables apantallados para obtener una protección óptima contra interferencias. Por lo general, los mejores resultados se obtienen poniendo a tierra la pantalla del S7-1200. Las pantallas de los cables de comunicación deben ponerse a tierra en las carcasas de los conectores de comunicación del S7-1200 utilizando conectores que se encastren en la pantalla del cable o bien uniendo las pantallas de los cables de comunicación a una tierra aparte. Las pantallas de los demás cables deben ponerse a tierra mediante terminales o cinta de cobre alrededor de la pantalla con el fin de ofrecer una conexión de gran superficie al punto de puesta a tierra.

Al cablear circuitos de entrada alimentados por una fuente externa, prevea dispositivos protectores contra sobrecorriente en estos circuitos. La protección externa no se requiere en los circuitos alimentados por la alimentación de sensores de 24 V DC del S7-1200, puesto que la alimentación de sensores ya está protegida contra sobrecorriente.

Todos los módulos S7-1200 incorporan conectores extraíbles para el cableado de usuario. Para evitar conexiones flojas, asegúrese que el conector está encajado correctamente y que el cable está insertado de forma segura en el conector.

Para impedir flujos de corriente indeseados en la instalación, el S7-1200 provee límites de aislamiento galvánico en ciertos puntos. Tenga en cuenta estos límites de aislamiento al

planificar el cableado del sistema. En los datos técnicos (Página 1307) encontrará más información acerca de la ubicación de los puntos de aislamiento galvánico y la capacidad que ofrecen. Los circuitos con una tensión nominal AC incluyen un aislamiento de seguridad respecto de otros circuitos. Las barreras de aislamiento entre circuitos de 24 V DC son solamente funcionales y no deben tomarse para definir la seguridad.

A continuación se muestra un resumen de las reglas de cableado para las CPU S7-1200 y los módulos SM y SB:

Tabla 4-15 Reglas de cableado para CPU S7-1200, SM y SB

Reglas de cableado para...	**Conector de CPU y SM**		**Conector de SB**
Sistema de conexión	**Inserción directa (push in)**	**Tornillo**	**Tornillo**
Secciones de conductor conectable para hilos estándar	De 2 mm^2 a 0,3 mm^2 (de 14 AWG a 22 AWG)		De 1,3 mm^2 a 0,3 mm^2 (de 16 AWG a 22 AWG)
Número de hilos por conexión	1 o combinación de 2 hilos en doble funda hasta 2 mm^2 (total)		1 o combinación de 2 hilos hasta 1,3 mm^2 (total)
Longitud de pelado del cable	Uso de fundas para una conexión eléctrica segura	6,4 mm	De 6,3 a 7 mm
Par de apriete* (máximo)	n/d	0,56 Nm	0,33 Nm
Herramienta	Destornillador de hoja plana de 2,5 a 3,0 mm de ancho		

* No apriete excesivamente los tornillos para impedir que se dañe el conector.

Nota

El uso de punteras o férulas en los conductores multifilares reduce el riesgo de cortocircuitos causados por hilos sueltos. Las punteras que sean más largas que la longitud de pelado recomendada deberán disponer de un collar aislante para impedir cortocircuitos provocados por el movimiento lateral de los conductores. Los límites de sección de los conductores pelados son válidos también para las punteras.

Consulte también

Datos técnicos (Página 1245)

Directrices para las cargas de lámparas

Las cargas de lámparas, incluidas las de los LED, pueden averiar los contactos de relé, debido a la elevada extracorriente momentánea de conexión. Esta extracorriente momentánea es nominalmente 10 a 15 veces superior a la corriente en régimen permanente de una lámpara de tungsteno. Se recomienda intercalar un relé sustituible o un limitador de sobretensión para las cargas de lámparas que deben conmutarse con frecuencia durante la vida útil de la aplicación.

Directrices relativas a las cargas inductivas

Utilice circuitos supresores con cargas inductivas para limitar el incremento de tensión producido al desactivarse las salidas. Los circuitos supresores protegen las salidas de averías prematuras causadas por crestas de alta tensión que se producen cuando se interrumpe el flujo de corriente que pasa por una carga inductiva.

Además, estos circuitos limitan las interferencias generadas al conmutar las cargas inductivas. Las interferencias de alta frecuencia provenientes de cargas inductivas con una mala supresión pueden perjudicar el funcionamiento del PLC. La manera más efectiva de reducir las interferencias es disponer un circuito supresor externo paralelo eléctricamente a la carga y ubicado físicamente cerca de la carga.

Las salidas DC del S7-1200 incluyen circuitos supresores internos adecuados para las cargas inductivas en la mayoría de las aplicaciones. Puesto que los contactos de salida de relé del S7-1200 pueden utilizarse para conmutar cargas tanto DC como AC, no proporcionan protección interna.

Una buena solución de supresión consiste en utilizar contactores y otras cargas inductivas para los que el fabricante proporcione circuitos supresores integrados en el dispositivo de carga o bien como accesorio opcional. Sin embargo, los circuitos supresores proporcionados por algunos fabricantes podrían no ser adecuados para su aplicación. En este caso, se necesitaría un circuito supresor adicional para reducir las interferencias de forma óptima y aumentar la vida útil de los contactos.

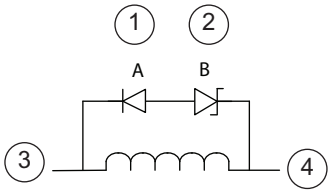
Para cargas AC puede utilizarse un varistor de óxido metálico (MOV) u otro dispositivo de fijación para tensión con un circuito RC paralelo, aunque no es efectivo cuando se utiliza solo. Un supresor MOV sin circuito RC paralelo suele provocar interferencias de alta frecuencia considerables hasta la tensión del borne.

Una cresta de tensión de desconexión bien controlada tendrá una frecuencia de anillo no superior a 10 kHz, aunque preferentemente inferior a 1 kHz. El voltaje máximo para líneas AC debería estar a +/- 1200 V de tierra. El voltaje máximo negativo para cargas DC que utilicen la supresión interna de PLC será de ~40 V por debajo de la tensión de alimentación de 24 V DC. La supresión externa debería limitar la cresta de tensión a 36 V de la alimentación para descargar la supresión interna.

Nota

La eficacia de un circuito supresor depende de la aplicación y debe verificarse para cada caso en particular. Asegúrese de que todos los componentes tienen el régimen correcto y utilice un osciloscopio para observar la cresta de tensión de desconexión.

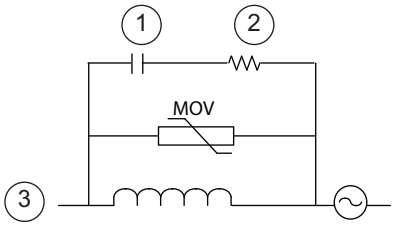
Circuito de supresión típico para salidas de relé o DC que conmutan cargas inductivas DC



① Diodo 1N4001 o equivalente
② Zener de 8,2 V (salidas DC)
Zener de 36 V (salidas de relé)
③ Salida
④ M, 24 V de referencia

En la mayoría de las aplicaciones es suficiente prever adicionalmente un diodo (A) paralelo a una carga inductiva DC. No obstante, si la aplicación requiere tiempos de desconexión más rápidos, se recomienda utilizar un diodo Zener (B). Vigile que el diodo Zener tenga suficiente capacidad para la cantidad de corriente en el circuito de salida.

Circuito de supresión típico para salidas de relé que conmutan cargas inductivas AC



① Consulte la tabla para el valor C
② Consulte la tabla para el valor R
③ Salida

Vigile que la tensión de empleo del varistor de óxido metálico (MOV) sea como mínimo un 20% superior a la tensión de línea nominal.

Elija resistores no inductivos con frecuencia de impulsos y condensadores recomendados para aplicaciones de impulsos (normalmente película metálica). Compruebe que los componentes cumplan los requisitos de potencia media, potencia máxima y tensión máxima.

Si diseña su propio circuito supresor, la tabla siguiente ofrece sugerencias para valores de resistores y condensadores para varias cargas AC. Los valores se basan en cálculos con parámetros de componentes ideales. "I rms" hace referencia en la tabla a la intensidad en estado estacionario de la carga cuando está completamente conectada.

Tabla 4-16 Valores de resistores y condensadores para circuitos supresores AC

Carga inductiva			**Valores de supresión**		
I rms	230 V AC	120 V AC	Resistor		Condensador
Amperios	VA	VA	Ω	W (potencia noominal)	nF
0,02	4,6	2,4	15000	0,1	15
0,05	11,5	6	5600	0,25	47
0,1	23	12	2700	0,5	100
0,2	46	24	1500	1	150
0,5	115	60	560	2,5	470

Carga inductiva			Valores de supresión		
1	230	120	270	5	1000
2	460	240	150	10	1500

Condiciones que cumplen los valores de la tabla:
Etapa de transición de desconexión máxima < 500 V
Voltaje máximo del resistor < 500 V
Voltaje máximo del condensador < 1250 V
Intensidad de supresión < 8% de la corriente de carga (50 Hz)
Intensidad de supresión < 11% de la corriente de carga (60 Hz)
Condensador dV/dt < 2 V/µs
Disipación de impulsos del condensador :$\int(dv/dt)^2\ dt < 10000\ V^2/\mu s$
Frecuencia resonante < 300 Hz
Potencia del resistor para una frecuencia de conmutación máx. de 2 Hz
Factor de potencia de 0,3 previsto para una carga inductiva típica

Directrices para las entradas y salidas diferenciales

Las entradas y salidas diferenciales se comportan de forma distinta a las entradas y salidas estándar. Hay dos pines por entrada y salida diferencial. Para determinar si una entrada o salida diferencial está activada o desactivada es necesario medir la diferencia de tensión entre estos dos pines.

Consulte las especificaciones detalladas de la CPU 1217C en el Apéndice A (Página 1307).

Principios básicos del PLC

5

5.1 Ejecución del programa de usuario

La CPU soporta los siguientes tipos de bloques lógicos que permiten estructurar eficientemente el programa de usuario:

- Los bloques de organización (OBs) definen la estructura del programa. Algunos OBs tienen reacciones y eventos de arranque predefinidos. No obstante, también es posible crear OBs con eventos de arranque personalizados.
- Las funciones (FCs) y los bloques de función (FBs) contienen el código de programa correspondiente a tareas específicas o combinaciones de parámetros. Cada FC o FB provee parámetros de entrada y salida para compartir datos con el bloque invocante. Un FB también utiliza un bloque de datos asociado (denominado DB de instancia) para conservar los valores de datos para la instancia de la llamada de FB. Puede llamar a un FB muchas veces, cada vez con un DB de instancia único. Las llamadas al mismo FB con DB de instancia distintos no afectan a los valores de datos del resto de los DB de instancia.
- Los bloques de datos (DBs) almacenan datos que pueden ser utilizados por los bloques del programa.

La ejecución del programa de usuario comienza con uno o varios bloques de organización (OB) de arranque que se ejecutan una vez al cambiar a estado operativo RUN, seguidos de uno o varios OB de ciclo que se ejecutan cíclicamente. También puede asociar un OB con un evento de alarma, que puede ser un evento estándar o un evento de error. Estos OB se ejecutan cuando se produce el evento estándar o de error correspondiente.

Una función (FC) o un bloque de función (FB) es un bloque de código del programa que puede llamarse desde un OB, o bien desde otra FC u otro FB. Son posibles las profundidades de anidamiento siguientes:

- 16 desde OB de ciclo o de arranque
- 6 desde cualquier OB de evento de alarma
  Nota: Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Las FC no están asociadas con ningún bloque de datos concreto (DB). Los FB están vinculados directamente a un DB que utilizan para transferir parámetros, así como para almacenar valores intermedios y resultados.

El tamaño del programa de usuario, los datos y la configuración está limitado por la memoria de carga disponible y la memoria de trabajo de la CPU. No hay un límite determinado para el número de cada bloque OB, FC, FB y DB individual. No obstante, el número total de bloques se limita a 1024.

En cada ciclo se escribe en las salidas, se leen las entradas, se ejecutan las instrucciones del programa de usuario y se realiza el procesamiento en segundo plano. En inglés, el ciclo también se llama "scan cycle" o "scan".

La solución de automatización S7-1200 puede constar de un rack central con la CPU S7-1200 y módulos adicionales. El término "rack central" hace referencia al montaje en perfil o en panel de

la CPU y los módulos asociados. Los módulos (SM, SB, BB, CB, CM o CP) solo se detectan e incorporan en el proceso de arranque.

- No está permitido enchufar o desenchufar un módulo del rack central con la alimentación conectada (en caliente). No inserte ni extraiga nunca un módulo del rack central cuando la CPU tenga tensión.

  ⚠ ADVERTENCIA

  Requisitos de seguridad para enchufar o desenchufar módulos

  Si no se desconecta la alimentación eléctrica de la CPU antes de enchufar o desenchufar un módulo (SM, SB, BB, CD, CM o CP) del rack central puede causar un comportamiento impredecible, que podría provocar la muerte o graves lesiones en personas y/o daños materiales.

  Desconecte siempre la alimentación de la CPU y del rack central y siga las medidas de seguridad pertinentes antes de enchufar o desenchufar un módulo del rack central.

- Una SIMATIC Memory Card sí puede enchufarse y desenchufarse mientras la CPU tiene tensión. Sin embargo, el enchufe o desenchufe de una Memory Card cuando la CPU está en RUN provoca el paso a STOP de la CPU.

  ATENCIÓN

  Riesgos asociados al desenchufe de la Memory Card cuando la CPU está en estado operativo RUN.

  El enchufe o desenchufe de una Memory Card mientras la CPU está en estado operativo RUN provoca el paso a STOP de la CPU, lo que podría causar daños en el equipo o en el proceso que se está controlando.

  Siempre que se inserta o extrae una Memory Card, la CPU pasa inmediatamente al estado operativo STOP. Antes de enchufar o desenchufar una Memory Card, asegúrese siempre de que la CPU no está controlando activamente una máquina o un proceso. Prevea siempre un circuito de parada de emergencia para la aplicación o el proceso.

- Si se inserta o extrae un módulo en un rack de E/S descentralizadas (AS-i, PROFINET o PROFIBUS) cuando la CPU está en estado operativo RUN, la CPU generará una entrada en el búfer de diagnóstico, ejecutará el OB de presencia de módulo, si existe, y permanecerá de forma predeterminada en estado operativo RUN.

Actualización de memorias imagen de proceso y memorias imagen parciales de proceso

La CPU actualiza las E/S digitales y analógicas locales de forma síncrona con el ciclo utilizando un área de memoria interna denominada memoria imagen de proceso. La memoria imagen de proceso contiene una instantánea de las entradas y salidas físicas (es decir, las E/S físicas de la CPU, de la Signal Board y de los módulos de señales).

Puede configurar E/S para que se actualicen en la memoria imagen de proceso en cada ciclo o cuando se genere una alarma de evento específica. También puede configurar una E/S para que se excluya de las actualizaciones de memorias imagen de proceso. Por ejemplo, el proceso puede necesitar solo algunos valores de datos cuando se produzca un evento, como por ejemplo una alarma de proceso. Al configurar la actualización de la memoria imagen de proceso para estas E/S que se van asociar con una memoria imagen parcial que asigne a un OB de alarma de

proceso, evitará que la CPU tenga que actualizar estos valores de datos innecesariamente durante cada ciclo cuando no sea necesario actualizar el proceso continuamente.

En el caso de E/S que se actualicen en cada ciclo, la CPU realizará las siguientes tareas durante cada ciclo:

- La CPU escribe las salidas desde la memoria imagen de proceso de las salidas en las salidas físicas.
- La CPU lee las entradas físicas inmediatamente antes de ejecutar el programa de usuario y almacena los valores de entrada en la memoria imagen de proceso de las entradas. De esta forma, estos valores mantendrán la coherencia durante la ejecución de las instrucciones del usuario.
- La CPU ejecuta la lógica de las instrucciones programadas y actualiza los valores de salida en la memoria imagen de proceso de las salidas, en vez de escribirlos en las salidas físicas reales. Este proceso ofrece una lógica coherente al ejecutar las instrucciones programadas durante un ciclo determinado y previene la fluctuación de las salidas físicas cuyo estado puede cambiar varias veces en la memoria imagen de proceso de las salidas.

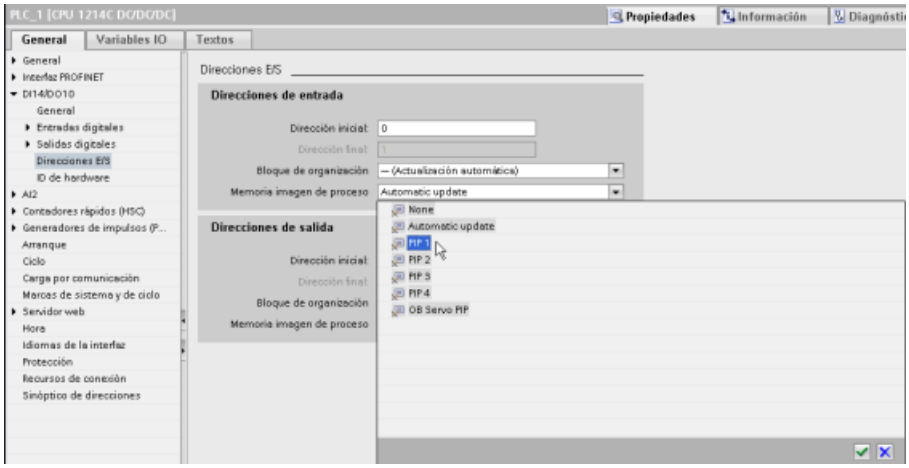
Para controlar si su proceso actualiza las entradas E/S automáticamente en cada ciclo o cuando se disparan eventos, S7-1200 proporcionará cinco memorias imagen parciales de proceso. La primera memoria imagen parcial de proceso, MIPP0, se ha designado para la E/S que se actualiza automáticamente en cada ciclo y es la asignación predeterminada. Puede usar las cuatro memorias imagen parciales de proceso restantes (MIPP1, MIPP2, MIPP3 e MIPP4) para asignar actualizaciones de memoria imagen de proceso de E/S a varios eventos de alarma. El usuario asigna E/S a memorias imagen parciales de proceso en Configuración de dispositivos y asigna memorias imagen parciales de proceso a eventos de alarma cuando crea OB de alarma (Página 178) o edita propiedades de OB (Página 178).

De forma predeterminada, cuando inserta un módulo en la vista de dispositivos, STEP 7 establece su actualización de memoria imagen de proceso de E/S en "Actualización automática". En el caso de las E/S configuradas para "Actualización automática", la CPU gestiona el intercambio de datos entre el módulo y el área de memoria imagen de proceso de forma automática durante cada ciclo.

Para asignar E/S digitales o analógicas a una memoria imagen parcial de proceso o para excluir las E/S de las actualizaciones de memoria imagen de proceso, proceda del siguiente modo:

1. Vea la ficha Propiedades de Configuración de dispositivos para acceder al dispositivo pertinente.
2. Amplíe las selecciones en "General", según sea necesario, para encontrar las E/S deseadas.
3. Seleccione "Direcciones E/S".

4. Seleccione opcionalmente un OB específico en la lista desplegable "Bloque de organización".
5. En la lista desplegable "Memoria imagen de proceso", cambie "Actualización automática" a "MIPP1", "MIPP2", "MIPP3", "MIPP4" o "Ninguna". Si selecciona "Ninguna" solo podrá leer o escribir en estas E/S utilizando instrucciones inmediatas. Para volver a incluir las E/S en la actualización automática de la memoria imagen de proceso, cambie de nuevo esta selección a "Actualización automática".



Es posible leer inmediatamente los valores de las entradas físicas y escribir inmediatamente los valores de las salidas físicas cuando se ejecuta una instrucción. Una lectura inmediata accede al estado actual de la entrada física y no actualiza la memoria imagen de proceso de las entradas, independientemente de si se ha configurado que la entrada se almacene en la memoria imagen de proceso. Una escritura inmediata en una salida física actualiza tanto el área de la memoria imagen de proceso de las salidas (si se ha configurado que la salida se almacene en la memoria imagen de proceso) y el punto de salida física. Añada el sufijo ":P" a la dirección E/S si desea que el programa acceda inmediatamente a los datos de E/S directamente desde el punto físico en lugar de utilizar la memoria imagen de proceso.

Nota

Uso de memorias imagen parciales de proceso

Si asigna E/S a una de las memorias imagen parciales de proceso MIPP1 a MIPP4 y no asigna un OB a esa memoria imagen parcial, la CPU nunca actualizará esas E/S a o desde la memoria imagen de proceso. La asignación de E/S a un IPP que no tenga una asignación de OB correspondiente es lo mismo que asignar la memoria imagen de proceso a "Ninguna". Puede leer las E/S directamente desde las E/S físicas con una instrucción de lectura inmediata o escribir en las E/S físicas con una instrucción de escritura inmediata. La CPU no actualiza la memoria imagen de proceso.

La CPU soporta periferia descentralizada para PROFINET, PROFIBUS y redes AS-i (Página 585).

5.1.1 Estados operativos de la CPU

La CPU tiene tres estados operativos, a saber: STOP, ARRANQUE y RUN. Los LEDs de estado en el frente de la CPU indican el estado operativo actual.

- En modo STOP, la CPU no ejecuta el programa. Se puede descargar un proyecto.
- En estado operativo ARRANQUE, los OB de arranque (si existen) se ejecutan una vez. La CPU no procesa eventos de alarma de proceso durante el estado operativo ARRANQUE.
- En estado operativo RUN, los OB de ciclos se ejecutan repetidamente. Pueden generarse eventos de alarma en cualquier punto durante el estado operativo RUN, lo que provoca la ejecución de los OB de evento de alarma correspondientes. Puede cargar algunas partes de un proyecto en estado operativo RUN (Página 1224).

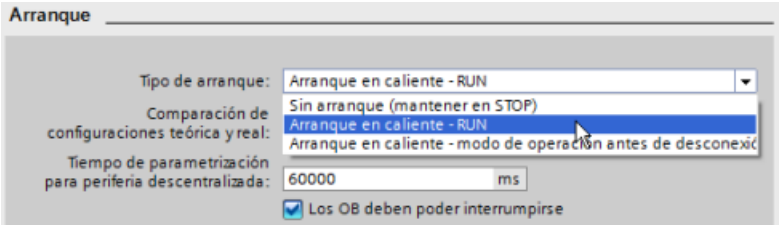
La CPU soporta el arranque en caliente para pasar al estado operativo RUN. El arranque en caliente no incluye la inicialización de la memoria. La CPU inicializa todos los datos de sistema no remanentes y los datos de usuario en un arranque en caliente y retiene los valores de todos los datos de usuario remanentes.

El borrado total borra toda la memoria de trabajo, así como las áreas de memoria remanentes y no remanentes. Además, copia la memoria de carga en la memoria de trabajo y establece las salidas en la "Reacción a STOP de la CPU" configurada. El borrado total no borra el búfer de diagnóstico ni tampoco los valores almacenados permanentemente de la dirección IP.

Se puede configurar el ajuste "arranque tras POWER ON" de la CPU. Este ajuste se encuentra en la "Configuración de dispositivos" de la CPU en "Arranque". Al arrancar, la CPU ejecuta una secuencia de pruebas de diagnóstico de arranque e inicialización del sistema. Durante la inicialización del sistema, la CPU borra toda el área de marcas (M) no remanente e inicializa todos los contenidos de DB no remanentes a los valores iniciales de la memoria de carga. La CPU retiene el área de marcas (M) remanente y los contenidos de DB remanentes y, a continuación,

entra en el estado operativo correspondiente. Determinados errores impiden que la CPU pase al estado operativo RUN. La CPU admite las siguientes opciones de configuración:

- Sin rearranque (permanecer en modo STOP)
- Arranque en caliente - RUN
- Arranque en caliente - modo previo a POWER OFF



ATENCIÓN

Los errores reparables pueden hacer que la CPU pase al estado operativo STOP.

La CPU puede pasar a STOP debido a errores reparables, como los siguientes:

- fallo de un módulo de señales sustituible
- fallos temporales, como perturbaciones en el cable de alimentación o eventos de arranque imprevisibles

Estas condiciones podrían provocar daños materiales.

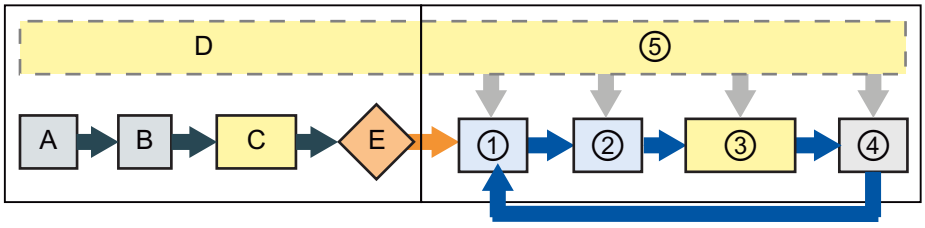
Si ha configurado la CPU a "Arranque en caliente - modo previo a POWER OFF", la CPU pasará al estado operativo que tenía antes de la pérdida de alimentación o del error. Si la CPU estaba en STOP cuando se produjo la pérdida de alimentación o el error, la CPU pasará al estado operativo STOP al arrancar. La CPU se mantendrá en ese estado hasta que reciba un comando para pasar al estado operativo RUN. Si la CPU estaba en RUN cuando se produjo la pérdida de alimentación o el error, la CPU pasará al estado operativo RUN en el siguiente arranque. La CPU pasará al estado operativo RUN siempre y cuando no detecte errores que impidan el paso al estado operativo RUN.

Configure las CPU que quiera que funcionen con independencia de una conexión STEP 7 con "Arranque en caliente - RUN". Este modo de arranque hace que la CPU vuelva al estado operativo RUN en el siguiente ciclo de desconexión/conexión (Power OFF - Power ON)

Se pueden utilizar los comandos "STOP" o "RUN" (Página 1210) de las herramientas online del software de programación para cambiar el modo de operación actual. También se puede

insertar una instrucción STP (Página 304) en el programa para cambiar la CPU a STOP. Esta instrucción permite detener la ejecución del programa en función de la lógica.

- En estado operativo STOP, la CPU procesa las órdenes de comunicación (según sea necesario) y realiza el autodiagnóstico. La CPU no ejecuta el programa de usuario. La memoria imagen de proceso no se actualiza automáticamente.
- En estado operativo ARRANQUE y RUN, la CPU ejecuta las tareas que muestra la figura siguiente:



ARRANQUE

A	Copia el estado de las entradas físicas en la memoria I
B	Inicializa el área de memoria (imagen) de salida Q bien a cero, al último valor, o bien al valor sustitutivo configurado. Pone a cero las salidas de PB, PN y AS-i.
C	Inicializa la memoria M no remanente y los bloques de datos a su valor inicial y habilita los eventos de alarma cíclica y de hora configurados. Ejecuta los OB de arranque.
D	Almacena los eventos de alarma de la cola de espera que deben procesarse una vez que se haya pasado al estado operativo RUN
E	Habilita la escritura de la memoria Q en las salidas físicas

RUN

①	Escribe la memoria Q en las salidas físicas
②	Copia el estado de las entradas físicas en la memoria I
③	Ejecuta los OBs de ciclo
④	Realiza autodiagnóstico
⑤	Procesa alarmas y comunicaciones en cualquier parte del ciclo

Nota

La comunicación, incluida la comunicación HMI, no puede interrumpir OB que no sean OB de ciclo de programa.

Procesamiento del ARRANQUE

Cada vez que el modo operativo cambia de STOP a RUN, la CPU borra las entradas de la memoria imagen de proceso, inicializa las salidas de la memoria imagen de proceso y procesa los OBs de arranque. En los accesos de lectura a las entradas de la memoria imagen de proceso realizados por instrucciones de los OBs de arranque se lee cero, en vez del valor actual de la entrada física. Por tanto, para leer el estado actual de una entrada física durante el estado operativo ARRANQUE, es preciso realizar una lectura inmediata. Los OB de arranque y cualquier FC y FB asociados se ejecutan después. Si existe más de un OB de arranque, la CPU ejecuta cada OB en el orden correspondiente al número de OB, comenzando con el número de OB más bajo.

Todo OB de arranque incluye información de arranque que ayuda a determinar la validez de los datos remanentes y el reloj en tiempo real. Es posible programar instrucciones dentro de los OBs de arranque para examinar estos valores de arranque y realizar las acciones apropiadas. Los OBs de arranque soportan las siguientes ubicaciones de arranque:

Tabla 5-1 Ubicaciones de arranque soportadas por el OB de arranque

Entrada	Tipo de datos	Descripción
LostRetentive	Bool	Este bit es verdadero (TRUE) si se han perdido las áreas de almacenamiento de datos remanentes
LostRTC	Bool	Este bit es verdadero (TRUE) si se ha perdido el reloj en tiempo real

La CPU también ejecuta las siguientes tareas durante el procesamiento del arranque:

- Las alarmas se ponen en cola de espera, pero no se procesan durante la fase de arranque
- El tiempo de ciclo no se vigila durante la fase de arranque
- La configuración de HSC (contadores rápidos), PWM (modulación del ancho de pulso) y módulos PtP (comunicación punto a punto) se puede modificar durante el arranque
- Los HSC, la PWM y los módulos de comunicación punto a punto solo funcionan en estado operativo RUN

Una vez finalizada la ejecución de los OBs de arranque, la CPU pasa a estado operativo RUN y procesa las tareas de control en un ciclo continuo.

5.1.2 Procesamiento del ciclo en estado operativo RUN

En cada ciclo, la CPU escribe en las salidas, lee las entradas, ejecuta el programa de usuario, actualiza los módulos de comunicación y reacciona a los eventos de alarma de usuario y peticiones de comunicación. Las peticiones de comunicación se procesan periódicamente durante el ciclo.

Estas acciones (excepto los eventos de alarma de usuario) se procesan con regularidad y en orden secuencial. Los eventos de alarma habilitados se procesan según su prioridad en el orden en que aparecen. En el caso de los eventos de alarma, la CPU lee las entradas, ejecuta el OB y, a continuación, escribe las salidas mediante la memoria imagen parcial del proceso asociada (IPP), si corresponde.

El sistema garantiza que el ciclo se procese dentro de un periodo denominado tiempo de ciclo máximo. De lo contrario, se generará un evento de error de tiempo.

- Todo ciclo comienza con la consulta de los valores actuales de las salidas digitales y analógicas de la memoria imagen de proceso. Estos valores se escriben luego en las salidas físicas de la CPU, la SB y los módulos SM configurados para la actualización automática de E/S (configuración predeterminada). Cuando una instrucción accede a una salida física, se actualizan tanto la memoria imagen de proceso de las salidas como la salida física.
- El ciclo continúa con la lectura de los valores actuales de las entradas digitales y analógicas de la CPU, la SB y los SMs configurados para la actualización automática de E/S (configuración predeterminada). Estos valores se escriben luego en la memoria imagen de proceso. Cuando una instrucción accede a una entrada física, se modifica el valor de ésta, pero no se actualiza la memoria imagen de proceso de las entradas.
- Tras leer las entradas, el programa de usuario se ejecuta desde la primera hasta la última instrucción. Esto incluye todos los OBs de ciclo, así como sus FCs y FBs asociados. Los OBs de ciclo se ejecutan en el orden correspondiente al número de OB, comenzando con el número de OB más bajo.

Las comunicaciones se procesan periódicamente durante todo el ciclo, siendo posible que se interrumpa la ejecución del programa de usuario.

El autodiagnóstico incluye comprobaciones periódicas del sistema y de estado de los módulos de E/S.

Las alarmas pueden ocurrir en cualquier parte del ciclo y son controladas por eventos. Cuando ocurre un evento, la CPU interrumpe el ciclo y llama el OB configurado para procesar ese evento. Una vez que el OB haya finalizado el procesamiento del evento, la CPU reanuda la ejecución del programa de usuario en el punto de interrupción.

5.1.3 Bloques de organización (OB)

Los OB controlan la ejecución del programa de usuario. Determinados eventos de la CPU disparan la ejecución de un bloque de organización. Los OB no pueden llamarse entre sí. Una FC o un FB no pueden llamar un OB. Solo un evento como una alarma de diagnóstico o un intervalo de tiempo pueden iniciar la ejecución de un OB. La CPU maneja los OB de acuerdo con sus respectivas clases de prioridad, es decir, ejecuta los OB de mayor prioridad antes que los de menor prioridad. La clase de prioridad más baja es 1 (para el ciclo de programa principal) y la más alta es 26.

5.1.3.1 OB de ciclo

Los OB de ciclo se ejecutan cíclicamente cuando la CPU se encuentra en estado operativo RUN. El bloque principal del programa es un OB de ciclo. Aquí es donde se almacenan las instrucciones que controlan el programa y donde se llaman bloques de usuario adicionales. Puede tener varios OB de ciclo, que la CPU ejecuta en orden numérico. El predeterminado es el "principal" (OB 1).

Eventos de ciclo

El evento de ciclo ocurre una vez por ciclo del programa. Durante el ciclo del programa, la CPU escribe en las salidas, lee las entradas y ejecuta los OB de ciclo. El evento de ciclo es necesario y siempre está habilitado. Es posible no tener OB de ciclo o bien tener varios OB seleccionados para el evento de ciclo. Una vez que se produzca el evento de ciclo, la CPU ejecuta el ciclo con el número más bajo (normalmente el "principal" OB1). La CPU ejecuta los demás OB de ciclo secuencialmente (en orden numérico) dentro del ciclo. La ejecución del programa es cíclica de forma que el evento de ciclo se produce en los siguientes momentos:

- Cuando el último OB de arranque finaliza la ejecución
- Cuando el último OB de ciclo finaliza la ejecución

Tabla 5-2 Información de arranque para un OB de ciclo

Entrada	Tipo de datos	Descripción
Initial_Call	Bool	Aplicable para la llamada inicial del OB.
Remanence	Bool	Aplicable si los datos remanentes están disponibles.

5.1.3.2 OB de arranque

Los OB de arranque se ejecutan una vez cuando el estado operativo de la CPU cambia de STOP a RUN, al arrancar en el estado operativo RUN y en una transición ordenada de STOP a RUN. Una vez finalizado, se comienza a ejecutar el "ciclo" principal.

Eventos de arranque

El evento de arranque ocurre una vez al producirse una transición de STOP a RUN y hace que la CPU ejecute los OB de arranque. Puede configurar varios OB para el evento de arranque. Los OB de arranque se ejecutan en orden numérico.

Tabla 5-3 Información de arranque para un OB de arranque

Entrada	Tipo de datos	Descripción
LostRetentive	Bool	Aplicable si los datos remanentes se han perdido.
LostRTC	Bool	Aplicable si se ha perdido la fecha y hora.

5.1.3.3 OB de alarma de retardo

Los OB de alarma de retardo se ejecutan después de un retardo que ha configurado.

Eventos de alarma de retardo

Los eventos de alarma de retardo se configuran para que ocurran cuando ha transcurrido un retardo especificado. El retardo se asigna con la instrucción SRT_DINT. Los eventos de alarma de retardo interrumpirán el ciclo con el fin de ejecutar el OB de alarma de retardo correspondiente. Solo puede asignar un OB de alarma de retardo a un evento de retardo. La CPU soporta cuatro eventos de retardo.

Tabla 5-4 Información de inicio para un OB de alarma de retardo

Entrada	Tipo de datos	Descripción
Sign	Word	Identificador transferido a llamada de disparo de SRT_DINT

5.1.3.4 OB de alarma cíclica

Los OB de alarma cíclica se ejecutan en intervalos periódicos. Puede configurar hasta un total de cuatro eventos de alarma cíclica con un OB correspondiente a cada evento de alarma cíclica.

Eventos de alarma cíclica

Los eventos de alarma cíclica permiten configurar la ejecución de un OB de alarma en un tiempo de ciclo configurado. El tiempo de ciclo inicial se configura al crear un OB de alarma cíclica. Un evento cíclico interrumpe el ciclo y ejecuta el OB de alarma cíclica correspondiente. Tenga en cuenta que el evento de alarma cíclica tiene una clase de prioridad más alta que el evento de ciclo.

Solo puede asignar un OB de alarma cíclica a un evento cíclico.

Puede asignar un desfase a cada alarma cíclica de forma que la ejecución de alarmas cíclicas pueda ejecutarse con un offset la una de la otra por la cantidad de desfase. Así, por ejemplo, si tiene un evento cíclico cada 5 ms y un evento cíclico cada 10 ms, cada diez milisegundos se producen ambos eventos en el mismo momento. Si desfasa el evento de 5 ms entre 1 y 4 ms y el evento de 10 ms en 0 ms, los dos eventos nunca sucederán en el mismo momento.

El desfase predeterminado es 0. Para modificar el desfase inicial o para cambiar el tiempo de ciclo inicial de un evento cíclico, proceda del siguiente modo:

1. Haga clic con el botón derecho del ratón en el OB de alarma cíclica del árbol del proyecto.
2. Seleccione "Propiedades" en el menú contextual.
3. Haga clic en "Alarma cíclica" desde el cuadro de diálogo "Alarma cíclica [OB 30]" e introduzca los nuevos valores iniciales.

El desfase máximo es de 6000 ms (6 segundos) o el tiempo de ciclo máximo, en función de cuál sea el menor de ambos.

También se puede consultar y modificar el tiempo de ciclo y el desfase desde el programa con las instrucciones de consulta de alarma cíclica (QRY_CINT) y ajuste de alarma cíclica (SET_CINT). Los valores de tiempo de ciclo y desfase definidos en la instrucción SET_CINT no se conservan tras desconectar y conectar la alimentación, o tras pasar a estado operativo STOP; los valores de tiempo de ciclo y desfase vuelven a los valores iniciales tras desconectar y conectar la alimentación, o tras pasar a estado operativo STOP. La CPU admite un total de cuatro eventos de alarma cíclica.

5.1.3.5 OB de alarma de proceso

Los OB de alarma de proceso se ejecutan cuando se produce el evento de hardware pertinente. Los OB de alarma de proceso interrumpen la ejecución cíclica normal del programa como reacción a una señal de un evento de hardware.

Eventos de alarma de proceso

Los cambios en el hardware, como por ejemplo un flanco ascendente o descendente en una entrada o bien un evento de contador rápido (HSC), disparan los eventos de alarma de proceso. El S7-1200 admite un OB de alarma para cada evento de alarma de proceso. Los eventos de hardware se habilitan en la configuración de dispositivos y se asigna un OB para un evento en la configuración de dispositivos o con una instrucción ATTACH en el programa de usuario. La CPU soporta varios eventos de alarma de proceso. El modelo de CPU y el número de entradas determinan los eventos exactos que están disponibles.

Los límites para los eventos de alarma de proceso son los siguientes:

Flancos:

- Eventos de flanco ascendente: 16 como máximo
- Eventos de flanco descendente: 16 como máximo

Eventos de HSC:

- CV=PV: 6 como máximo
- Cambio de sentido: 6 como máximo
- Inicialización externa: 6 como máximo

Tabla 5-5 Información de arranque para un OB de alarma de proceso

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	La ID de hardware del módulo que ha disparado la alarma de proceso
USI	WORD	Identificador de estructura de usuario (16#0001 a 16#FFFF), reservado para usos futuros
IChannel	USINT	Número del canal que ha disparado la alarma
EventType	BYTE	Identificador del tipo de evento específico del módulo y asociado al evento que ha disparado la alarma, por ejemplo un flanco ascendente.

Los bits de EventType dependen del módulo causante de la alarma, tal como se muestra a continuación:

Módulo / submódulo	Valor	Evento de proceso
E/S integradas de CPU o SB	16#0	Flanco ascendente
	16#1	Flanco descendente
HSC	16#0	HSC CV=RV1
	16#1	Cambio de sentido en HSC
	16#2	Reset de HSC
	16#3	HSC CV=RV2

5.1.3.6 OB de alarma de error de tiempo

Si se ha configurado, el OB de alarma de error de tiempo (OB 80) se ejecuta cuando el ciclo supera el tiempo de ciclo máximo o cuando se produce un evento de error de tiempo. Si se dispara, se ejecuta e interrumpe la ejecución cíclica normal del programa o cualquier otro OB de evento.

La aparición de cualquiera de esos eventos genera una entrada en el búfer de diagnóstico que describe el evento. La entrada del búfer de diagnóstico se genera independientemente de la existencia del OB de alarma de error de tiempo.

Eventos de alarma de error de tiempo

La aparición de cualquiera de las condiciones de error de tiempo diferentes causa un evento de error de tiempo:

- El ciclo rebasa el tiempo de ciclo máximo.
  El "error de rebase del tiempo de ciclo máximo" ocurre si el ciclo no finaliza dentro del tiempo de ciclo máximo especificado. Para más información sobre la condición de tiempo de ciclo máximo, cómo configurar el tiempo máximo de ciclo en las propiedades de la CPU y cómo reiniciar el tiempo de ciclo consulte el apartado "Vigilancia y configuración del tiempo de ciclo" (Página 91).
- La CPU no puede iniciar el OB solicitado porque se inicia una segunda alarma de tiempo (cíclica o de retardo) antes de que la CPU acabe de ejecutar el primer OB de alarma.
- Desbordamiento de la cola de espera
  La produce una condición "Desbordamiento de la cola de espera" si las alarmas ocurren más rápidamente de lo que la CPU puede procesarlas. La CPU limita el número de eventos pendientes (en cola de espera) utilizando una cola diferente para cada tipo de evento. Si ocurre un evento estando llena la cola de espera correspondiente, la CPU genera un evento de error de tiempo.

Todos los eventos de error de tiempo disparan la ejecución del OB de alarma de error de tiempo (si existe). Si el OB de alarma de error de tiempo no existe, la CPU pasa al estado operativo STOP.

El programa de usuario puede prolongar el tiempo de ejecución del ciclo hasta diez veces el tiempo de ciclo máximo configurado ejecutando la instrucción RE_TRIGR (Página 303) para reiniciar la vigilancia del tiempo de ciclo. Sin embargo, si el tiempo de ciclo máximo se rebasa dos veces en un mismo ciclo sin que se inicialice el temporizador de vigilancia del ciclo, la CPU pasará a STOP, independientemente de si existe el OB de alarma de error de tiempo. Consulte el apartado "Vigilancia del tiempo de ciclo" (Página 91) del manual de sistema S7-1200.

El OB de alarma de error de tiempo incluye información de arranque que permite determinar qué evento y OB han generado el error de tiempo. Puede programar instrucciones dentro del OB para examinar estos valores de arranque y realizar las acciones apropiadas.

Tabla 5-6 Información de arranque para el OB de error de tiempo (OB 80)

Entrada	Tipo de datos	Descripción
fault_id	BYTE	16#01 - rebase del tiempo de ciclo máximo 16#02 - no se puede iniciar el OB solicitado 16#07 y 16#09 - desbordamiento de la cola de espera
csg_OBnr	OB_ANY	Número de OB que se estaba ejecutando cuando ocurrió el error
csg_prio	UINT	Prioridad del OB que ha causado el error

Para incluir un OB de alarma de error de tiempo en el proyecto, debe añadir una alarma de error de tiempo haciendo doble clic en "Agregar nuevo bloque" en "Bloques de programa" y luego elegir "Bloque de organización" y "Time error interrupt" (Alarmas de error de tiempo).

La prioridad de una nueva CPU V4.0 es 22. Si se sustituye una CPU V3.0 por una CPU V4.0 (Página 1448), la prioridad será 26, que es la prioridad que era efectiva para V3.0. En ambos casos, el campo de prioridad es editable, de modo que la prioridad se puede ajustar a cualquier valor entre 22 y 26.

5.1.3.7 OB de alarma de error de diagnóstico

El OB de alarma de error de diagnóstico se ejecuta cuando la CPU detecta un error de diagnóstico o si un módulo apto para el diagnóstico detecta un error y el usuario ha habilitado la alarma de error de diagnóstico para el módulo. El OB de alarma de error de diagnóstico interrumpe la ejecución cíclica normal del programa. Puede incluir una instrucción STP en el OB de alarma de error de diagnóstico para poner la CPU en estado operativo STOP si desea que la CPU pase a este estado operativo cuando reciba este tipo de error.

Si no incluye un OB de alarma de error de diagnóstico en el programa, la CPU ignorará el error y se mantendrá en estado operativo RUN.

Eventos de error de diagnóstico

Los dispositivos analógicos (locales), PROFINET, PROFIBUS y algunos dispositivos digitales (locales) pueden detectar y notificar errores de diagnóstico. La aparición o eliminación de cualquiera de las diferentes condiciones de error de diagnóstico ocasiona un evento de error de diagnóstico. Se soportan los siguientes errores de diagnóstico:

- Falta alimentación externa
- Límite alto excedido
- Límite bajo excedido
- Rotura de hilo
- Cortocircuito

Los eventos de error de diagnóstico disparan la ejecución del OB de alarma de error de diagnóstico (OB 82), si existe. Si no existe, la CPU ignora el error.

Para incluir un OB de alarma de error de diagnóstico en el proyecto, debe añadir una alarma de error de diagnóstico haciendo doble clic en "Agregar nuevo bloque" en "Bloques de programa" y luego elegir "Bloque de organización" y "Diagnostic error interrupt" (Alarmas de error de diagnóstico).

Nota

Errores de diagnóstico para dispositivos analógicos locales multicanal (E/S, RTD y termopar)

El OB de error de diagnóstico no puede procesar más de un error de diagnóstico de canal al mismo tiempo.

Si dos canales de un dispositivo multicanal tienen un error, el segundo error solo dispara el OB de alarma de error de diagnóstico en las condiciones siguientes: el primer error de canal se borra, la ejecución del OB de alarma de error de diagnóstico que ha disparado el primer error ha finalizado y el segundo error persiste.

El OB de alarma de error de diagnóstico incluye información de arranque que ayuda a determinar si el evento se debe a la aparición o desaparición de un error, así como el dispositivo y canal que han notificado el error. Puede programar instrucciones dentro del OB de alarma de error de diagnóstico para examinar estos valores de arranque y realizar las acciones apropiadas.

Nota

La información de arranque del OB de alarma de diagnóstico se refiere al submódulo como un todo si no hay eventos de diagnóstico pendientes

En V3.0, la información de arranque de un error de diagnóstico saliente indicaba siempre el origen del evento. En V4.0, si el evento sale del submódulo sin diagnósticos pendientes, la información de arranque se referirá al submódulo como un todo (16#8000), incluso si el origen del evento es un canal específico.

Así, por ejemplo, si una rotura de hilo provoca un error de diagnóstico en el canal 2, después el fallo se corrige y el error de diagnóstico se acusa, entonces la información de arranque no hará referencia al canal 2 sino al submódulo (16#8000).

Tabla 5-7 Información de arranque para el OB de alarma de error de diagnóstico

Entrada	Tipo de datos	Descripción
IOstate	WORD	Estado de E/S del dispositivo: • Bit 0 = 1 si la configuración es correcta, y = 0 si la configuración ya no es correcta. • El bit 4 es 1 si existe un error (p. ej. una rotura de hilo). (Bit 4 = 0 si no hay ningún error.) • Bit 5 = 1 si la configuración **no** es correcta, y = 0 si la configuración vuelve a ser correcta. • Bit 7 = 1 si se ha producido un error de acceso a E/S. Véase LADDR para conocer el identificador de hardware de E/S con error de acceso. (Bit 6 = 0 si no hay ningún error.)
LADDR	HW_ANY	ID de hardware del dispositivo o unidad funcional que ha notificado el error[1]
Channel	UINT	Número de canal
MultiError	BOOL	TRUE (verdadero) si ha ocurrido más de un error

[1] La entrada en LADDR contiene el identificador de hardware del dispositivo o unidad funcional que ha devuelto el error. El identificador de hardware se asigna automáticamente cuando se insertan componentes en la vista de dispositivos o redes, y aparece en la ficha "Constantes" de "Variables PLC". También se asigna automáticamente un nombre al identificador de hardware. Estas entradas de la ficha "Constantes" de "Variables PLC" no se pueden modificar.

5.1.3.8 OB de presencia de módulo

El OB de presencia de módulo se ejecuta cuando un módulo o submódulo de E/S descentralizadas y no desactivado (PROFIBUS, PROFINET, AS-i) genera un evento relacionado con la inserción o extracción de un módulo.

Evento de presencia de módulo

Las siguientes condiciones generan un evento de presencia de módulo:

- Alguien extrae o inserta un módulo configurado.
- Un módulo configurado no está físicamente presente en un rack de ampliación.
- Hay un módulo incompatible en un rack de ampliación que no corresponde al módulo configurado.
- Un módulo compatible para un módulo configurado es un rack de ampliación, pero la configuración no permite los módulos de reserva.
- Un módulo o submódulo tiene errores de parametrización

Si no se ha programado este OB, la CPU permanecerá en el estado operativo RUN cuando se produzca cualquiera de estas condiciones con un módulo E/S descentralizado configurado y no desactivado.

Independientemente de si ha programado este OB, la CPU cambia al estado operativo STOP cuando se cumple alguna de estas condiciones con un módulo en el rack central.

Tabla 5-8 Información de arranque para el OB de presencia de módulo

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	Identificador de hardware
Event_Class	Byte	16#38: módulo insertado 16#29: módulo extraído
Fault_ID	Byte	Identificador de errores

5.1.3.9 OB de fallo del rack o estación

El OB de "Fallo del rack o estación" se ejecuta cuando la CPU detecta el fallo o la pérdida de comunicación de un rack o estación descentralizados.

Evento de fallo del rack o estación

La CPU genera un evento de fallo de rack o estación cuando detecta una de las situaciones siguientes:

- El fallo de un sistema maestro DP o de un sistema PROFINET IO (en caso de un evento entrante o saliente).
- El fallo de un esclavo DP o de un dispositivo IO (en caso de un evento entrante o saliente).
- El fallo de parte de los submódulos o de un PROFINET I-device.

Si no se ha programado este OB, la CPU permanece en el estado operativo RUN cuando se cumple alguna de estas condiciones.

Tabla 5-9 Información de arranque para el OB de fallo del rack o estación

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	Identificador de hardware
Event_Class	Byte	16#38: evento saliente 16#39: evento entrante
Fault_ID	Byte	Identificador de errores

5.1.3.10 OB de hora

Los OB de hora se ejecutan en función de las condiciones horarias configuradas. La CPU admite dos OB de hora.

Eventos de hora

Puede configurar un evento de alarma horaria para que ocurra una vez en una fecha u hora especificada o bien cíclicamente con uno de los ciclos siguientes:

- Cada minuto: La alarma se produce cada minuto.
- Cada hora: La alarma se produce cada hora.
- Diariamente: La alarma se produce cada día en una fecha especificada (hora o minuto).
- Semanal: La alarma se produce cada semana a una hora determinada de un día especificado de la semana (por ejemplo, cada martes a las 4:30 de la tarde).
- Mensual: La alarma se produce cada mes a una hora determinada de un día especificado del mes. El día debe estar comprendido entre el 1 y el 28, ambos incluidos.
- Cada final de mes: La alarma se produce el último día de cada mes a una hora especificada.
- Anual: La alarma se produce cada año en la fecha especificada (mes y día). No puede especificar el 29 de febrero como fecha.

Tabla 5-10 Información de arranque para un OB de evento de hora

Entrada	Tipo de datos	Descripción
CaughtUp	Bool	Se captura la llamada de OB porque se adelantó la fecha/hora.
SecondTimes	Bool	Se ha iniciado la llamada de OB por segunda vez porque se ha retrasado la fecha/hora.

5.1.3.11 OB de estado

Los OB de estado se ejecutan si un esclavo DPV1 o PNIO dispara una alarma de estado. Este puede ser el caso si un componente (módulo o rack) de un esclavo DPV1 o PNIO cambia su estado operativo, por ejemplo de RUN a STOP.

Eventos de estado

Para obtener información detallada sobre los eventos que disparan una alarma de estado, consulte la documentación del fabricante con relación al esclavo DPV1 o PNIO.

Tabla 5-11 Información de arranque para un OB de estado

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	Identificador de hardware
Slot	UInt	Número de slot
Specifier	Word	Indicador de alarma

5.1.3.12 OB de actualización

Los OB de actualización se ejecutan si un esclavo DPV1 o PNIO dispara una alarma de actualización.

Eventos de actualización

Para obtener información detallada sobre los eventos que disparan una alarma de actualización, consulte la documentación del fabricante con relación al esclavo DPV1 o PNIO.

Tabla 5-12 Información de arranque para un OB de actualización

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	Identificador de hardware
Slot	UInt	Número de slot
Specifier	Word	Indicador de alarma

5.1.3.13 OB de perfil

Los OB de perfil se ejecutan si un esclavo DPV1 o PNIO dispara una alarma específica de perfil.

Eventos de perfil

Para obtener información detallada sobre los eventos que disparan una alarma de perfil, consulte la documentación del fabricante con relación al esclavo DPV1 o PNIO.

Tabla 5-13 Información de arranque para un OB de perfil

Entrada	Tipo de datos	Descripción
LADDR	HW_IO	Identificador de hardware
Slot	UInt	Número de slot
Specifier	Word	Indicador de alarma

5.1.3.14 OB MC-Servo y MC-Interpolator

STEP 7 crea automáticamente los OB MC-Servo y MC-Interpolator de solo lectura cuando se crea un objeto tecnológico de movimiento y la interfaz del motor se pone a "Conexión analógica de accionamiento" o "PROFIdrive". No es necesario editar las propiedades del OB ni crear el OB directamente. La CPU utiliza estos OB para la regulación en lazo cerrado. Encontrará más detalles en el sistema de información de STEP 7.

5.1.3.15 MC-PreServo

El OB MC-PreServo puede programarse de forma que contenga la lógica del programa para que el programa STEP 7 se ejecute justo antes de que se ejecute el OB MC-Servo.

Eventos de MC-PreServo

El OB MC-PreServo permite leer en microsegundos la información del ciclo de aplicación configurado.

Tabla 5-14 Información de arranque del OB MC-PreServo

Entrada	Tipo de datos	Descripción
Initial_Call	BOOL	TRUE indica la primera llamada de este OB en una transición de STOP a RUN
PIP_Input	BOOL	TRUE indica que la memoria imagen de proceso de las entradas asociada está actualizada.
PIP_Output	BOOL	TRUE indica que la CPU transfirió a las salidas la memoria imagen de proceso de las salidas asociada en un tiempo adecuado después del último ciclo.
IO_System	USINT	Número del sistema de periferia descentralizada que dispara la alarma
Event_Count	INT	n: número de ciclos perdidos -1: número desconocido de ciclos perdidos (p. ej. porque el ciclo ha cambiado)
Synchronous	BOOL	Reservado
CycleTime	UDINT	Visualización del ciclo de aplicación configurado para el OB MC-Servo en microsegundos

5.1.3.16 MC-PostServo

El OB MC-PostServo puede programarse de forma que contenga la lógica del programa para que el programa STEP 7 se ejecute justo después de que se ejecute el OB MC-Servo.

Eventos de MC-PostServo

El OB MC-PostServo permite leer en microsegundos la información del ciclo de aplicación configurado.

Tabla 5-15 Información de arranque del OB MC-PostServo

Entrada	Tipo de datos	Descripción
Initial_Call	BOOL	TRUE indica la primera llamada de este OB en una transición de STOP a RUN
PIP_Input	BOOL	TRUE indica que la memoria imagen de proceso de las entradas asociada está actualizada.
PIP_Output	BOOL	TRUE indica que la CPU transfirió a las salidas la memoria imagen de proceso de las salidas asociada en un tiempo adecuado después del último ciclo.
IO_System	USINT	Número del sistema de periferia descentralizada que dispara la alarma
Event_Count	INT	n: número de ciclos perdidos -1: número desconocido de ciclos perdidos (p. ej. porque el ciclo ha cambiado)

Entrada	Tipo de datos	Descripción
Synchronous	BOOL	Reservado
CycleTime	UDINT	Visualización del ciclo de aplicación configurado para el OB MC-Servo en microsegundos

5.1.3.17 Prioridades y colas de espera para la ejecución de eventos

El procesamiento de la CPU es controlado por eventos. Un evento dispara la ejecución de un OB de alarma. Se puede definir el OB de alarma para un evento al crear el bloque, al configurar dispositivos o con una instrucción ATTACH o DETACH. Algunos eventos ocurren con regularidad, tales como los eventos de ciclo o cíclicos. Otros eventos ocurren una sola vez, tales como el evento de arranque y los eventos de alarma de retardo. Algunos eventos ocurren cuando el hardware dispara un evento, como p. ej. un flanco, en una entrada o un evento de contador rápido. Los eventos como errores de diagnóstico o de tiempo solo se producen cuando ocurre un error. Las prioridades de eventos y las colas de espera sirven para determinar el orden de procesamiento de los OB de alarma.

La CPU procesa los eventos en orden de prioridad, siendo 1 la prioridad más baja y 26 la más alta. Antes de la V4.0 de la CPU S7-1200, cada tipo de OB pertenecía a una clase de prioridad fija (1 a 26). A partir de la versión 4.0 se puede asignar una clase de prioridad a cada OB que se configure. El número de prioridad se configura en los atributos de las propiedades del OB.

Modos de ejecución con y sin interrupciones

Los OB (Página 75) se ejecutan en orden de prioridad de los eventos que los disparan. En las propiedades de arranque de la configuración de dispositivos de la CPU (Página 153) es posible configurar que la ejecución de los OB sea con o sin interrupciones. Recuerde que los OB de ciclo siempre son con interrupciones, pero todos los demás OB se pueden configurar para que sean con o sin interrupciones.

Con el modo con interrupciones activado, si se está ejecutando un OB y se produce un evento de mayor prioridad antes de que se complete la ejecución del OB, esta se interrumpe para permitir la ejecución del OB de evento de prioridad superior. El OB de evento de prioridad superior se ejecuta y cuando la ejecución se ha completado continúa el OB que se había interrumpido. Si se producen varios eventos mientras se está ejecutando un OB con interrupciones, la CPU procesará los eventos en orden de prioridad.

Si no activa el modo con interrupciones, un OB se ejecutará hasta el final una vez disparado, independientemente de cualquier otro evento que se dispare durante su ejecución.

Observe los dos casos siguientes, en los que unos eventos de alarma disparan un OB de ciclo y un OB de retardo de tiempo. En ambos casos, el OB de retardo (OB 201) carece de asignación de memoria imagen parcial de proceso (Página 67) y se ejecuta con prioridad 4. El OB de ciclo (OB 200) tiene una asignación de memoria imagen parcial de proceso MIPP1 y se ejecuta con prioridad 2. Las siguientes ilustraciones muestran la diferencia en la ejecución entre los modos de ejecución con y sin interrupciones:

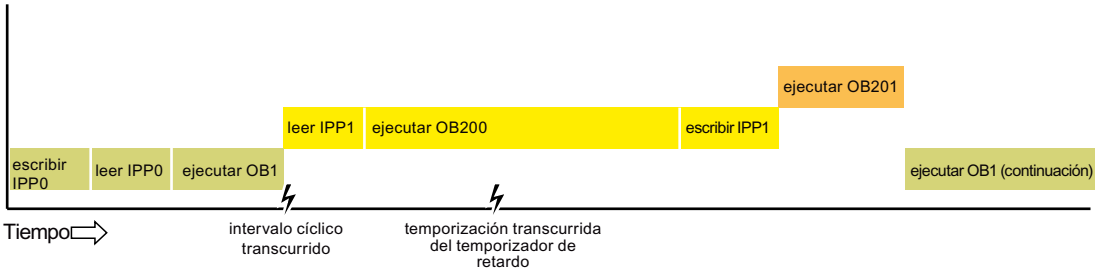


Figura 5-1 Caso 1: ejecución de OB sin interrupciones

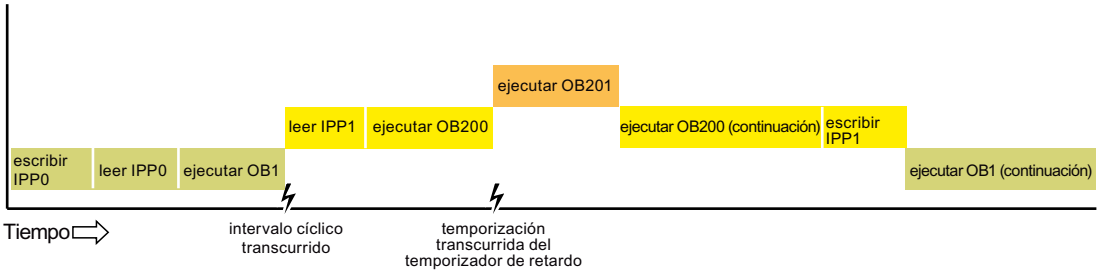


Figura 5-2 Caso 2: ejecución de OB con interrupciones

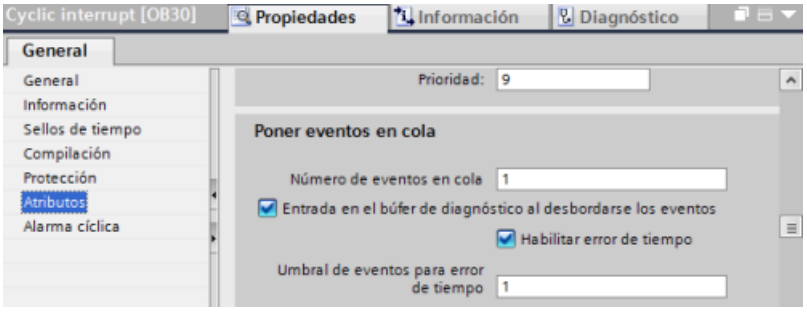
Nota

Si configura el modo de ejecución del OB sin interrupciones, un OB de error de tiempo no puede interrumpir OB que no sean OB de ciclo. Antes de la V4.0 de la CPU S7-1200, un OB de error de tiempo podía interrumpir cualquier OB en ejecución. A partir de la versión V4.0 hay que configurar la ejecución del OB con interrupciones para que un OB de error de tiempo (o cualquier otro OB de prioridad superior) pueda interrumpir los OB que se estén ejecutando y no sean OB de ciclo.

Prioridades y colas de espera para la ejecución de eventos

La CPU limita el número de eventos pendientes (en cola de espera) de una sola fuente utilizando una cola diferente para cada tipo de evento. Al alcanzar el límite de eventos pendientes de un determinado tipo, se pierde el evento siguiente. Se puede utilizar un OB de error de tiempo (Página 79) como respuesta a desbordamientos de la cola.

Tenga en cuenta que STEP 7 permite configurar algunos parámetros específicos de la cola de eventos tanto para el OB de alarma cíclica como para el OB de alarma horaria.



Encontrará más información sobre el comportamiento de la CPU en caso de sobrecarga y la cola de eventos en el sistema de información de STEP 7.

Todo evento de la CPU tiene asignada una prioridad. Generalmente, la CPU procesa los eventos según su prioridad (primero los de mayor prioridad). La CPU procesa los eventos de igual prioridad según su orden de aparición.

Tabla 5-16 Eventos de OB

Evento	**Cantidad permitida**	**Prioridad de OB predeterminada**
Ciclo del programa	1 evento de ciclo de programa Se admiten varios OB	1[1]
Arranque	1 evento de arranque [1] Se admiten varios OB	1[1]
Retardo	Hasta 4 eventos de tiempo 1 OB por evento	OB 20: 3 OB 21: 4 OB 22: 5 OB 23: 6 OB 123 a OB 32767: 3
Alarma cíclica	Hasta 4 eventos 1 OB por evento	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 a OB 32767: 7
Alarma de hardware	Hasta 50 eventos de alarma de hardware[2]	18
	1 OB por evento, pero se puede utilizar el mismo OB para varios eventos	18
Error de tiempo	1 evento (solo si está configurado)[3]	22 o 26[4]
Error de diagnóstico	1 evento (solo si está configurado)	5
Enchufe o desenchufe de módulos	1 evento	6
Fallo de rack o estación	1 evento	6
Hora	Hasta 2 eventos	2
Estado	1 evento	4
Actualizar	1 evento	4
Perfil	1 evento	4

Evento	Cantidad permitida	Prioridad de OB predeterminada
MC-Servo	1 evento	25
MC-Interpolator	1 evento	24

[1] Los eventos de arranque y de ciclo no ocurren nunca simultáneamente, ya que el evento de arranque debe haber finalizado antes de poder iniciar el evento de ciclo.

[2] Se puede contar con más de 50 OB de evento de alarma de hardware si se utilizan las instrucciones DETACH y ATTACH.

[3] Es posible configurar la CPU de modo que permanezca en RUN si se excede el tiempo de ciclo máximo o bien utilizar la instrucción RE_TRIGR para inicializar el tiempo de ciclo. Sin embargo, la CPU pasa a STOP la segunda vez que se excede el tiempo de ciclo máximo en un mismo ciclo.

[4] La prioridad de una CPU nueva V4.0 o V4.1 es 22. Si se sustituye una CPU V3.0 por una CPU V4.0 o V4.1, la prioridad será 26: es decir, la prioridad que regía para V3.0. En ambos casos, el campo de prioridad es editable, de modo que la prioridad se puede ajustar a cualquier valor entre 22 y 26.

Encontrará más detalles en el apartado "Sustitución de una CPU V3.0 por una CPU V4.x (Página 1448)".

Además, la CPU reconoce otros eventos que carecen de OB asociados. La tabla siguiente describe esos eventos y las correspondientes acciones de la CPU:

Tabla 5-17 Eventos adicionales

Evento	Descripción	Acción de la CPU
Error de acceso a E/S	Error de escritura/lectura directa de E/S	La CPU registra la primera aparición en el búfer de diagnóstico y permanece en estado RUN. Se accede a la causa del error por medio de la instrucción GET_ERROR_ID (Página 304).
Error de tiempo de ciclo máximo	La CPU excede dos veces el tiempo de ciclo configurado	La CPU registra el error en el búfer de diagnóstico y pasa al estado STOP.
Error de acceso a la periferia	Error de E/S durante una actualización de la memoria imagen de proceso	La CPU registra la primera aparición en el búfer de diagnóstico y permanece en estado RUN.
Error de programación	Error de ejecución del programa	• Si el tratamiento de errores local de bloques está activado, el sistema introduce una causa del error en la estructura del error. Se accede a la causa del error por medio de la instrucción GET_ERROR_ID (Página 304). • Si el tratamiento de errores global está activado, el sistema introduce un evento de arranque para el error de acceso en el búfer de diagnóstico y permanece en estado RUN.

Latencia de alarmas

La latencia de los eventos de alarma (es decir, el tiempo que transcurre desde que la CPU notifica que ha ocurrido un evento hasta que comienza la ejecución de la primera instrucción en el OB que procesa este evento) es de aproximadamente 175 µs, siempre que un OB de ciclo de programa sea el único subprograma activo que procese el evento de alarma en el momento de su aparición.

5.1.4 Vigilancia y configuración del tiempo de ciclo

El tiempo de ciclo es el tiempo que requiere el sistema operativo de la CPU para ejecutar la fase cíclica del estado operativo RUN. La CPU ofrece dos métodos para vigilar el tiempo de ciclo:

- Tiempo de ciclo máximo
- Tiempo de ciclo mínimo

La vigilancia del tiempo de ciclo comienza una vez finalizado el evento de arranque. Esta función se configura en la "Configuración de dispositivos" de la CPU en "Tiempo de ciclo".

La CPU vigila el ciclo de ejecución y reacciona si el tiempo de ciclo excede el tiempo máximo de ciclo configurado. La CPU genera un error y reacciona del siguiente modo si el tiempo de ciclo excede el tiempo máximo de ciclo configurado.

- Si el programa de usuario incluye un OB de alarma de error de tiempo (Página 79), la CPU lo ejecutará.
- Si el programa de usuario no incluye un OB de alarma de error de tiempo, el evento de error de tiempo generará una entrada en el búfer de diagnóstico. La CPU pasa al estado operativo STOP.

La instrucción RE_TRIGR (Página 303) (volverá a lanzar la vigilancia del tiempo de ciclo) permite inicializar el temporizador que mide el tiempo de ciclo. Si el tiempo transcurrido para ejecutar el ciclo actual es inferior al tiempo del ciclo máximo configurado multiplicado por diez, la instrucción RE_TRIGR volverá a lanzar la vigilancia de tiempo de ciclo y devolverá ENO = TRUE. De lo contrario, la instrucción RE_TRIGR no volverá a lanzar la vigilancia de tiempo del ciclo. Devolverá ENO = FALSE.

Generalmente, el ciclo se ejecuta tan rápido como sea posible y el ciclo siguiente comienza cuando finaliza el ciclo actual. En función del programa de usuario y las tareas de comunicación, el tiempo de ciclo puede fluctuar de ciclo en ciclo. Para eliminar esta variación, la CPU admite un tiempo de ciclo mínimo opcional. Si activa esta función opcional y proporciona un tiempo de ciclo mínimo en ms, la CPU se retardará después de ejecutar los OB de ciclo de programa hasta que transcurra el tiempo de ciclo mínimo antes de repetir el ciclo.

Si la CPU finaliza el ciclo normal antes del tiempo de ciclo mínimo especificado, aprovechará el tiempo restante para realizar tareas de diagnóstico en runtime y/o procesar órdenes de comunicación.

Si la CPU no finaliza el ciclo normal dentro del tiempo de ciclo mínimo especificado, lo finalizará normalmente (incluyendo el procesamiento de las órdenes de comunicación), sin que el rebase

del tiempo de ciclo mínimo cause una reacción del sistema. La tabla siguiente define los rangos y valores predeterminados de las funciones de observación para el tiempo de ciclo:

Tabla 5-18 Rango para el tiempo de ciclo

Tiempo de ciclo	Rango (ms)	Valor predeterminado
Tiempo de ciclo máximo[1]	1 a 6000	150 ms
Tiempo de ciclo mínimo[2]	1 hasta tiempo de ciclo máximo	Deshabilitado

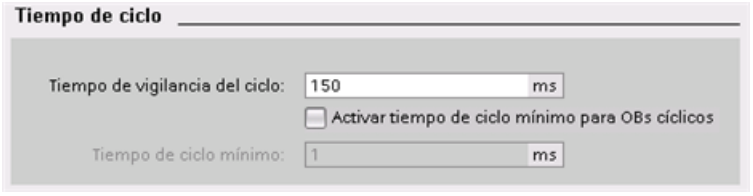
[1] El tiempo de ciclo máximo siempre está habilitado. Configure un tiempo de ciclo entre 1 ms y 6000 ms. El valor predeterminado es de 150 ms.

[2] El tiempo de ciclo mínimo es opcional y está deshabilitado de forma predeterminada. En caso necesario, configure un tiempo de ciclo comprendido entre 1 ms y el tiempo de ciclo máximo.

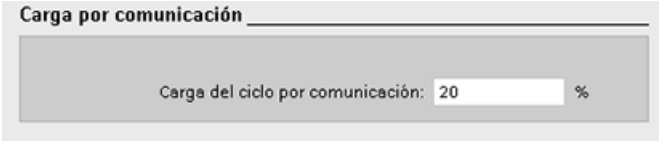
Configurar el tiempo de ciclo y la carga de comunicación

Las propiedades de la CPU en la "Configuración de dispositivos" permiten configurar los siguientes parámetros:

- Ciclo: Puede introducir un tiempo de vigilancia de ciclo máximo. También puede habilitar e introducir un tiempo de ciclo mínimo.



- Carga de comunicación: Es posible configurar un porcentaje del tiempo que debe dedicarse a las tareas de comunicación.



Nota

Prioridad de la comunicación

Las tareas de comunicación tienen la prioridad 1. Puesto que 1 es la prioridad más baja, es posible que otros eventos de la CPU interrumpan el procesamiento de la comunicación. Las interrupciones provocadas por otros eventos pueden influir negativamente en el procesamiento de la comunicación durante el tiempo de ciclo. Existe la posibilidad de ajustar el porcentaje de "Carga del ciclo por comunicación" para incrementar la proporción del tiempo de ciclo dedicada al procesamiento de la comunicación.

Encontrará más información sobre el ciclo en el apartado "Vigilancia del tiempo de ciclo". (Página 91)

5.1.5 Memoria de la CPU

Gestión de la memoria

La CPU provee las áreas de memoria siguientes para almacenar el programa de usuario, los datos y la configuración:

- La memoria de carga permite almacenar de forma no volátil el programa de usuario, los datos y la configuración. Cuando se carga un proyecto en la CPU, esta almacena primero el programa en el área de memoria de carga. Esta área se encuentra bien sea en una SIMATIC Memory Card (si está disponible) o en la CPU. La CPU conserva esta área de memoria no volátil incluso tras un corte de alimentación. La SIMATID Memory Card ofrece mayor espacio de almacenamiento que el integrado en la CPU.
- La memoria de trabajo ofrece almacenamiento volátil para algunos elementos del proyecto mientras se ejecuta el programa de usuario. La CPU copia algunos elementos del proyecto desde la memoria de carga en la memoria de trabajo. Esta área volátil se pierde si se desconecta la alimentación. La CPU la restablece al retornar la alimentación.
- La memoria remanente permite almacenar de forma no volátil un número limitado de valores de la memoria de trabajo. La CPU utiliza el área de memoria remanente para almacenar los valores de algunas posiciones de memoria durante un corte de alimentación. Cuando se produce una caída o un corte de la alimentación, la CPU restaura esos valores remanentes al restablecer la alimentación.

Para ver el uso de memoria de un bloque de programa compilado, haga clic con el botón derecho del ratón en la carpeta "Bloques de programa" del árbol del proyecto STEP 7 y seleccione "Propiedades" en el menú contextual. En las propiedades de compilación se muestra la memoria de carga y la memoria de trabajo del bloque compilado.

Para ver el uso de memoria de la CPU online, haga doble clic en "Online y diagnóstico" en STEP 7, expanda "Diagnóstico" y seleccione "Memoria".

Memoria remanente

Es posible evitar la pérdida de datos tras un corte de alimentación marcando determinados datos como remanentes. La CPU permite configurar como remanentes los datos siguientes:

- Área de marcas (M): El tamaño de la memoria remanente para marcas puede definirse en la tabla de variables PLC o en el plano de ocupación. El área de marcas remanente comienza siempre en MB0, abarcando consecutivamente un determinado número de bytes. Para definir este valor, haga clic en el botón "Remanencia" de la barra de herramientas de la tabla de variables PLC o del plano de ocupación. Introduzca el número de bytes M que deben ser remanentes a partir de MB0.
  Nota: existe la posibilidad de visualizar el plano de ocupación para cada bloque seleccionando un bloque en la carpeta de bloques de programa y, seguidamente, seleccionando el comando de menú **Herramientas > Plano de ocupación**.
- Variables de un bloque de función (FB): Si un FB es del tipo "Accedo optimizado al bloque", el editor de interfaces para dicho FB incluye una columna "Remanencia". En esta columna puede seleccionarse "Remanente", "No remanente" o "Activar en IDB" para cada variable individual. Cuando se incluye un FB de este tipo en el programa, el DB de instancia que equivale al FB incluye también esta columna "Remanencia". El estado remanente de una variable solo se puede modificar desde el editor de la interfaz del DB de instancia si se ha seleccionado "Ajustar en IDB" (activado en el bloque de datos de instancia) en la selección "Remanencia" de la variable en el FB optimizado.
  Si un FB **no** es del tipo "Acceso optimizado al bloque", el editor de interfaces para dicho FB no incluye una columna "Remanencia". Sin embargo, cuando se incluye un FB de este tipo en el programa, el DB de instancia que equivale al FB incluye una columna "Remanencia" que puede editarse. En este caso, si se selecciona la opción "Remanente" para cualquier variable se seleccionan **todas** las variables. Análogamente, si se deselecciona la opción para cualquier variable se deseleccionan **todas** las variables.
  Para ver si un FB está optimizado o modificarlo, abra las propiedades del FB y seleccione los atributos.
- Variables de un bloque de datos global: Si se selecciona "Acceso optimizado al bloque" para los atributos en las propiedades del bloque de datos, podrá definirse para cada variable si será remanente o no. Si no se selecciona "Acceso optimizado al bloque", todas las variables del bloque de datos tendrán el mismo estado. Las variables serán todas remanentes o todas no remanentes.

La CPU soporta un total de 14336 bytes de datos remanentes para un proyecto V4.5 cargado en un PLC con firmware V4.5 Si está cargado un proyecto V4.4, los datos remanentes ascienden a 10240 bytes. Para ver cuánto espacio está disponible, haga clic en el botón "Remanencia" de la barra de herramientas de la tabla de variables PLC o del plano de ocupación. Aunque aquí se especifica el rango remanente para la memoria M, la segunda fila indica la memoria restante disponible en total para M y DB conjuntamente. Hay que tener en cuenta que para que este valor sea preciso, se deben compilar todos los bloques de datos con variables remanentes.

Nota

Cuando se carga un programa no se borran ni se cambian los valores existentes en la memoria remanente. Si desea borrar la memoria remanente antes de realizar una carga, restablezca la configuración de fábrica de la CPU antes de cargar el programa.

5.1.5.1 Marcas de sistema y de ciclo

Los bytes de "marcas de sistema" y "marcas de ciclo" se habilitan en las propiedades de la CPU. La lógica del programa puede referenciar los distintos bits de estas funciones por sus nombres de variable.

- Un byte del área de marcas (M) se puede asignar a las marcas de sistema. El byte de marcas de sistema pone a disposición los siguientes cuatro bits que puede referenciar el programa de usuario mediante los siguientes nombres de variables:
  - Primer ciclo: El bit (nombre de variable "FirstScan") se pone a 1 durante el primer ciclo tras finalizar el OB de arranque. (Una vez finalizada la ejecución del primer ciclo, el bit "Primer ciclo" se pone a 0.)
  - Ha cambiado el estado de diagnóstico: (nombre de la variable: "DiagStatusUpdate") se pone a 1 durante un ciclo, cuando la CPU registra un evento de diagnóstico. Puesto que la CPU no activa el bit "DiagStatusUpdate" hasta el final de la primera ejecución de los OBs de ciclo de programa, el programa de usuario no puede detectar si ha cambiado el diagnóstico durante la ejecución de los OBs de arranque o bien durante la primera ejecución de los OBs de ciclo de programa.
  - Siempre 1 (high): El bit (nombre de variable "AlwaysTRUE") está siempre puesto a 1.
  - Siempre 0 (low): El bit (nombre de variable "AlwaysFALSE") está siempre puesto a 0.
- Es posible asignar un byte de marcas de ciclo en el área de marcas. Todo bit del byte de marcas de ciclo genera un impulso de onda cuadrada. El byte de marcas de ciclo ofrece 8 frecuencias diferentes, comprendidas entre 0,5 Hz (lenta) hasta 10 Hz (rápida). Estos bits pueden utilizarse como bits de control para disparar acciones cíclicas en el programa de usuario, especialmente si se combinan con instrucciones de detección de flancos.

La CPU inicializa estos bytes cuando el estado operativo cambia de STOP a ARRANQUE. Los bits de las marcas de ciclo cambian de forma síncrona al reloj de la CPU durante los estados operativos ARRANQUE y RUN.

⚠ PRECAUCIÓN

Riesgos asociados con la sobrescritura de los bits de marca de sistema o de ciclo

Si se sobrescriben los bits de marcas de sistema o de ciclo, se podrían corromper los datos en estas funciones. Debido a ello, el programa de usuario funcionará incorrectamente, lo que podría ocasionar daños materiales y lesiones corporales.

Puesto que las marcas de ciclo y de sistema forman no están reservadas en la memoria M, las instrucciones o comunicaciones pueden escribir en estas posiciones de memoria y corromper los datos.

Evite escribir datos en estas direcciones para garantizar el funcionamiento correcto de estas funciones y prevea siempre un circuito de parada de emergencia para el proceso o la máquina.

La marca de sistema configura un byte con bits que se activan (se ponen a 1) con un evento determinado.

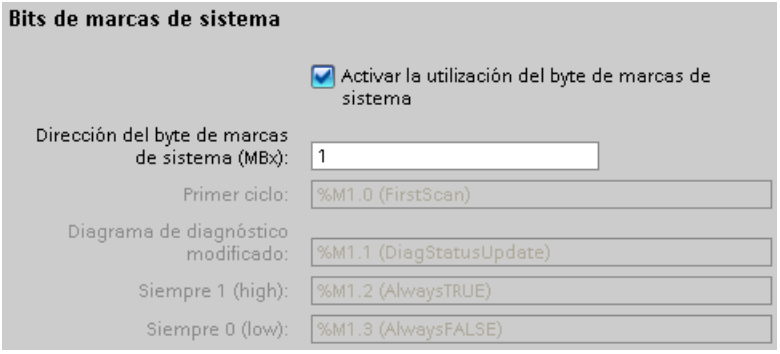


Tabla 5-19 Marcas de sistema

7	6	5	4	3	2	1	0
Reservado Valor 0				Siempre off Valor 0	Siempre ON Valor 1	Indicador de estado de diagnóstico • 1: Cambiar • 0: No cambiar	Indicador de primer ciclo • 1: Primer ciclo tras arranque • 0: No es primer ciclo

Las marcas de ciclo configuran un byte que activa y desactive los distintos bits en intervalos fijos. Cada bit de reloj genera un impulso de onda cuadrada en el bit correspondiente del área de marcas. Estos bits pueden utilizarse como bits de control para disparar acciones cíclicas en el programa de usuario, especialmente si se combinan con instrucciones de detección de flancos.



Tabla 5-20 Marcas de ciclo

Número de bit	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Nombre de la variable								
Período (s)	2,0	1,6	1,0	0,8	0,5	0,4	0,2	0,1
Frecuencia (Hz)	0,5	0,625	1	1,25	2	2,5	5	10

Dado que la marca de ciclo es asíncrona respecto al ciclo de la CPU, el estado de la marca de ciclo puede cambiar varias veces durante un ciclo largo.

5.1.6 Búfer de diagnóstico

La CPU soporta un búfer de diagnóstico que contiene una entrada para cada evento de diagnóstico. Toda entrada incluye la fecha y hora del evento, así como su categoría y descripción. Las entradas se visualizan en orden cronológico. El evento más reciente aparece en primer lugar. En este búfer están disponibles los 50 eventos más recientes. Cuando se llena el búfer, un evento nuevo reemplaza al evento más antiguo. Cuando se corta la alimentación, se almacenan los eventos.

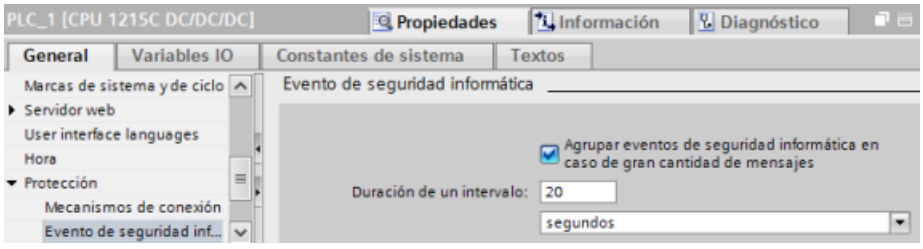
Los siguientes tipos de eventos se registran en el búfer de diagnóstico:

- Todo evento de diagnóstico del sistema, p. ej. errores de la CPU y de los módulos
- Todo cambio de estado de la CPU (todo arranque, toda transición a STOP, toda transición a RUN)

Para acceder al búfer de diagnóstico (Página 1211) es preciso estar online. En la vista "Online y diagnóstico" el búfer de diagnóstico está en "Diagnóstico > Búfer de diagnóstico".

Reducir el número de eventos de diagnóstico de seguridad

Algunos eventos de seguridad generan entradas repetidas en el búfer de diagnóstico. Estos avisos pueden llenar el búfer de diagnóstico y también ocultar otros avisos de eventos. Es posible configurar el PLC para que limite el número de avisos de diagnóstico procedentes de eventos de seguridad. Las selecciones que deben realizarse en la configuración de dispositivos de la CPU dependen del intervalo de tiempo en el que deban suprimirse los avisos repetitivos:



Si se elige agrupar los eventos de seguridad en un intervalo de tiempo, este puede ser en segundos, minutos u horas y su valor numérico puede estar comprendido entre 1 y 255.

Si se elige restringir los eventos de seguridad, se restringirán los tipos de eventos siguientes:

- Pasar a online con la contraseña correcta o incorrecta
- Se han detectado datos de comunicación manipulados
- Se han detectado datos manipulados en la Memory Card
- Se ha detectado un archivo de actualización de firmware manipulado
- Se ha descargado a la CPU un nivel de protección distinto (protección de acceso)
- La legitimación de contraseña está restringida o activada (por instrucción o display de la CPU)
- Acceso online denegado debido a que se ha excedido el número admisible de intentos de acceso simultáneos
- Tiempo excedido cuando una conexión online existente está inactiva
- Inicio de sesión en el servidor web con la contraseña correcta o incorrecta

- Crear una copia de seguridad de la CPU
- Restaurar la configuración de la CPU

5.1.7 Reloj en tiempo real

La CPU soporta un reloj en tiempo real. Un condensador de alto rendimiento suministra la energía necesaria para que el reloj pueda seguir funcionando mientras está desconectada la alimentación de la CPU. El condensador de alto rendimiento se carga mientras está conectada la alimentación de la CPU. Tras haber estado conectada la alimentación de la CPU como mínimo 24 horas, la carga del condensador de alto rendimiento será suficiente para que el reloj pueda funcionar unos 20 días.

STEP 7 ajusta el reloj en tiempo real a la hora del sistema, que está a un valor predeterminado tras el primer encendido o tras un restablecimiento de los ajustes de fábrica. Para usar el reloj en tiempo real primero hay que ajustarlo. Los sellos de tiempo como los usados en las entradas del búfer de diagnóstico, los archivos de registro o las entradas de registros se basan en la hora del sistema. La hora se ajusta mediante la función "Ajustar la hora" (Página 1204) de la vista "Online y diagnóstico" de la CPU online. STEP 7 calcula entonces la hora del sistema a partir de la hora ajustada y de la diferencia del sistema operativo Windows con el tiempo universal coordinado (UTC). El ajuste de la hora a la hora local actual genera una hora del sistema UTC si los ajustes de horario de verano y de zona horaria del sistema operativo Windows corresponden a la ubicación regional.

STEP 7 dispone de instrucciones (Página 328) para leer y escribir la hora del sistema (RD_SYS_T y WR_SYS_T), para leer la hora local (RD_LOC_T) y para ajustar la zona horaria (SET_TIMEZONE). La instrucción RD_LOC_T calcula la hora local usando las diferencias por zona horaria y por horario de verano ajustadas en la configuración "Hora" de las propiedades generales de la CPU (Página 153). Esos ajustes permiten establecer la zona horaria para la hora local, habilitar opcionalmente el horario de verano y especificar las fechas y horas iniciales y finales del horario de verano. También se puede usar la instrucción SET_TIMEZONE para configurar esos ajustes.

5.1.8 Configurar las salidas en una transición de RUN a STOP

Es posible configurar la reacción de las salidas digitales y analógicas cuando la CPU se encuentre en estado operativo STOP. Es posible congelar los valores de las salidas o aplicar un valor sustitutivo a cualquier salida de una CPU, SB o SM:

- Sustituir un valor de salida específico (ajuste predeterminado): Para cada salida (canal) de la CPU, de la SB o del SM se define un valor sustitutivo.
  El valor sustitutivo predeterminado de los canales de salida digitales es OFF y el de los canales de salida analógicos es 0.
- Congelar las salidas a su último estado: Las salidas conservan su valor actual en el momento de la transición de RUN a STOP. Después del arranque, las salidas se ajustan al valor sustitutivo predeterminado.

La reacción de las salidas se configura en la "Configuración de dispositivos". Seleccione los dispositivos individuales y utilice la ficha "Propiedades" para configurar las salidas de cada dispositivo.

Nota

Algunos módulos de periferia descentralizada ofrecen ajustes adicionales para la reacción a STOP de la CPU. Seleccione dichos módulos en la lista de opciones de la configuración de dispositivos.

Cuando la CPU cambia de RUN a STOP, conserva la memoria imagen de proceso y escribe los valores correspondientes en las salidas digitales y analógicas según la configuración.

5.2 Almacenamiento de datos, áreas de memoria, E/S y direccionamiento

5.2.1 Acceder a los datos del S7-1200

STEP 7 facilita la programación simbólica. Se crean nombres simbólicos o "variables" para las direcciones de los datos, ya sea como variables PLC asignadas a direcciones de memoria y E/S o como variables locales utilizadas dentro de un bloque lógico. Para utilizar estas variables en el programa de usuario basta con introducir el nombre de variable para el parámetro de instrucción.

Para una mejor comprensión de cómo la CPU estructura y direcciona las áreas de memoria, los siguientes párrafos explican el direccionamiento "absoluto" al que se refieren las variables PLC. La CPU ofrece varias opciones para almacenar datos durante la ejecución del programa de usuario:

- Memoria global: La CPU ofrece distintas áreas de memoria, incluyendo entradas (I), salidas (Q) y marcas (M). Todos los bloques lógicos pueden acceder sin restricción alguna a esta memoria.
- Tabla de variables PLC: se pueden especificar nombres simbólicos en la tabla de variables PLC de STEP 7 para posiciones de memoria específicas. Esas variables son globales dentro del programa STEP 7 y permiten la programación con nombres significativos para la aplicación.
- Bloque de datos (DB): Es posible incluir DBs en el programa de usuario para almacenar los datos de los bloques lógicos. Los datos almacenados se conservan cuando finaliza la ejecución del bloque lógico asociado. Un DB "global" almacena datos que pueden ser utilizados por todos los bloques lógicos, mientras que un DB de instancia almacena datos para un bloque de función (FB) especifico y está estructurado según los parámetros del FB.
- Memoria temporal: Cada vez que se llama un bloque lógico, el sistema operativo de la CPU asigna la memoria temporal o local (L) que debe utilizarse durante la ejecución del bloque. Cuando finaliza la ejecución del bloque lógico, la CPU reasigna la memoria local para la ejecución de otros bloques lógicos.

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria. Las referencias a las áreas de memoria de entrada (I) o salida (Q), como I0.3 o Q1.7, acceden a la memoria imagen

del proceso. Para acceder inmediatamente a la entrada o salida física es preciso añadir ":P" a la dirección (p. ej. I0.3:P, Q1.7:P o "Stop:P").

Tabla 5-21 Áreas de memoria

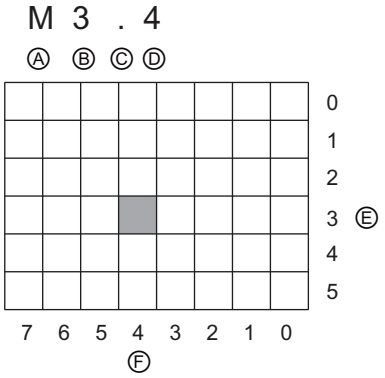
Área de memoria	**Descripción**	**Forzado permanente**	**Remanente**
I Memoria imagen de proceso de las entradas I_:P [1] (entrada física)	Se copia de las entradas físicas al inicio del ciclo	No	No
	Lectura inmediata de las entradas físicas de la CPU, SB y SM	Sí	No
Q Memoria imagen de proceso de las salidas Q_:P [1] (salida física)	Se copia en las salidas físicas al inicio del ciclo	No	No
	Escritura inmediata en las salidas físicas de la CPU, SB y SM	Sí	No
M Área de marcas	Control y memoria de datos	No	Sí (opcional)
L Memoria temporal	Datos locales temporales de un bloque	No	No
DB Bloque de datos	Memoria de datos y de parámetros de FBs	No	Sí (opcional)

[1] Para acceder inmediatamente (leer o escribir) a las entradas o salidas físicas es preciso añadir ":P" a la dirección o variable (p. ej. I0.3:P, Q1.7:P o "Stop:P").

Toda posición de memoria diferente tiene una dirección unívoca. El programa de usuario utiliza estas direcciones para acceder a la información de la posición de memoria. La dirección absoluta consta de los elementos siguientes:

- Identificador de área de memoria (como I, Q o M)
- Tamaño de los datos a los que se va a acceder ("B" para Byte, "W" para Word, o "D" para DWord)
- Dirección inicial de los datos (como byte 3 o palabra 3)

Al acceder a un bit en la dirección para un valor booleano, no se introduce ningún nemónico para el tamaño. Solo se introduce el área de memoria, la ubicación del byte y la ubicación del bit de los datos (como I0.0, Q0.1, o M3.4).



A	Identificador de área	E	Bytes del área de memoria
B	Dirección de byte: Byte 3	F	Bits del byte seleccionado
C	Separador ("byte.bit")		
D	Bit del byte (bit 4 de 8)		

En el ejemplo, el área de memoria y la dirección del byte (M = área de memoria de bit; y 3 = Byte 3) van seguidas de un punto ("."), que separa la dirección del bit (bit 4).

Acceder a los datos en las áreas de memoria de la CPU

STEP 7 facilita la programación simbólica. Por lo general, las variables se crean en la tabla de variables PLC, en un bloque de datos o en la interfaz de un OB, FC o FB. Estas variables incluyen un nombre, tipo de datos, offset y comentario. Además, en un bloque de datos es posible especificar un valor de arranque. Estas variables pueden utilizarse durante la programación, introduciendo el nombre de la variable en el parámetro de la instrucción. Opcionalmente se puede introducir el operando absoluto (área de memoria, tamaño y offset) en el parámetro de la instrucción. Los ejemplos de los apartados siguientes muestran cómo introducir operandos absolutos. El editor de programación antepone automáticamente el carácter % al operando absoluto. Es posible cambiar entre las siguientes vistas del editor de programación: simbólica, simbólica y absoluta o absoluta.

I (memoria imagen de proceso de las entradas): La CPU consulta las entradas de periferia (físicas) inmediatamente antes de ejecutar el OB de ciclo en cada ciclo y escribe estos valores en la memoria imagen de proceso de las entradas. A la memoria imagen de proceso de las entradas se puede acceder en formato de bit, byte, palabra o palabra doble. Aunque se permiten accesos de lectura y escritura, generalmente solo se leen las entradas de la memoria imagen de proceso.

Tabla 5-22 Direccionamiento absoluto para memoria I

Bit	I[dirección de byte].[dirección de bit]	I0.1
Byte, palabra o palabra doble	I[tamaño][dirección de byte inicial]	IB4, IW5 o ID12

Agregando una ":P" a la dirección es posible leer inmediatamente las entradas digitales y analógicas de CPU, SB, SM o módulo descentralizado. La diferencia entre un acceso que utiliza I_:P en vez de I es que los datos provienen directamente de las entradas direccionadas, en vez de la memoria imagen de proceso de las entradas. El acceso I_:P también se denomina "lectura inmediata", puesto que los datos se leen inmediatamente del origen y no de una copia creada la última vez que se actualizó la memoria imagen de proceso de las entradas.

Puesto que las entradas físicas reciben sus valores directamente de los aparatos de campo conectados a ellas, está prohibido escribir en estas entradas. Por tanto, los accesos I_:P son de solo lectura, a diferencia de los accesos I que pueden ser de lectura o escritura.

Los accesos I_:P también están restringidos por el tamaño de las entradas que soporta una única CPU, SB o SM, redondeado al byte más próximo. Así, por ejemplo, si las entradas de un SB de 2 DI / 2 DQ se configuran para que se inicien en I4.0, es posible acceder a las entradas como I4.0:P e I4.1:P o como IB4:P. Los accesos a I4.2:P a través de I4.7:P no se rechazan pero no tienen sentido, porque estas entradas no se utilizan. Los accesos a IW4:P y ID4:P están prohibidos, puesto que exceden el offset de bytes asociado a la SB.

Los accesos mediante I_:P no afectan el valor correspondiente almacenado en la memoria imagen de proceso de las entradas.

Tabla 5-23 Direccionamiento absoluto para memoria I (inmediata)

Bit	I[dirección de byte].[dirección de bit]:P	I0.1:P
Byte, palabra o palabra doble	I[tamaño][dirección de byte inicial]:P	IB4:P, IW5:P o ID12:P

Q (memoria imagen de proceso de las salidas): La CPU copia los valores almacenados en la imagen de proceso de las salidas en las salidas físicas. A la memoria imagen de proceso de las salidas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura a la memoria imagen de proceso de las salidas.

Tabla 5-24 Direccionamiento absoluto para memoria Q

Bit	Q[dirección de byte].[dirección de bit]	Q1.1
Byte, palabra o palabra doble	Q[tamaño][dirección de byte inicial]	QB5, QW10, QD40

Agregando una ":P" a la dirección es posible escribir inmediatamente en las salidas digitales y analógicas físicas de CPU, SB, SM o módulo descentralizado. La diferencia entre un acceso que utiliza Q_:P en vez de Q es que los datos se escriben directamente en las salidas direccionadas y también en la memoria imagen de proceso de las salidas. El acceso Q_:P se denomina a veces "escritura inmediata", puesto que los datos se escriben inmediatamente en la salida de destino. Por tanto, esta no tiene que esperar hasta la siguiente actualización desde la memoria imagen de proceso de las salidas.

Puesto que las salidas físicas controlan directamente los aparatos de campo conectados a ellas, está prohibido leer estas salidas. Por tanto, los accesos Q_:P son de solo escritura, a diferencia de los accesos Q que pueden ser de lectura o escritura.

Los accesos Q_:P también están restringidos por el tamaño de las salidas que soporta una única CPU, SB o SM, redondeado al byte más próximo. Así, por ejemplo, si las salidas de una SB de 2 DI / 2 DQ se configuran para que arranquen en Q4.0, es posible acceder a las salidas como Q4.0:P y Q4.1:P o como QB4:P. Los accesos a Q4.2:P a través de Q4.7:P no se rechazan pero no tienen sentido, porque estas salidas no se utilizan. Los accesos a QW4:P y QD4:P están prohibidos, puesto que exceden el offset de bytes asociado a la SB.

Los accesos mediante Q_:P afectan tanto la salida física como el valor correspondiente almacenado en la memoria imagen de proceso de las salidas.

Tabla 5-25 Direccionamiento absoluto para memoria Q (inmediata)

Bit	Q[dirección de byte].[dirección de bit]:P	Q1.1:P
Byte, palabra o palabra doble	Q[tamaño][dirección de byte inicial]:P	QB5:P, QW10:P o QD40:P

M (área de marcas): El área de marcas (memoria M) puede utilizarse para relés de control y datos para almacenar el estado intermedio de una operación u otra información de control. Al área de marcas se puede acceder en formato de bit, byte, palabra o palabra doble. Se permiten accesos de lectura y escritura al área de marcas.

Tabla 5-26 Direccionamiento absoluto para memoria M

Bit	M[dirección de byte].[dirección de bit]	M26.7
Byte, palabra o palabra doble	M[tamaño][dirección de byte inicial]	MB20, MW30, MD50

Temp (memoria temporal): La CPU asigna la memoria temporal según sea necesario. La CPU asigna la memoria temporal del bloque lógico e inicializa las posiciones de memoria a 0 en el momento en el que se inicia el bloque lógico (para un OB) o llama al bloque lógico (para una FC o un FB).

La memoria temporal es similar al área de marcas, con una excepción importante: el área de marcas tiene un alcance "global", en tanto que la memoria temporal tiene un alcance "local".

- Área de marcas: Cualquier OB, FC o FB puede acceder a los datos del área de marcas. Esto significa que los datos están disponibles globalmente para todos los elementos del programa de usuario.
- Memoria temporal: La CPU restringe el acceso los datos de la memoria temporal al OB, la FC o el FB que ha creado o declarado la posición de memoria temporal. Las posiciones de memoria temporal son siempre locales y los diferentes bloques lógicos no comparten la memoria temporal, incluso si un bloque lógico llama otro bloque lógico. Ejemplo: Cuando un OB llama una FC, esta no puede acceder a la memoria temporal del OB que ha efectuado la llamada.

La CPU pone a disposición memoria temporal (local) para cada nivel de prioridad de OB:

- 16 KB para arranque y ciclo, incluyendo los FBs y FCs asociados
- 6 KB para cada subproceso de evento de alarma, incluidos los FB y las FC

A la memoria temporal se puede acceder solo con direccionamiento simbólico.

La estructura de llamada en STEP 7 indica la cantidad de memoria temporal (local) que utilizan los bloques del programa. Seleccione Información del programa en el árbol del proyecto y, a continuación, la pestaña Estructura de llamada. Se mostrarán todos los OB del programa, que pueden desplegarse para ver los bloques que llaman. De cada bloque se ve la asignación de datos locales. También es posible acceder a la pantalla Estructura de llamada con el comando de menú **Herramientas > Estructura de llamada** de STEP 7.

DB (bloque de datos): Los bloques de datos se utilizan para almacenar diferentes tipos de datos, incluyendo el estado intermedio de una operación u otros parámetros de control de FBs, así

como estructuras de datos requeridas para numerosas instrucciones, p. ej. temporizadores y contadores. A los bloques de datos se puede acceder en formato de bit, byte, palabra o palabra doble. A los bloques de datos que se pueden leer y escribir se permiten accesos de lectura y escritura. A los bloques de datos de solo lectura se permiten solo los accesos de lectura.

Tabla 5-27 Direccionamiento absoluto para memoria DB

Bit	DB[número de bloque de datos].DBX[dirección de byte].[dirección de bit]	DB1.DBX2.3
Byte, palabra o palabra doble	DB[número de bloque de datos].DB [tamaño][dirección de byte inicial]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Nota

Cuando especifica una dirección absoluta en LAD o FDB, STEP 7 coloca el carácter "%" antes de esta dirección para indicar que se trata de una dirección absoluta. Durante la programación, puede especificar una dirección absoluta con o sin el carácter "%" (por ejemplo: %I0.0 o bien I.0). Si no se especifica, STEP 7 incluye el carácter "%".

En SCL, debe introducir "%" antes de la dirección para indicar que se trata de una dirección absoluta. Sin el símbolo "%", STEP 7 genera un error de variable no definido durante el tiempo de compilación.

Configuración de las E/S de la CPU y los módulos de E/S





Al agregar una CPU y módulos E/S a la configuración del dispositivo, STEP 7 asigna automáticamente direcciones de entrada y salida. El direccionamiento predeterminado puede cambiarse seleccionando el campo de dirección en la configuración de dispositivos e introduciendo números nuevos.

- STEP 7 asigna las entradas y salidas digitales en grupos de 8 E/S (1 byte), sin importar si el módulo utiliza todas las E/S o no.
- STEP 7 distribuye entradas y salidas analógicas en grupos de 2 de forma que cada punto analógico ocupa 2 bytes (16 bits).

La figura muestra un ejemplo de una CPU 1214C con dos SM y una SB. En este ejemplo se puede cambiar la dirección del módulo DI8 a 2 en lugar de 8. La herramienta ayuda a cambiar los rangos de direcciones que tienen un tamaño incorrecto o que crean un conflicto con otras direcciones.

5.3 Procesamiento de valores analógicos

Los módulos de señales analógicas proporcionan señales de entrada o esperan valores de salida que representen un rango de tensión o de corriente. Estos rangos son ±10 V, ±5 V, ±2,5 V o 0 - 20 mA. Los valores que devuelven los módulos son valores enteros en los que 0 a 27648 representa el rango nominal de corriente, y -27648 a 27648 de tensión. Cualquier valor fuera del rango representa un rebase por exceso o por defecto. Consulte las tablas de representación de entradas analógicas (Página 1352) y representación de salidas analógicas (Página 1353) para más detalles sobre los tipos de valores fuera de rango.

En el programa de control puede ser necesario utilizar estos valores en unidades de ingeniería, por ejemplo, para representar un volumen, temperatura, peso o cualquier otro valor cuantitativo. En el caso de una entrada analógica, para hacerlo primero hay que normalizar el valor analógico a un valor real (coma flotante) de 0,0 a 1,0. A continuación hay que escalarlo a los valores mínimo y máximo de las unidades de ingeniería que representa. En el caso de valores de unidades de ingeniería que deben convertirse a valores de salida analógicos, primero hay que normalizar el valor en las unidades de ingeniería a un valor entre 0,0 y 1,0, y a continuación escalarlo entre 0 y 27648 o -27648 a 27648, dependiendo del rango del módulo analógico. Para este propósito, STEP 7 proporciona las instrucciones NORM_X y SCALE_X (Página 290). También se puede utilizar la instrucción CALCULATE (Página 236) para escalar los valores analógicos.

Ejemplo: procesamiento de valores analógicos

Considere, por ejemplo, una entrada analógica que tiene un rango de intensidad de 0 - 20 mA. El módulo de entradas analógicas devuelve valores en el rango de 0 a 27648 para los valores medidos. Para este ejemplo, imagine que está empleando este valor de entrada analógica para medir un rango de temperatura entre 50 °C y 100 °C. Unos cuantos valores de muestreo tendrían el significado siguiente:

Valor de entrada analógica	Unidades físicas
0	50 °C
6192	62,5 °C
12384	75 °C
18576	87,5 °C
27648	100 °C

El cálculo para determinar unidades físicas a partir del valor de entrada analógica en este ejemplo es el siguiente:

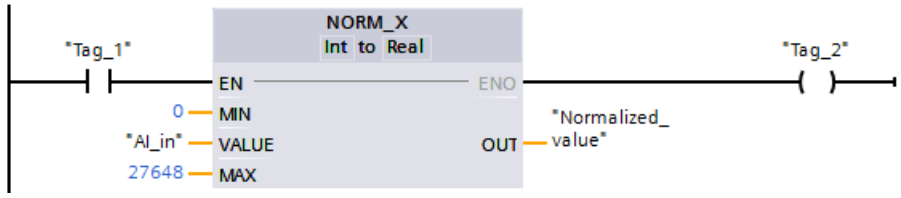
Valor de unidades físicas = 50 + (valor de entrada analógica) * (100 - 50) / (27648 - 0)

Por regla general, la ecuación sería:

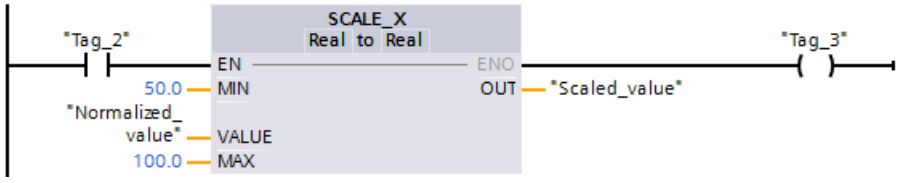
valor de unidades físicas = (rango inferior de unidades físicas) +
(valor de entrada analógica) *
(rango superior de unidades físicas - rango inferior de unidades físicas) /
(rango máximo de entrada analógica - rango mínimo de entrada analógica)

En las aplicaciones de PLC, el método típico es la normalización del valor de entrada analógica a un valor en coma flotante entre 0,0 y 1,0. Seguidamente, el valor resultante se escalaría a un valor en coma flotante en el rango de las unidades físicas. Para simplificar, las instrucciones KOP siguientes usan valores constantes para los rangos; en realidad, podría elegir el uso de variables:

Segmento 1



Segmento 2



5.4 Tipos de datos

Los tipos de datos se utilizan para determinar el tamaño de un elemento de datos y cómo deben interpretarse los datos. Todo parámetro de instrucción soporta como mínimo un tipo de datos. Algunos parámetros soportan varios tipos de datos. Sitúe el cursor sobre el campo de parámetro de una instrucción para ver qué tipos de datos soporta el parámetro en cuestión.

Un parámetro formal es el identificador en una instrucción que indica la ubicación de los datos que deben utilizarse (ejemplo: la entrada IN1 de una instrucción ADD). Un parámetro actual es la posición de memoria (precedida por el carácter "%") o constante que contiene los datos que debe utilizar la instrucción (ejemplo: %MD400 "Número_de_widgets"). El tipo de datos del parámetro actual definido por el usuario debe concordar con uno de los tipos de datos que soporta el parámetro formal especificado por la instrucción.

Al definir un parámetro actual es preciso indicar una variable (direccionamiento simbólico) o una dirección absoluta (direccionamiento directo). Las variables asocian un nombre simbólico (nombre de variable) con un tipo de datos, área de memoria, offset y comentario. Se pueden crear bien sea en el editor de variables PLC, o bien en la interfaz del bloque (OB, FC, FB y DB). Si se introduce una dirección absoluta que no tenga una variable asociada, es preciso utilizar un tamaño apropiado que coincida con el tipo de datos soportado. Al realizar la entrada se creará una variable predeterminada.

Todos los tipos de datos excepto String. Struct, Array y DTL están disponibles en el editor de variables PLC y en los editores de interfaz de bloque. String, Struct, Array y DTL solo están disponibles en los editores de interfaz de bloque. También es posible introducir un valor de constante para numerosos parámetros de entrada.

- Bit y secuencias de bits (Página 107): Bool (valor booleano o de bit), Byte (valor de byte de 8 bits), Word (valor de byte de 16 bits), DWord (valor de palabra doble de 32 bits)
- Entero (Página 108)
  - USInt (entero de 8 bits sin signo), SInt (entero de 8 bits con signo),
  - UInt (entero de 16 bits sin signo), Int (entero de 16 bits con signo)
  - UDInt (entero de 32 bits sin signo), DInt (entero de 32 bits con signo)

- Real en coma flotante (Página 109): Real (valor en coma flotante o real de 32 bits), LReal (valor en coma flotante o real de 64 bits)
- Fecha y hora (Página 109): Time (valor de temporizador CEI de 32 bits), Date (valor de fecha de 16 bits), TOD (valor de hora de 32 bits), DTL (estructura de fecha y hora de 12 bytes)
- Carácter y cadena (Página 112): Char (carácter individual de 8 bits), String (cadena de longitud variable de hasta 254 caracteres)
- Matriz (Página 114)
- Estructura de datos (Página 115): Struct
- Tipo de datos PLC (Página 115)
- Tipo de datos Variant (Página 116)

Aunque no están disponibles como tipos de datos, las instrucciones de conversión soportan el siguiente formato numérico BCD:

Tabla 5-28 Tamaño y rango del formato BCD

Formato	Tamaño (bits)	Rango numérico	Ejemplos de entrada de constantes
BCD16	16	-999 a 999	123, -123
BCD32	32	-9999999 a 9999999	1234567, -1234567

Consulte también

Tipos de datos soportados (Página 820)

5.4.1 Tipos de datos Bool, Byte, Word y DWord

Tabla 5-29 Tipos de datos bit y secuencia de bits

Tipo de datos	Tamaño en bits	Tipo de número	Rango numérico	Ejemplos de constante	Ejemplos de dirección
Bool	1	Booleano	FALSE o TRUE	TRUE	I1.0 Q0.1 M50.7 DB1.DBX2.3 Nombre_de_variable
		Binario	2#0 o 2#1	2#0	
		Entero sin signo	0 o 1	1	
		Octal	8#0 o 8#1	8#1	
		Hexadecimal	16#0 o 16#1	16#1	
Byte	8	Binario	2#0 a 2#1111_1111	2#1000_1001	IB2 MB10 DB1.DBB4 Nombre_de_variable
		Entero sin signo	0 a 255	15	
		Entero con signo	-128 a 127	-63	
		Octal	8#0 a 8#377	8#17	
		Hexadecimal	B#16#0 a B#16#FF, 16#0 a 16#FF	B#16#F, 16#F	

Tipo de datos	Tamaño en bits	Tipo de número	Rango numérico	Ejemplos de constante	Ejemplos de dirección
Word	16	Binario	2#0 a 2#1111_1111_1111_1111	2#1101_0010_1001_0110	MW10 DB1.DBW2 Nombre_de_variable
		Entero sin signo	0 a 65535	61680	
		Entero con signo	-32768 a 32767	72	
		Octal	8#0 a 8#177_777	8#170_362	
		Hexadecimal	W#16#0 a W#16#FFFF, 16#0 a 16#FFFF	W#16#F1C0, 16#A67B	
DWord	32	Binario	2#0 a 2#1111_1111_1111_1111_1111_1111_1111_1111	2#1101_0100_1111_1110_1000_1100	MD10 DB1.DBD8 Nombre_de_variable
		Entero sin signo*	0 a 4_294_967_295	15_793_935	
		Entero con signo*	-2_147_483_648 a 2_147_483_647	-400000	
		Octal	8#0 a 8#37_777_777_777	8#74_177_417	
		Hexadecimal	DW#16#0000_0000 a DW#16#FFFF_FFFF, 16#0000_0000 a 16#FFFF_FFFF	DW#16#20_F30A, 16#B_01F6	

* El guión bajo "_" es un separador de miles que mejora la legibilidad de los números de más de ocho dígitos.

5.4.2 Tipos de datos de entero

Tabla 5-30 Tipos de datos de entero (U = sin signo, S = simple, D= doble)

Tipo de datos	Tamaño en bits	Rango numérico	Ejemplos de constante	Dirección Ejemplos
USInt	8	0 a 255	78, 2#01001110	MB0, DB1.DBB4, Nombre_variable
SInt	8	128 a 127	+50, 16#50	
UInt	16	0 a 65.535	65295, 0	MW2, DB1.DBW2, Nombre_variable
Int	16	-32.768 a 32.767	30000, +30000	
UDInt	32	0 a 4.294.967.295	4042322160	MD6, DB1.DBD8, Nombre_variable
DInt	32	-2.147.483.648 a 2.147.483.647	-2131754992	

5.4.3 Tipos de datos de real en coma flotante

Los números reales (o en coma flotante) se representan como números de 32 bits de precisión simple (Real) o de 64 bits de precisión doble (LReal) según la norma ANSI/IEEE 7541985. Los números en coma flotante de precisión simple tienen una exactitud de hasta 6 dígitos significativos, en tanto que los de precisión doble tienen una exactitud de hasta 15 dígitos significativos. Al introducir una constante en coma flotante, pueden indicarse como máximo 6 (Real) o 15 (LReal) dígitos significativos para conservar la precisión.

Tabla 5-31 Tipos de datos de real en coma flotante (L=largo)

Tipo de datos	Tamaño en bits	Rango numérico	Ejemplos de constante	Ejemplos de dirección
Real	32	-3.402823e+38 a -1.175 495e-38, ±0, +1.175 495e-38 a +3.402823e+38	123.456, -3.4, 1.0e-5	MD100, DB1.DBD8, Nombre_variable
LReal	64	-1.7976931348623158e+308 a -2.2250738585072014e-308, ±0, +2.2250738585072014e-308 a +1.7976931348623158e+308	12345.123456789e40, 1.2E+40	Nombre_DB.nombre_var Reglas: • No se soporta el direccionamiento directo • Se puede asignar en una tabla de interfaz de OB, FB o FC

Los cálculos que comprenden una serie de valores prolongada, incluyendo números muy grandes y muy pequeños, pueden producir resultados inexactos. Esto puede suceder si los números difieren en 10 a la potencia de x, siendo x > 6 (Real) ó 15 (LReal). Por ejemplo (Real): 100 000 000 + 1 = 100 000 000.

5.4.4 Tipos de datos de fecha y hora

Tabla 5-32 Tipos de datos de fecha y hora

Tipo de datos	Tamaño	Rango	Ejemplos de entrada de constantes
Time	32 bits	T#-24d_20h_31m_23s_648ms a T#24d_20h_31m_23s_647ms Almacenado como: -2.147.483.648 ms hasta +2.147.483.647 ms	T#5m_30s T#1d_2h_15m_30s_45ms TIME#10d20h30m20s630ms 500h10000ms 10d20h30m20s630ms
Date	16 bits	D#1990-1-1 a D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31

Tipo de datos	Tamaño	Rango	Ejemplos de entrada de constantes
Hora	32 bits	TOD#0:0:0.0 a TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (fecha y hora en formato largo)	12 bytes	Mín.: DTL#1970-01-01-00:00:00.0 Máx.: DTL#2262-04-11:23:47:16.854 775 807	DTL#2008-12-16-20:30:20.250

Time

El dato TIME se guarda como entero doble con signo y se interpreta como milisegundos. El formato del editor puede utilizar información para día (d), horas (h), minutos (m), segundos (s) y milisegundos (ms).

No es necesario especificar todas las unidades de tiempo. Son válidos por ejemplo T#5h10s y 500h.

El valor combinado de todos los valores de unidad especificados no puede superar los límites superior o inferior en milisegundos para el tipo de datos Time (-2.147.483.648 ms a +2.147.483.647 ms).

Date

DATE se guarda como valor entero sin signo y se interpreta como número de días agregados a la fecha patrón 01/01/1990 para obtener la fecha específica. El formato del editor debe especifica un año, un mes y un día.

TOD

TOD (TIME_OF_DAY) se guarda como entero doble sin signo y se interpreta como el número en milisegundos desde medianoche para obtener la hora específica del día (medianoche = 0 ms). Deben especificarse hora (24h/día), minuto y segundo. Las fracciones de segundo son opcionales.

DTL

El tipo de datos DTL (fecha y hora largo) utiliza una estructura de 12 bytes para guardar información sobre la fecha y la hora. DTL se puede definir en la memoria temporal de un bloque o en un DB. Debe indicarse un valor para todos los componentes en la columna "Valor inicial" del editor de DB.

Tabla 5-33 Tamaño y rango para DTL

Longitud (bytes)	Formato	Rango de valores	Ejemplo de un valor de entrada
12	Reloj y calendario Año-Mes-Día:Hora:Minuto:Segundo.Nanosegundos	Mín.: DTL#1970-01-01-00:00:00.0 Máx.: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

Todo componente de DTL contiene un tipo de datos y un rango de valores diferentes. El tipo de datos de un valor especificado debe concordar con el tipo de datos de los componentes correspondientes.

Tabla 5-34 Elementos de la estructura DTL

Byte	Componente	Tipo de datos	Rango de valores
0	Año	UINT	1970 a 2554
1			
2	Mes	USINT	1 a 12
3	Día	USINT	1 a 31
4	Día de la semana [1]	USINT	1(domingo) a 7(sábado) [1]
5	Hora	USINT	0 a 23
6	Minuto	USINT	0 a 59
7	Segundo	USINT	0 a 59
8	Nanosegundos	UDINT	0 a 999 999 999
9			
10			
11			

[1] El formato Año-Mes-Día:Hora:Minuto:
Segundo.Nanosegundos no incluye el día de la semana.

5.4.5 Tipos de datos Carácter y Cadena

Tabla 5-35 Tipos de datos Carácter y Cadena

Tipo de datos	Tamaño	Rango	Ejemplos de entrada de constantes
Char	8 bits	16#00 a 16#FF	'A', 't', '@', 'ä', '∑'
WChar	16 bits	16#0000 a 16#FFFF	'A', 't', '@', 'ä', '∑', caracteres asiáticos, caracteres cirílicos y otros
String	n+ 2 bytes	n = (0 a 254 bytes)	"ABC"
WString	n+ 2 palabras	n = (0 a 65534 palabras)	"ä123@XYZ.COM"

Char y WChar

Un Char ocupa un byte en la memoria y guarda un único carácter codificado en formato ASCII, incluidos los códigos de caracteres ASCII ampliados. Un WChar ocupa una palabra en la memoria y puede contener cualquier representación de caracteres de dos bytes.

La sintaxis del editor utiliza un carácter de comilla simple delante y detrás del carácter. Existe la posibilidad de emplear caracteres visibles y de control.

String y WString

La CPU soporta el tipo de datos STRING para almacenar una secuencia de caracteres de un byte. El tipo de datos STRING contiene el número de caracteres total (número de caracteres de la cadena) y el número de caracteres actual. El tipo de datos String ofrece como máximo 256 bytes para almacenar el número máximo de caracteres total (1 byte), el número de caracteres actual (1 byte) y como máximo 254 caracteres en la cadena. Cada byte de un tipo de datos String puede ser cualquier valor entre 16#00 y 16#FF.

El tipo de datos WString ofrece cadenas más largas de valores de una palabra (dos bytes). La primera palabra contiene el recuento de caracteres total máximo; la palabra siguiente contiene el recuento de caracteres total y la cadena siguiente pueden contener hasta 65534 palabras. Cada palabra de un tipo de datos WString puede ser un valor cualquiera entre 16#0000 y 16#FFFF.

Es posible utilizar cadenas literales (constantes) para los parámetros de instrucción del tipo IN entre comillas sencillas. Por ejemplo, 'ABC' es una cadena de tres caracteres que podría utilizarse como entrada para el parámetro IN de la instrucción S_CONV. También es posible crear variables de cadena seleccionando el tipo de datos "String" o "WString" en los editores de la interfaz de bloques OB, FC, FB y DB. En el editor de variables PLC no se pueden crear cadenas.

El tamaño máximo de la cadena se puede especificar en bytes (String) o palabras (WString) introduciendo corchetes después de la palabra clave "String" o "WString" después de haber seleccionado uno de estos tipos de datos en la lista desplegable de tipos de datos. Por ejemplo, "MyString String[10]" especificaría un tamaño máximo de 10 bytes para MyString. Si se omiten los corchetes con un indicador de tamaño máximo, se presupone que el tamaño máximo es de 254 para String y 65534 para WString. "MyWString WString[1000]" especificaría un WString de 1000 palabras.

El ejemplo siguiente define un String con un número máximo de 10 caracteres y un número actual de 3 caracteres. Esto significa que el String contiene actualmente 3 caracteres de un byte, pero que podría ampliarse hasta 10 caracteres de un byte.

Tabla 5-36 Ejemplo de tipo de datos STRING

Número de caracteres total	Número de caracteres actual	Carácter 1	Carácter 2	Carácter 3	...	Carácter 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte 11

El ejemplo siguiente define un WString con un número máximo de 500 caracteres y un número actual de 300 caracteres. Esto significa que el WString contiene actualmente 300 caracteres de una palabra, pero que podría ampliarse hasta 500 caracteres de una palabra.

Tabla 5-37 Ejemplo de tipo de datos WString

Número de caracteres total	Número de caracteres actual	Carácter 1	Caracteres 2 a 299	Carácter 300	...	Carácter 500
500	300	'ä' (16#0084)	Palabras de caracteres ASCII	'M' (16#004D)	...	-
Palabra 0	Palabra 1	Palabra 2	Palabras 3 a 300	Palabra 301	...	Palabra 501

Pueden usarse caracteres de control ASCII en datos Char, WChar, String y WString. La tabla siguiente muestra ejemplos de sintaxis de caracteres de control.

Tabla 5-38 Caracteres de control ASCII válidos

Caracteres de control	Valor hex ASCII (Char)	Valor hex ASCII (WChar)	Función de control	Ejemplos
$L o $l	16#0A	16#000A	Avance línea	'$LText', '$0AText'
$N o $n	16#0A y 16#0D	16#000A y 16#000D	Salto de línea La línea nueva muestra dos caracteres en la cadena.	'$NText', '$0A $0DText'
$P o $p	16#0C	16#000C	Alimentación de página	'$PText', '$0CText'
$R o $r	16#0D	16#000D	Retorno de carro (CR)	'$RText','$0DText'
$T o $t	16#09	16#0009	Tab	'$TText', '$09Text'
$$	16#24	16#0024	Símbolo del dólar	'100$$', '100$24'
$'	16#27	16#0027	Comilla simple	'$'Text$'','$27Text $27'

5.4.6 Tipo de datos ARRAY

Matrices

Se puede crear una matriz que contenga varios elementos del mismo tipo de datos. Las matrices pueden crearse en las interfaces de bloques OB, FC, FB y DB. En el editor de variables PLC no se pueden crear matrices.

Para crear una matriz en la interfaz del bloque, asigne un nombre a la matriz y seleccione el tipo de datos "Array [lo .. hi] of type", modifique luego "lo", "hi" y "type" como se indica a continuación:

- lo - el índice inicial (más bajo) de la matriz
- hi - el índice final (más alto) de la matriz
- type - uno de los tipos de datos, como BOOL, SINT, UDINT

Tabla 5-39 Reglas para el tipo de datos ARRAY

Tipo de datos	Sintaxis de una matriz		
ARRAY	Nombre [index1_min..index1_max, index2_min..index2_max] de <tipo de datos> • Todos los parámetros de la matriz deben tener el mismo tipo de datos. • El índice puede ser negativo, pero el límite inferior debe ser inferior o igual que el límite superior. • Las matrices pueden tener entre una y seis dimensiones. • Las declaraciones multidimensionales mín..máx están separadas por caracteres coma. • No se permiten matrices anidadas ni matrices de matrices. • El tamaño de memoria de una matriz = (tamaño de un elemento * número total de elementos de una matriz)		
	Índice de matriz	**Tipos de datos índice válidos**	**Reglas para índice de matriz**
	Constante o variable	USInt, SInt, UInt, Int, UDInt, DInt	• Límites de valores: -32768 a +32767 • Válido: Constantes y variables mezcladas • Válido: Expresiones constantes • No válido: Expresiones variables

Ejemplo: Declaraciones de matriz	ARRAY[1..20] of REAL	Una dimensión, 20 elementos
	ARRAY[-5..5] of INT	Una dimensión, 11 elementos
	ARRAY[1..2, 3..4] of CHAR	Dos dimensiones, 4 elementos
Ejemplo: Direcciones de matriz	ARRAY1[0]	ARRAY1 elemento 0
	ARRAY2[1,2]	ARRAY2 elemento [1,2]
	ARRAY3[i,j]	Si i =3 y j=4, entonces se direcciona ARRAY3 elemento [3, 4]

5.4.7 Tipo de datos Estructura de datos

Se puede utilizar el tipo de datos "Struct" para definir una estructura de datos formada por otros tipos de datos. El tipo de datos STRUCT puede emplearse para gestionar un grupo de datos de proceso relacionados como una unidad de datos simple. Se asigna un nombre a un tipo de datos STRUCT y la estructura de datos interna se declara en el editor de bloques de datos o un editor de interfaces de bloque.

Las matrices y estructuras también se pueden combinar en una estructura más grande. Se puede anidar una estructura hasta ocho niveles de profundidad. Por ejemplo, se puede crear una estructura de estructuras con matrices.

5.4.8 Tipo de datos PLC

El editor del tipo de datos PLC permite definir estructuras de datos, que pueden usarse varias veces en el programa. Para crear un tipo de datos PLC abra la rama "Tipos de datos PLC" del árbol del proyecto y haga doble clic en el elemento "Añadir nuevo tipo de datos". En el tipo de datos PLC recién creado, haga dos clics individuales para cambiar el nombre predeterminado y un doble clic para abrir el editor del tipo de datos PLC.

Para crear una estructura de tipo de datos PLC personalizada se utilizan los mismos métodos de edición que se utilizan en el editor de bloques de datos. Agregue nuevas filas para los tipos de datos que sean necesarios para crear la estructura de datos deseada.

Cuando se crea un nuevo tipo de datos PLC, su nombre aparece en las listas desplegables de selección de tipo de datos del editor de DB y en el editor de interfaces de bloque lógico.

En principio, los tipos de datos PLC pueden utilizarse de las formas siguientes:

- Como tipo de datos en una interfaz de bloque lógico o en bloques de datos
- Como plantilla para crear varios bloques de datos globales que utilizan la misma estructura de datos
- Como tipo de datos para declaraciones de variables PLC en las áreas de memoria de entrada y salida de la CPU

Por ejemplo, un tipo de datos PLC puede ser una receta de colores mezclados. Así, es posible asignar este tipo de datos PLC a varios bloques de datos. Las variables pueden ajustarse dentro de cada bloque de datos para crear un color específico.

5.4.9 Tipo de datos de puntero Variant

El tipo de datos Variant puede apuntar a variables de diferentes tipos de datos o parámetros. El puntero Variant puede apuntar a estructuras y componentes estructurales individuales. El puntero Variant no ocupa ningún espacio en la memoria.

Tabla 5-40 Propiedades del puntero Variant

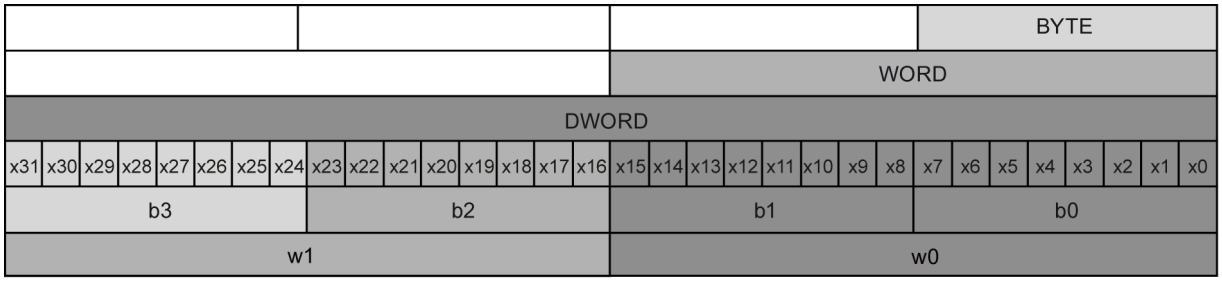
Longitud (bytes)	Representación	Formato	Ejemplo de entrada:
0	Simbólica	Operando	MyTag
		Nombre_DB.Nombre_estruct.nombre_elemento	MiDB.Estruct1.presión1
	Absoluta	Operando	%MW10
		Número_DB.Operando Tipo Longitud	P#DB10.DBX10.0 INT 12

5.4.10 Acceder a un "segmento" de un tipo de datos de variable

Es posible acceder al nivel de bit, byte o palabra de las variables PLC y de bloque de datos en función de su tamaño. A continuación aparece la sintaxis para acceder a un segmento de datos de este tipo:

- "<nombre de la variable PLC>".xn (acceso a bit)
- "<nombre de la variable PLC>".bn (acceso a byte)
- "<nombre de la variable PLC>".wn (acceso a palabra)
- "<nombre del bloque de datos>".<nombre de la variable>.xn (acceso a bit)
- "<nombre del bloque de datos>".<nombre de la variable>.bn (acceso a byte)
- "<nombre del bloque de datos>".<nombre de la variable>.wn (acceso a palabra)

A una variable del tamaño de palabra doble se accede a través de los bits 0 - 31, bytes 0 - 3 o palabras 0 - 1. A una variable del tamaño de palabra se accede a través de los bits 0 - 15, bytes 0 - 1 o palabra 0. A una variable del tamaño de byte se accede a través de los bits 0 - 7 o byte 0. Los segmentos bit, byte y palabra se pueden utilizar dondequiera que bits, bytes y palabras estén previstos como operandos.



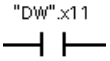
Nota

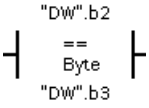
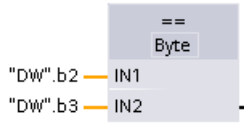
Los tipos de datos válidos a los que se puede acceder por segmento son Byte, Char, Conn_Any, Date, DInt, DWord, Event_Any, Event_Att, Hw_Any, Hw_Device, HW_Interface, Hw_Io, Hw_Pwm, Hw_SubModule, Int, OB_Any, OB_Att, OB_Cyclic, OB_Delay, OB_WHINT, OB_PCYCLE, OB_STARTUP, OB_TIMEERROR, OB_Tod, Port, Rtm, SInt, Time, Time_Of_Day, UDInt, UInt, USInt, y Word. A las variables PLC del tipo Real se puede acceder por segmento pero no a las variables de bloque de datos del tipo Real.

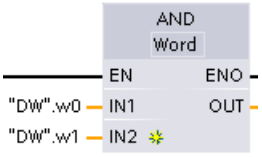
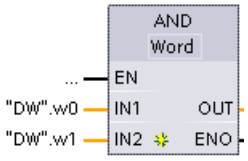
Ejemplos

En la tabla de variables PLC, "DW" es una variable declarada del tipo DWORD. Los ejemplos muestran el acceso al segmento bit, byte y palabra:

	KOP	FUP	SCL

Acceso a bit			`IF "DW".x11 THEN`
			`...`
			`END_IF;`

Acceso a byte			`IF "DW".b2 = "DW".b3`
			`THEN`
			`...`
			`END_IF;`

Acceso a palabra			`out:= "DW".w0 AND` `"DW".w1;`

5.4.11 Acceso a una variable con una superposición de AT

La superposición de la variable AT permite acceder a una variable ya declarada de un bloque con una declaración superpuesta de un tipo de datos diferente. Por ejemplo, se pueden direccionar los distintos bits de una variable de tipo Byte, Word o DWord con una matriz de boolean. Las superposiciones de AT están disponibles para los siguientes tipos de variables:

- variables de un bloque de acceso estándar
- variables remanentes de un boque optimizado

Declaración

Para superponer un parámetro, declare un parámetro adicional directamente después del parámetro que se va a superponer y seleccione el tipo de datos "AT". El editor crea la superposición y, a continuación, se puede elegir el tipo de datos, la estructura o la matriz que se desea utilizar para la superposición.

Ejemplo

En este ejemplo se muestran los parámetros de entrada de un FB de acceso estándar. Una matriz de booleanos es una superposición para la variable de byte B1:

	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

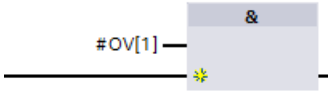
Otro ejemplo es una variable DWord superpuesta con un Struct. Struct incluye una palabra, un byte y dos booleanos:

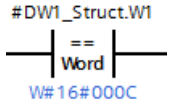
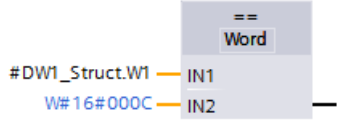
	DW1		DWord	2.0
	DW1_Struct	AT "DW1"	Struct	2.0
	W1		Word	0.0
	B1		Byte	2.0
	BO1		Bool	3.0
	BO2		Bool	3.1

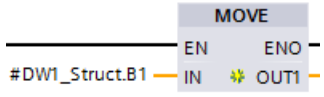
La columna Offset de la interfaz de bloque muestra la ubicación de los tipos de datos superpuestos en relación con la variable original.

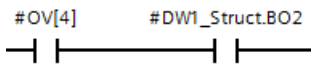
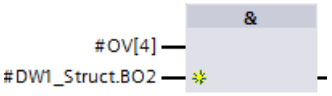
Los tipos de superposición se pueden direccionar directamente en la lógica del programa:

KOP	FUP	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;`

Reglas

- La superposición de variables es posible en bloques FB y FC con acceso estándar (no optimizado).
- En bloques FB y FC optimizados, la superposición de variables es posible para cualquier variable remanente.
- Se pueden superponer parámetros de todos los tipos de bloques y secciones de declaración.
- Un parámetro superpuesto se puede utilizar como cualquier otro parámetro de bloque.
- No se pueden superponer parámetros de tipo VARIANT.
- El tamaño del parámetro que se superpone debe ser menor o igual que el tamaño del parámetro superpuesto.
- Debe declarar la variable que se superpone inmediatamente después de la variable que queda superpuesta y seleccionar la palabra clave "AT" como selección de tipo de datos inicial.

5.5 Utilizar una Memory Card

Nota

La CPU solo admite las SIMATIC Memory Cards (Página 1424) preformateadas.

Antes de copiar cualquier programa en la SIMATIC Memory Card formateada, borre todo programa almacenado previamente en ella.

Puede utilizar una SIMATIC Memory Card de las siguientes formas:

- Utilice una SIMATIC Memory Card como tarjeta de transferencia o de programa. Las tarjetas de transferencia y de programa contienen todos los bloques lógicos y de datos, los objetos tecnológicos y la configuración del dispositivo. Las tarjetas de transferencia y de programa **no** contienen, por ejemplo, tablas de forzado permanente, tablas de observación o tablas de variables PLC.
  - Utilice una tarjeta de transferencia (Página 123) para copiar un programa a la memoria de carga interna de la CPU sin usar STEP 7.
  - Utilice una tarjeta de transferencia vacía para acceder a una CPU protegida por contraseña si ha perdido u olvidado la contraseña (Página 135).
  - Utilice una tarjeta de programa (Página 125) como memoria de carga externa para la CPU.
- Utilice una SIMATIC Memory Card para descargar actualizaciones de firmware (Página 132).
- Utilice una SIMATIC Memory Card para establecer o cambiar la contraseña para proteger datos de configuración confidenciales del PLC.

5.5.1 Insertar una Memory Card en la CPU

ATENCIÓN

Cómo proteger la Memory Card y la ranura frente a la descarga electroestática

Las descargas electroestáticas pueden deteriorar la Memory Card o la ranura para tarjetas en la CPU.

Al manejar la Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar una muñequera antiestática. Guarde la Memory Card en una caja conductiva.



Asegúrese de que la Memory Card no está protegida contra escritura. Deslice el interruptor de protección fuera de la posición "Lock".

Tenga en cuenta que si se inserta en la CPU una Memory Card protegida contra escritura, STEP 7 mostrará un aviso de diagnóstico durante el próximo arranque para avisar de tal hecho. La CPU arranca sin fallos, pero las instrucciones que contengan recetas o registros de datos, por ejemplo, devolverán errores si la tarjeta está protegida contra escritura.

⚠ ADVERTENCIA

Verifique que la CPU no está ejecutando ningún proceso en ese momento antes de insertar la Memory Card.

Si inserta una Memory Card (tanto si se configura como tarjeta de programa o como tarjeta de transferencia o tarjeta de actualización de firmware) en una CPU que está en funcionamiento, la CPU pasará al estado operativo STOP, lo que podría provocar interrupciones en el proceso que causen la muerte o lesiones corporales graves.

Antes de enchufar o desenchufar una Memory Card, asegúrese siempre de que la CPU no está controlando activamente una máquina o un proceso. Prevea siempre un circuito de parada de emergencia para la aplicación o el proceso.

Nota

No enchufe tarjetas de transferencia de programa V3.0 en CPU S7-1200 V4.x.

Las tarjetas de transferencia de programa de la versión 3.0 no son compatibles con CPU S7-1200 de la versión 4.x. Insertar una tarjeta de memoria que contiene un programa V3.0 provoca un error de CPU.

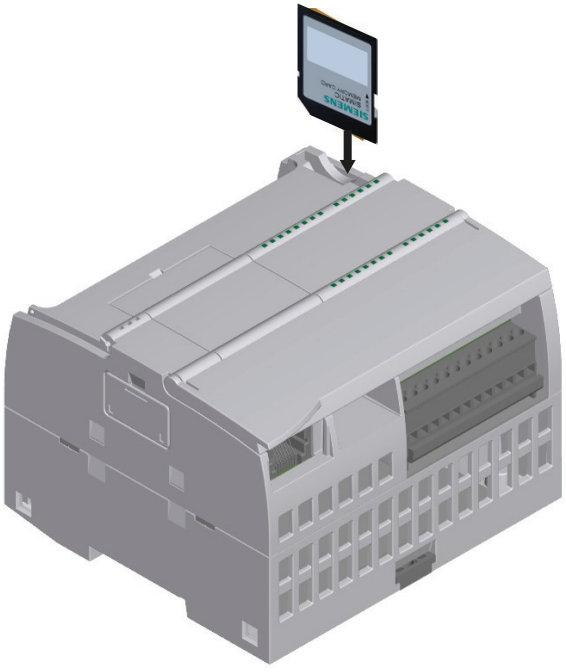
Si enchufa una tarjeta de transferencia de programa (Página 123) de una versión no válida, deberá desenchurar la tarjeta y realizar una transición de STOP a RUN, un borrado total (MRES) o un ciclo de encendido. Después de sacar la CPU de la condición de error, puede descargar un programa de CPU V4.x válido.

Para transferir un programa V3.0 a un programa V4.x, deberá utilizar el TIA Portal para cambiar el dispositivo en la configuración hardware.

Nota

Si se inserta una Memory Card estando la CPU en estado operativo STOP, el búfer de diagnóstico mostrará un mensaje de que se ha iniciado la evaluación de la Memory Card. La CPU evaluará la Memory Card la próxima vez que la CPU conmute al estado operativo RUN, se realice un borrado total de la CPU (MRES) o se desconecte y vuelva a conectar la alimentación de la CPU.

Tabla 5-41 Insertar una Memory Card

	Para insertar una Memory Card, abra la tapa superior de la CPU e inserte la Memory Card en la ranura. Un conector de trinquete facilita la inserción y extracción. La Memory Card está diseñada de manera que pueda insertarse en un único sentido.

Comportamiento de la CPU cuando se inserta una Memory Card

Cuando se inserta una Memory Card en la CPU, esta ejecuta los pasos siguientes:

1. Transición al estado operativo STOP (si todavía no estaba en STOP)
2. Ofrece una de las opciones siguientes:
   - Desconexión y conexión
   - Transición al estado operativo RUN
   - Realizar un borrado total
3. Revisar la tarjeta

Forma en que la CPU revisa la Memory Card

Si la CPU no está configurada para "Desactivar la copia de la memoria de carga interna a la memoria de carga externa" en las propiedades de protección de la configuración de dispositivos (Página 167), la CPU determina el tipo de Memory Card que se ha insertado:

- **Memory Card vacía**: una Memory Card vacía no tiene un archivo de tareas (S7_JOB.S7S). Si se enchufa una Memory Card vacía, la CPU agrega un archivo de tareas de programa. Seguidamente, copia la memoria de carga interna en la memoria de carga externa (el archivo de programa en la Memory Card) y borra la memoria de carga interna.
- **Tarjeta de programa en blanco**: una tarjeta de programa en blanco tiene un archivo de trabajo que está vacío. En este caso, la CPU copia la memoria de carga interna en la memoria de carga externa (el archivo de programa en la Memory Card) y borra la memoria de carga interna.

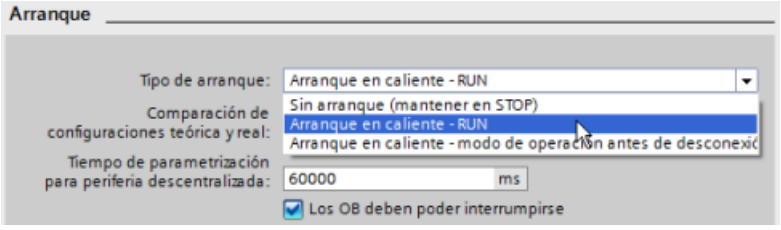
Si la CPU está configurada para "Desactivar la copia de la memoria de carga interna a la memoria de carga externa" en las propiedades de protección de la configuración de dispositivos, la CPU presenta el comportamiento siguiente:

- **Memory Card vacía**: una Memory Card vacía no tiene un archivo de tareas (S7_JOB.S7S). Si se inserta una Memory Card vacía, la CPU no hace nada. No crea un archivo de trabajo de programa ni copia la memoria de carga interna en la memoria de carga externa (el archivo de programa en la Memory Card). Tampoco borra la memoria de carga interna.
- **Tarjeta de programa en blanco**: una tarjeta de programa en blanco tiene un archivo de trabajo que está vacío. En este caso, la CPU no hace nada. No copia la memoria de carga interna en la memoria de carga externa (el archivo de programa en la Memory Card). Tampoco borra la memoria de carga interna.

Si se inserta en la CPU una tarjeta de programa (Página 125), tarjeta de transferencia (Página 123) o una tarjeta que contenga una actualización de firmware (Página 132), el ajuste de configuración para "Desactivar la copia de la memoria de carga interna a la memoria de carga externa" no afecta al modo en que la CPU revisa la Memory Card.

5.5.2 Configurar los parámetros de arranque de la CPU antes de copiar el proyecto en la Memory Card

Cuando un programa se copia en una tarjeta de transferencia o de programa, incluye los parámetros de arranque de la CPU. Antes de copiar el programa en la Memory Card, asegúrese de que se ha configurado el estado operativo de la CPU posterior a la desconexión y conexión de la alimentación. Seleccione si la CPU debe arrancar en estado operativo STOP o RUN, o bien en el estado operativo anterior (a la desconexión y conexión de la alimentación).



5.5.3 Utilizar la Memory Card como tarjeta de "Transferencia"

ATENCIÓN
Cómo proteger la Memory Card y la ranura frente a la descarga electroestática Las descargas electroestáticas pueden deteriorar la Memory Card o la ranura para tarjetas en la CPU. Manipule la Memory Card con seguridad de una de las siguientes maneras: • establezca contacto con una superficie conductiva puesta a tierra. • lleve una muñequera antiestática siempre que manipule una Memory Card. Guarde la Memory Card en una caja conductiva.

Crear una tarjeta de transferencia

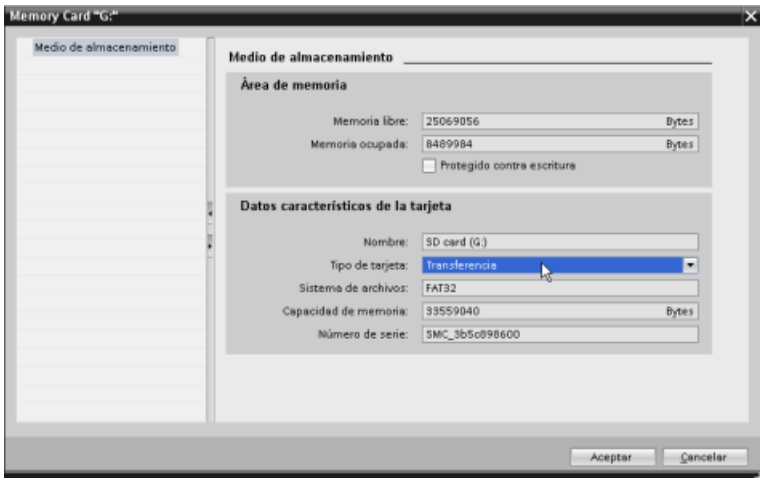
Recuerde que es necesario configurar los parámetros de arranque de la CPU (Página 122) antes de copiar un programa en la tarjeta de transferencia. Para crear una tarjeta de transferencia, proceda del siguiente modo:

1. Inserte una SIMATIC Memory Card vacía que no esté protegida contra escritura en el lector/grabadora de tarjetas SD conectado/a al PC. (Si la tarjeta está protegida contra escritura, deslice el interruptor de protección fuera de la posición de bloqueo).
   Si se quiere reutilizar una SIMATIC Memory Card que contenga un programa de usuario, registros de datos, recetas o una actualización de firmware, **es necesario** borrar los archivos antes de reutilizarla. Utilice Windows Explorer para visualizar el contenido de la Memory Card y borre el archivo "S7_JOB.S7S" y cualquier carpeta que contenga (p. ej. "SIMATIC.S7S" o "FWUPDATE.S7S", "DataLogs", y "Recipes").

ATENCIÓN
NO borrar los archivos ocultos "__LOG__" y "crdinfo.bin" de la Memory Card. Los archivos "__LOG__" y "crdinfo.bin" son necesarios para la Memory Card. Si borra estos archivos, no podrá utilizar la Memory Card con la CPU.

2. En el árbol del proyecto (vista del proyecto), expanda la carpeta "SIMATIC Card Reader" y seleccione el lector de tarjetas deseado.
3. Para abrir el cuadro de diálogo "Memory Card", haga clic con el botón derecho del ratón en la letra de unidad correspondiente a la Memory Card en el lector de tarjetas y elija el comando "Propiedades" del menú contextual.

4. En el cuadro de diálogo "Memory Card", seleccione "Transferencia" en el menú desplegable "Tipo de tarjeta".
   A continuación, STEP 7 crea una tarjeta de transferencia vacía. Si está creando una tarjeta de transferencia vacía p. ej. para realizar una recuperación tras olvidar la contraseña de la CPU (Página 135), extraiga la tarjeta de transferencia del lector de tarjetas.



5. Agregue el programa seleccionando la CPU (p. ej. PLC_1 [CPU 1214C DC/DC/DC]) en el árbol del proyecto y arrastrándola hasta la Memory Card. (Como alternativa, copie la CPU e insértela en la Memory Card.) Cuando la CPU se copia en la Memory Card se abre el diálogo "Cargar vista preliminar".
6. En el diálogo "Cargar vista preliminar", haga clic en el botón "Cargar" para copiar la CPU en la Memory Card.
7. Cuando aparezca un mensaje indicando que la CPU (el programa) se ha cargado sin errores, haga clic en el botón "Finalizar".

Utilizar una tarjeta de transferencia

⚠ ADVERTENCIA

Verifique que la CPU no está ejecutando ningún proceso en ese momento antes de insertar la Memory Card.

La inserción de una Memory Card provocará el paso de la CPU a STOP, lo que podría afectar al funcionamiento de un proceso online o una máquina. El manejo inesperado de un proceso o una máquina podría provocar lesiones o incluso la muerte de personas y/o daños materiales.

Antes de insertar una tarjeta de transferencia, asegúrese siempre de que la CPU esté en estado operativo STOP y de que el proceso esté en estado seguro.

Nota

No inserte tarjetas de transferencia de programa V3.0 en CPU de modelos posteriores.

Las tarjetas de transferencia de programa de la versión 3.0 no son compatibles con CPU S7-1200 de modelos posteriores. Insertar una tarjeta de memoria que contiene un programa V3.0 provoca un error de CPU.

Si inserta una tarjeta de transferencia de programa de una versión no válida, extraiga la tarjeta, realice una transición de STOP a RUN, un reset de memoria (MRES) o un ciclo de encendido. Después de recuperar la CPU de la condición de error, puede descargar un programa de CPU válido.

Para transferir el programa a una CPU, proceda del siguiente modo:

1. Inserte la tarjeta de transferencia en la CPU (Página 120). Si la CPU está en RUN, pasará a estado operativo STOP. El LED de mantenimiento (MAINT) parpadea para indicar que es necesario revisar la Memory Card. En este punto, el programa aún está en la CPU.
2. Desconecte y vuelva a conectar la CPU para revisar la Memory Card. Otros métodos alternativos de rearrancar la CPU consisten en hacer una transición de STOP a RUN o inicializar la memoria (MRES) desde STEP 7.
3. Tras el rearranque, la CPU evalúa la memoria de carga y copia el programa en la memoria de carga interna de la CPU.
   El LED RUN/STOP parpadea de forma alterna en verde y amarillo para indicar que el programa se está copiando. Cuando el LED RUN/STOP se enciende (en amarillo permanente) y el LED MAINT parpadea (amarillo), el proceso de copia ha finalizado. Ahora ya se puede desenchufar la Memory Card.
4. Vuelva a arrancar la CPU (ya sea restableciendo la alimentación o con los métodos de rearranque alternativos) para evaluar el nuevo programa transferido a la memoria de carga interna.

La CPU pasa entonces al modo de arranque (RUN o STOP) configurado para el proyecto.

Nota

Extraiga la tarjeta de transferencia antes de cambiar la CPU a estado operativo RUN.

5.5.4 Utilizar la Memory Card como tarjeta de "Programa"

ATENCIÓN
Las descargas electrostáticas pueden deteriorar la SIMATIC Memory Card o la ranura para tarjetas de la CPU. Al manejar la SIMATIC Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra y/o llevar una muñequera antiestática. Guarde la SIMATIC Memory Card en una caja conductiva.



Asegúrese de que la SIMATIC Memory Card no está protegida contra escritura. Deslice el interruptor de protección fuera de la posición "Lock".

Antes de copiar elementos de programa en la tarjeta de programa, borre todo programa almacenado previamente en la SIMATIC Memory Card.

Crear una tarjeta de programa

Si se utiliza como tarjeta de programa, la SIMATIC Memory Card es la memoria de carga externa de la CPU. Si se extrae la tarjeta de programa, la memoria de carga interna de la CPU estará vacía.

Nota

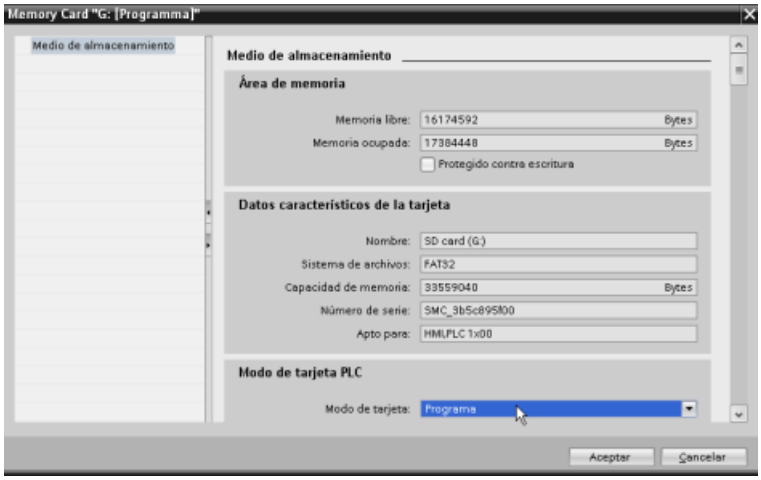
Si una SIMATIC Memory Card vacía se inserta en la CPU y se evalúa desconectando y conectando la alimentación de la CPU, cambiando el estado operativo de STOP a RUN o realizando un borrado total (MRES), el programa y los valores de forzado permanente contenidos en la memoria de carga interna de la CPU se copiarán en la SIMATIC Memory Card. (La SIMATIC Memory Card se convierte entonces en una tarjeta de programa.) Una vez finalizada la operación de copia, se borrará el programa en la memoria de carga interna de la CPU. La CPU pasa entonces al modo de arranque (RUN o STOP) configurado.

Recuerde siempre que es necesario configurar los parámetros de arranque de la CPU (Página 122) antes de copiar el proyecto en la tarjeta de programa. Para crear una tarjeta de programa, proceda del siguiente modo:

1. Enchufe una SIMATIC Memory Card vacía que no esté protegida contra escritura en el lector/grabadora de tarjetas SD conectado/a al PC. (Si la tarjeta está protegida contra escritura, deslice el interruptor de protección fuera de la posición de bloqueo).
   Si se quiere reutilizar una SIMATIC Memory Card que contenga un programa de usuario, registros de datos, recetas o una actualización de firmware, **es necesario** borrar los archivos antes de reutilizarla. Utilice Windows Explorer para visualizar el contenido de la SIMATIC Memory Card y borrar los siguientes archivos y carpetas, si existen:
   - S7_JOB.S7S
   - SIMATIC.S7S
   - FWUPDATE.S7S
   - DataLogs
   - Recipes
   - UserFles

ATENCIÓN
NO borrar los archivos ocultos "__LOG__" y "crdinfo.bin" de la SIMATIC Memory Card. Los archivos "__LOG__" y "crdinfo.bin" son necesarios para la SIMATIC Memory Card. Si borra estos archivos, no podrá utilizar la SIMATIC Memory Card con la CPU.

2. En el árbol del proyecto (vista del proyecto), expanda la carpeta "Card Reader/USB memory" y seleccione el lector de tarjetas deseado.
3. Para abrir el cuadro de diálogo "Memory Card", haga clic con el botón derecho del ratón en la letra de unidad correspondiente a la Memory Card en el lector de tarjetas y elija el comando "Propiedades" del menú contextual.
4. En el cuadro de diálogo "Memory Card", seleccione "Programa" en el menú contextual.



5. Agregue el programa seleccionando la CPU (p. ej. PLC_1 [CPU 1214C DC/DC/DC]) en el árbol del proyecto y arrastrándola hasta la SIMATIC Memory Card. (Como alternativa, copie la CPU e insértela en la SIMATIC Memory Card.) Cuando la CPU se copia en la SIMATIC Memory Card se abre el cuadro de diálogo "Vista preliminar Carga".
6. En el cuadro de diálogo "Vista preliminar Carga", haga clic en el botón "Cargar" para copiar la CPU en la SIMATIC Memory Card.
7. Cuando aparezca un mensaje indicando que la CPU (el programa) se ha cargado sin errores, haga clic en el botón "Finalizar".

Utilizar una tarjeta de programa como memoria de carga para la CPU

⚠ ADVERTENCIA

Riesgos asociados con la inserción de una tarjeta de programa

Verifique que la CPU no está ejecutando ningún proceso en ese momento antes de insertar la SIMATIC Memory Card.

La inserción de una SIMATIC Memory Card provocará el paso de la CPU a STOP, lo que podría afectar al funcionamiento de un proceso online o una máquina. El manejo inesperado de un proceso o una máquina podría provocar lesiones o incluso la muerte de personas y/o daños materiales.

Antes de insertar una SIMATIC Memory Card, asegúrese siempre de que la CPU está offline y en un estado seguro.

Para utilizar una tarjeta de programa en la CPU, proceda del siguiente modo:

1. Inserte la tarjeta de programa en la CPU. Si la CPU está en RUN, pasará a estado operativo STOP. El LED de mantenimiento (MAINT) parpadea para indicar que es necesario comprobar la SIMATIC Memory Card.
2. Desconecte y vuelva a conectar la CPU para revisar la SIMATIC Memory Card. Otros métodos alternativos de rearrancar la CPU consisten en hacer una transición de STOP a RUN o inicializar la memoria (MRES) desde STEP 7.
3. Una vez que la CPU ha rearrancado y evaluado la tarjeta de programa, borra su memoria de carga interna.

La CPU pasa entonces al modo de arranque (RUN o STOP) configurado para ella.

La tarjeta de programa debe permanecer en la CPU. Si se extrae la tarjeta de programa, la memoria de carga interna de la CPU no contendrá ningún programa.

⚠ ADVERTENCIA

Riesgos asociados con la extracción de una tarjeta de programa

Si se extrae la tarjeta de programa, la CPU perderá su memoria de carga externa y generará un error. La CPU pasa a estado operativo STOP y el LED de error parpadea.

Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

No extraiga la tarjeta de programa sin saber que está extrayendo el programa de la CPU.

Vida útil de una SIMATIC Memory Card

La vida útil de una SIMATIC Memory Card depende de diferentes factores, por ejemplo:

- Número de operaciones de escritura y borrado por bloque de memoria
- Número de bytes escritos
- Influencias externas, como la temperatura ambiente

Nota

Efecto de las operaciones de escritura y borrado sobre la vida útil de la SIMATIC Memory Card

Las operaciones de escritura o borrado, especialmente las reiteradas (cíclicas), reducen la vida útil de la SIMATIC Memory Card.

La ejecución cíclica de las acciones siguientes reduce la vida útil de la SIMATIC Memory Card en función del número de operaciones de escritura y de los datos:

- Administración de registros de datos (por ejemplo, DataLogWrite)
- Administración de recetas (por ejemplo, RecipeExport)
- Llamadas de función de sistema (SFC) que escriben en el sistema de archivos o lo borran (por ejemplo, WRIT_DBL, CREATE)
- Bloques de función de sistema (SFB) que escriben en el sistema de archivos o lo borran (por ejemplo, FileWriteC, FileDelete)
- Cualquier otra acción cíclica que modifique datos en el la memoria permanente (por ejemplo, Tracing, SET-TimeZone)

5.5.5 Uso de una SIMATIC Memory Card para proteger datos de configuración confidenciales del PLC

Puede utilizar una SIMATIC Memory Card para establecer o cambiar la contraseña para la protección de datos de configuración confidenciales del PLC.

ATENCIÓN
Cómo proteger la SIMATIC Memory Card y la ranura frente a descargas electrostáticas Las descargas electrostáticas pueden deteriorar la SIMATIC Memory Card o la ranura para tarjetas de la CPU. Cuando maneje la SIMATIC Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra o llevar una muñequera antiestática. Guarde la SIMATIC Memory Card en una caja conductiva.

ATENCIÓN
No utilice la función de formateo de Windows ni cualquier otra función de formateo para reformatear la SIMATIC Memory Card. Si se reformatea una SIMATIC Memory Card con la función de formateo de Microsoft Windows, la memoria ya no podrá utilizarse en una CPU S7-1200.

Riesgos asociados al proceso de puesta fuera de servicio

Las CPU S7-1200 no soportan el borrado seguro de la SIMATIC Memory Card y de la flash interna. Por tanto, para evitar la pérdida de información confidencial o propietaria deberá eliminar la CPU y la SIMATIC Memory Card de forma segura durante el proceso de puesta fuera de servicio.

Cómo crear una SIMATIC Memory Card con la contraseña para la protección de datos de configuración confidenciales del PLC

Para crear la SIMATIC Memory Card con esta contraseña, siga los siguientes pasos:

1. Enchufe una SIMATIC Memory Card vacía que no esté protegida contra escritura en el lector/grabadora de tarjetas SD conectado/a al PC. Si la tarjeta está protegida contra escritura, deslice el interruptor de protección fuera de la posición de bloqueo.
   Es posible reutilizar una SIMATIC Memory Card que contenga un programa de usuario o una actualización de firmware, pero es necesario borrar algunos archivos de la SIMATIC Memory Card. Para reutilizar una SIMATIC Memory Card, debe borrar el archivo "S7_JOB.S7S" antes de crear el archivo Protección de datos de configuración confidenciales del PLC. Utilice Windows Explorer para visualizar el contenido de la SIMATIC Memory Card y borrar el archivo "S7_JOB.S7S" y las carpetas.

ATENCIÓN
NO borrar los archivos ocultos "__LOG__" y "crdinfo.bin" de la SIMATIC Memory Card. Los archivos "__LOG__" y "crdinfo.bin" son necesarios para la SIMATIC Memory Card. Si borra estos archivos, no podrá utilizar la SIMATIC Memory Card con la CPU.

2. Cree un archivo en la raíz de la SIMATIC Memory Card con el nombre "S7_JOB.S7S". Abra el archivo con un editor de textos y escriba lo siguiente: SET_PWD.

3. Cree una carpeta en la raíz de la SIMATIC Memory Card con el nombre "SET_PWD.S7S.".
4. En la carpeta "SET_PWD.S7S", cree un archivo de texto con el nombre "PWD.TXT". El archivo debe llamarse "PWD.TXT". Introduzca su contraseña para la protección de datos de configuración confidenciales del PLC como texto del archivo. El archivo debe contener una sola línea de texto representando la contraseña para la protección de datos de configuración confidenciales del PLC. Siga las reglas de STEP 7 para contraseñas para crear la contraseña utilizando los siguientes caracteres:
   - 0123456789
   - A...Z a...z
   - !#$%&()*+,-./:;<=>?@ [\]_{|}~^
5. Para borrar del PLC la contraseña para la protección de datos de configuración confidenciales del PLC, el archivo debe estar vacío.
6. Extraiga la tarjeta de forma segura del lector/grabador de tarjetas.

Cómo establecer la contraseña para la protección de datos de configuración confidenciales del PLC

Para establecer la contraseña para la protección de datos de configuración confidenciales del PLC, siga los siguientes pasos:

ATENCIÓN
Verifique que la CPU no está ejecutando ningún proceso antes de establecer la contraseña para la protección de datos de configuración confidenciales del PLC. Al establecer la contraseña para la protección de datos de configuración confidenciales del PLC puede ocurrir que el programa PLC no se cargue. Antes de insertar la SIMATIC Memory Card, asegúrese de que la CPU esté offline y en un estado seguro.

1. Inserte la SIMATIC Memory Card en la CPU. Si la CPU está en RUN, pasará al estado operativo STOP. El LED de mantenimiento (MAINT) parpadea para indicar que es necesario comprobar la SIMATIC Memory Card.
2. Desconecte y vuelva a conectar la alimentación de la CPU para iniciar el proceso. Una vez que la CPU ha rearrancado, se establece la contraseña para la protección de datos de configuración confidenciales del PLC. Este proceso finaliza cuando el LED RUN/STOP se enciende (en amarillo permanente) y el LED MAINT parpadea. A continuación se debe desenchufar la SIMATIC Memory Card.
3. Tras extraer la SIMATIC Memory Card, rearranque nuevamente la CPU para empezar a usar la nueva contraseña para la protección de datos de configuración confidenciales del PLC.

Si el programa de usuario existente requiere una contraseña diferente para la protección de datos de configuración confidenciales del PLC , no se cargará tras el rearranque. Debe borrar el programa existente y descargar un programa que use la contraseña para la protección de datos de configuración confidenciales del PLC que haya establecido en los pasos anteriores.

Si el programa existente exige la contraseña proporcionada para la protección de datos de configuración confidenciales del PLC, el PLC puede pasar a RUN con la configuración del proyecto.

5.5.6 Actualización de firmware

Puede usar una SIMATIC Memory Card para actualizar el firmware.

ATENCIÓN
Cómo proteger la SIMATIC Memory Card y la ranura frente a descargas electrostáticas Las descargas electrostáticas pueden deteriorar la SIMATIC Memory Card o la ranura para tarjetas de la CPU. Cuando maneje la SIMATIC Memory Card deberá estar en contacto con una superficie conductiva puesta a tierra o llevar una muñequera antiestática. Guarde la SIMATIC Memory Card en una caja conductiva.

Utilice una SIMATIC Memory Card para descargar actualizaciones de firmware del Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/). Desde esta página web, navegue hasta "Descargas". Busque el tipo de módulo que necesita actualizar.

Como alternativa, puede acceder directamente a la página web de descargas de S7-1200 (https://support.industry.siemens.com/cs/ww/es/ps/13683/dl).

Nota

No se puede actualizar una CPU S7-1200 V3.0 o anterior a S7-1200 V4.0 (o posterior) mediante la actualización de firmware.

También es posible actualizar el firmware utilizando uno de los métodos siguientes:

- Las herramientas online y de diagnóstico de STEP 7 (Página 1204)
- La página web estándar "Información del módulo" del servidor web (Página 867)
- La SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/es/view/98161300)

ATENCIÓN
No utilice la función de formateo de Windows ni cualquier otra función de formateo para reformatear la SIMATIC Memory Card. Si se vuelve a formatear una SIMATIC Memory Card con la utilidad de formateo de Microsoft Windows, entonces la tarjeta de memoria ya no podrá utilizarse en una CPU S7-1200.

Para descargar la actualización de firmware en la SIMATIC Memory Card, siga los siguientes pasos:

1. Enchufe una SIMATIC Memory Card vacía que no esté protegida contra escritura en el lector/grabadora de tarjetas SD conectado/a al PC. (Si la tarjeta está protegida contra escritura, deslice el interruptor de protección fuera de la posición de bloqueo).
   Es posible reutilizar una SIMATIC Memory Card que contenga un programa de usuario u otra actualización de firmware. Para evitar cualquier confusión, también debe borrar, si existen, los archivos S7_JOB.SYS, SIMATIC.S7S y FWUPDATE.S7S.

   ATENCIÓN

   NO borrar los archivos ocultos "__LOG__" y "crdinfo.bin" de la SIMATIC Memory Card.

   Los archivos "__LOG__" y "crdinfo.bin" son necesarios para la SIMATIC Memory Card. Si borra estos archivos, no podrá utilizar la SIMATIC Memory Card con la CPU.

2. Seleccione el archivo zip para actualizar el firmware que corresponde al módulo utilizado y descárguelo en su equipo. Haga doble clic en el archivo, indique la ruta de destino del archivo de tal modo que sea el directorio raíz de la SIMATIC Memory Card e inicie el proceso de extracción. Una vez finalizada la extracción, el directorio raíz (carpeta) de la SIMATIC Memory Card contendrá un directorio "FWUPDATE.S7S" y el archivo "S7_JOB.S7S".
3. Extraiga la tarjeta de forma segura del lector/grabador de tarjetas.

Para instalar la actualización de firmware, proceda del siguiente modo:

⚠ ADVERTENCIA

Antes de instalar la actualización de firmware, compruebe que la CPU no esté ejecutando activamente ningún proceso.

Al instalar la actualización de firmware, la CPU pasará a STOP, lo que puede afectar la operación de un proceso online o de una máquina. El manejo inesperado de un proceso o una máquina podría provocar lesiones o incluso la muerte de personas y/o daños materiales.

Antes de insertar la SIMATIC Memory Card, asegúrese de que la CPU esté offline y en un estado seguro.

1. Inserte la SIMATIC Memory Card en la CPU. Si la CPU está en RUN, pasará al estado operativo STOP. El LED de mantenimiento (MAINT) parpadea para indicar que es necesario comprobar la SIMATIC Memory Card.
2. Apague y vuelva a encender la alimentación de la CPU para iniciar la actualización de firmware. Otros métodos alternativos de rearrancar la CPU consisten en hacer una transición de STOP a RUN o inicializar la memoria (MRES) desde STEP 7.

   Nota

   Con el fin de completar la actualización de firmware para el módulo, hay que asegurarse de que la alimentación externa de 24 V DC hacia el módulo permanece conectada.

   Después de rearrancar la CPU, inicia la actualización del firmware. El LED RUN/STOP parpadea de forma alterna en verde y amarillo, indicando que la actualización se está copiando. Cuando el LED RUN/STOP se enciende (en amarillo permanente) y el LED MAINT parpadea, el proceso de copia ha finalizado. A continuación se debe desenchufar la SIMATIC Memory Card.
3. Después de desenchufar la SIMATIC Memory Card, rearranque la CPU (restableciendo la alimentación o bien aplicando otro método) para cargar el nuevo firmware.

El programa de usuario y la configuración hardware no se ven afectados por la actualización de firmware. Al poner en marcha la CPU, esta adopta el estado de arranque configurado. (Si el modo de arranque de la CPU estaba configurado como "Modo de rearranque en caliente antes de POWER OFF", la CPU se encontrará en el estado operativo STOP porque el último estado de la CPU era STOP).

Durante la actualización, el procedimiento de actualización del firmware ignora los archivos UPD que no correspondan a ningún módulo hardware de la estación. Esto permite crear una SIMATIC Memory Card maestra para la actualización del firmware para actualizar todas las estaciones de CPU S7-1200 en su instalación. No se efectúa ninguna entrada en el búfer de diagnóstico para identificar los archivos UPD ignorados. Esto evita que el búfer de diagnóstico se tapone con entradas generalmente irrelevantes que pueden ocultar las entradas del proceso de actualización de interés para usted. Por ejemplo, en el búfer de diagnóstico se efectúa una entrada cuando se intenta actualizar el firmware, independientemente de si la actualización finaliza correctamente o no. En el búfer de diagnóstico puede buscar entonces cualquier anomalía inesperada sin que se vea estorbado por entradas extrañas.

Todos los intentos de actualizar el firmware se registran en el búfer de diagnóstico independientemente de su resultado. Por ejemplo, si una operación de actualización del firmware finaliza correctamente, se registra el aviso correspondiente en el búfer de diagnóstico. Del mismo modo, si una operación de actualización del firmware falla, esto también queda registrado con una explicación.

En una tarjeta de memoria de actualización de firmware solo está permitido un archivo UPD de una referencia (MLFB). Por ejemplo, si el actualizador encuentra archivos UPD con referencias (MLFB) duplicadas en la tarjeta de memoria de actualización de firmware, la actualización de firmware no se efectúa y una entrada del búfer de diagnóstico indica esta condición. Esto puede ocurrir cuando los archivos UPD tienen la misma referencia (MLFB) pero distintas versiones.

5.6 Recuperación si se olvida la contraseña

Si se ha olvidado la contraseña de una CPU protegida por contraseña, es preciso utilizar una tarjeta de transferencia vacía para borrar el programa protegido por contraseña. La tarjeta de transferencia vacía borra la memoria de carga interna de la CPU. Entonces es posible cargar un programa de usuario nuevo desde STEP 7 a la CPU.

Encontrará más información sobre cómo crear y utilizar una tarjeta de transferencia vacía en el apartado Tarjeta de transferencia (Página 123).

⚠ ADVERTENCIA

Verifique que la CPU no está ejecutando ningún proceso en ese momento antes de insertar la Memory Card.

Si se inserta una tarjeta de transferencia en una CPU que esté funcionando, la CPU pasará a STOP. Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

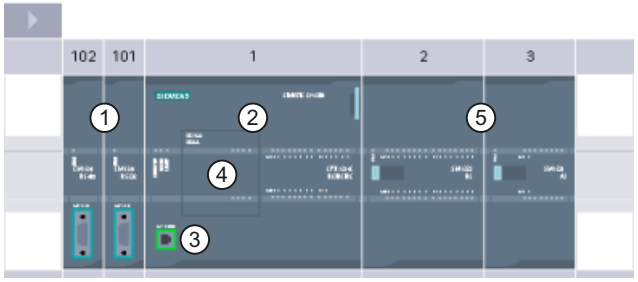
Antes de insertar una tarjeta de transferencia, asegúrese siempre de que la CPU esté en estado operativo STOP y de que el proceso esté en estado seguro.

Extraiga la tarjeta de transferencia antes de cambiar la CPU a estado operativo RUN.

Configuración de dispositivos

6

Para crear la configuración de dispositivos del PLC es preciso agregar una CPU y módulos adicionales al proyecto.



① Módulo de comunicación (CM) o procesador de comunicaciones (CP): máx. 3, insertados en los slots 101, 102 y 103

② CPU: Slot 1

③ Puerto PROFINET de la CPU

④ Signal Board (SB), Communication Board (CB) o Battery Board (BB): máx. 1, insertada en la CPU

⑤ Módulo de señales (SM) para E/S digitales o analógicas: máx. 8, insertados en los slots 2 a 9
(la CPU 1214C, la CPU 1215C y la CPU 1217C permiten 8, la CPU 1212C permite 2, la CPU 1211C no permite ninguno)

Control de configuración

La configuración de dispositivos para el S7-1200 también soporta el "control de configuración (Página 142)", que permite configurar una configuración máxima para un proyecto incluyendo módulos que posiblemente no se utilicen en realidad. Esta función, conocida también como "configuración futura", permite configurar una configuración máxima que puede usarse con variaciones en los módulos instalados en múltiples aplicaciones.

6.1 Insertar una CPU

Una CPU puede insertarse en el proyecto desde la vista del portal o desde la vista del proyecto de STEP 7:

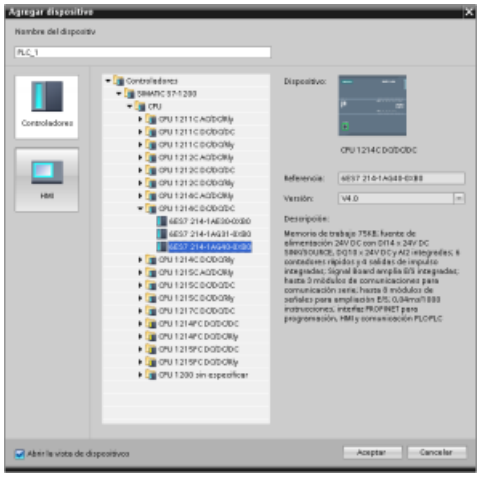
- En la vista de portal, seleccione "Dispositivos y redes" y haga clic en "Agregar dispositivo".



- En la vista del proyecto, bajo el nombre del proyecto, haga doble clic en "Agregar nuevo dispositivo".



En el cuadro de diálogo "Agregar nuevo dispositivo", seleccione el modelo y la versión de firmware correctos de la lista.

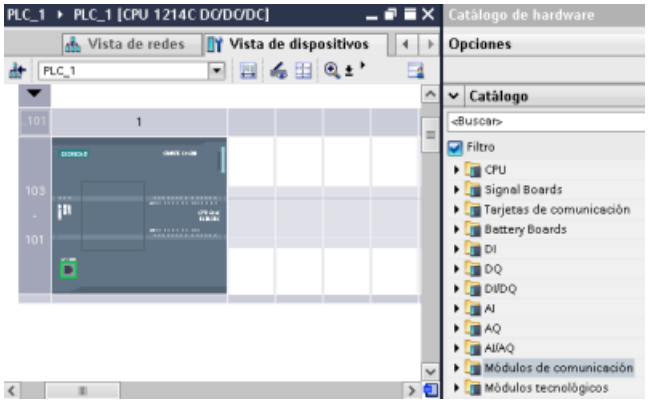


Configurar los ajustes de seguridad PLC para la CPU insertada

Al insertar una CPU S7-1200 V4.5, STEP 7 inicia el asistente de seguridad (Página 160) para ayudarle a configurar los ajustes de seguridad del PLC. Siga los pasos en el asistente para configurar los ajustes de seguridad del PLC.

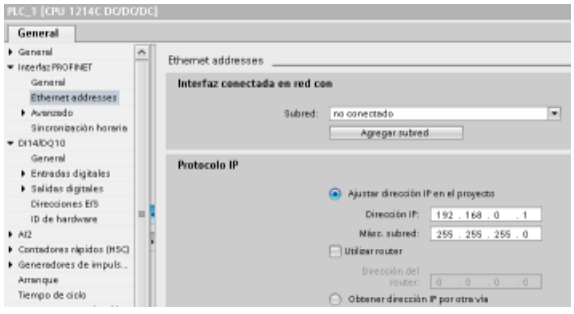
Configuración de dispositivos para la CPU insertada

Después de agregar una CPU, STEP 7 genera el rack y muestra la CPU en la vista de dispositivos:



Al hacer clic en la CPU, en la vista del dispositivo se visualizan las propiedades de la CPU en la ventana de inspección.

La dirección IP de la CPU se puede asignar manualmente durante la configuración del dispositivo. Si la CPU está conectada a un router de la red, también es preciso introducir la dirección IP del router.



6.2 Cargar la configuración de una CPU conectada

STEP 7 ofrece dos métodos para cargar la configuración hardware de una CPU conectada:

- Cargar el dispositivo conectado como una estación nueva
- Configurar una CPU no especificada y detectar la configuración hardware de la CPU conectada

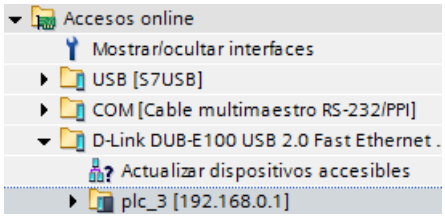
Sin embargo, tenga en cuenta que el primer método carga tanto la configuración hardware como el software de la CPU conectada.

Cargar un dispositivo como una estación nueva

Para cargar un dispositivo conectado como estación nueva, proceda del siguiente modo:

1. Amplíe su interfaz de comunicación desde el nodo "Acceso online" del árbol de proyecto.
2. Haga doble clic en "Actualizar dispositivos accesibles".

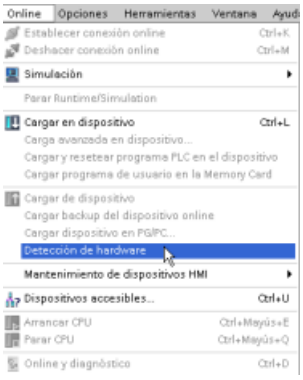
3. Seleccione el PLC de entre los dispositivos detectados.



4. Desde el menú "Online" de STEP 7 seleccione el comando "Carga del dispositivo como estación nueva (hardware y software)".

STEP 7 carga tanto la configuración hardware como los bloques de programa.

Detectar la configuración hardware de una CPU sin especificar



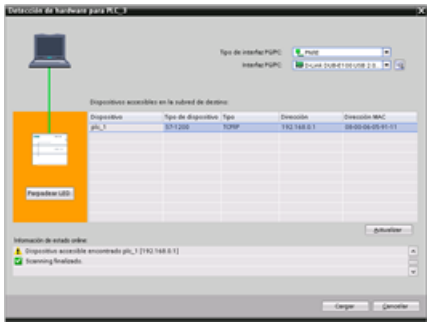
Si existe una conexión con una CPU, es posible cargar su configuración en el módulo (incluidos los módulos). Tan solo hay que crear un proyecto nuevo y seleccionar la "CPU sin especificar" en lugar de una específica. (También es posible omitir la configuración de dispositivo por completo seleccionando "Crear un programa PLC" en "Primeros pasos". Entonces, STEP 7 crea automáticamente una CPU sin especificar).

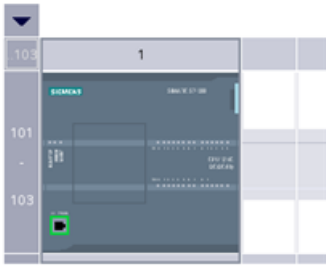
En el editor de programación, seleccione el comando "Detección de hardware" del menú "Online".

En el editor de configuración de dispositivos, seleccione la opción de detección del dispositivo conectado.



Tras seleccionar la CPU en el cuadro de diálogo online y pulsar el botón de carga, STEP 7 carga la configuración de hardware de la CPU, incluidos todos los módulos (SM, SB o CM). Entonces pueden configurarse los parámetros de la CPU y de los módulos (Página 153).





6.3 Agregar módulos a la configuración

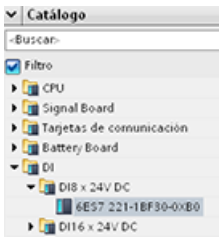
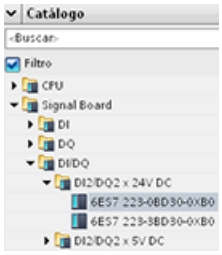
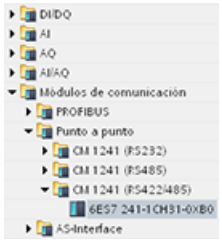
El catálogo de hardware se utiliza para agregar módulos a la CPU:

- El módulo de señales (SM) ofrece E/S digitales o analógicas adicionales. Estos módulos se conectan a la derecha de la CPU.
- La Signal Board (SB) ofrece unas pocas E/S adicionales a la CPU. La SB se inserta en el frente de la CPU.
- La nueva BB 1297 ofrece respaldo a largo plazo del reloj en tiempo real. La BB se inserta en la parte frontal de la CPU.
- La placa de comunicación (CB) ofrece un puerto de comunicación adicional (como RS485). La CB se inserta en la parte frontal de la CPU.
- El módulo de comunicación (CM) y el procesador de comunicación (CP) ofrecen un puerto de comunicación adicional, como para PROFIBUS o GPRS. Estos módulos se conectan a la izquierda de la CPU.

Para insertar un módulo en la configuración del dispositivo, selecciónelo en el catálogo de hardware y haga doble clic en él, o bien arrástrelo hasta el slot resaltado. Debe agregar los

módulos a la configuración del dispositivo y descargar la configuración de hardware en la CPU para que estén operativos.

Tabla 6-1 Adición de un módulo a la configuración del dispositivo

Módulo	Seleccionar el módulo	Insertar el módulo	Resultado
SM			
SB, BB o CB			
CM o CP			

Con la función "control de configuración" (Página 142) es posible agregar módulos de señales y Signal Boards a la configuración del dispositivo que no se corresponden con el hardware real para una aplicación en concreto, pero que serán usados en aplicaciones relacionadas que comparten un programa de usuario o modelo de CPU común y, posiblemente, algunos de los módulos configurados.

6.4 Control de configuración

6.4.1 Ventajas y aplicaciones del control de configuración

El control de configuración puede ser una solución útil cuando se crea una solución de automatización (máquina) que debe utilizarse con variaciones en varias instalaciones.

Existe la posibilidad de cargar una configuración de dispositivo y un programa de usuario STEP 7 a configuraciones de PLC con diferentes instalaciones. Solo hay que realizar algunas adaptaciones simples para que el proyecto de STEP 7 concuerde con la instalación real.

6.4.2 Configuración de la instalación central y módulos opcionales

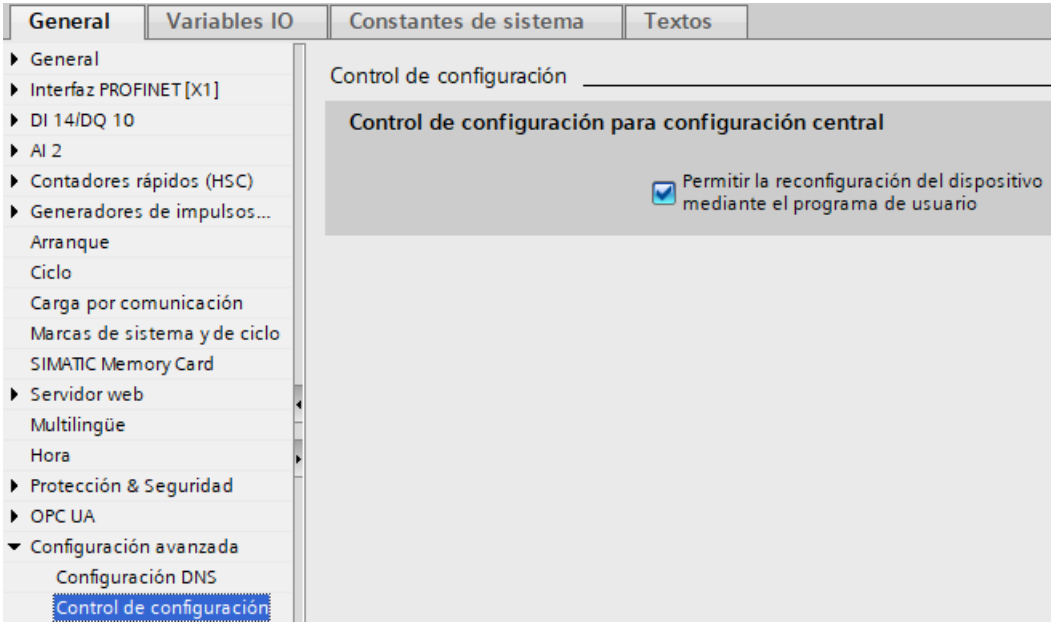
El control de configuración con STEP 7 y S7-1200 permite configurar una configuración máxima para una máquina estándar y utilizar versiones (opciones) que usen un subconjunto de esta configuración. El manual PROFINET con STEP 7 (https://support.industry.siemens.com/cs/ww/es/view/49948856) hace referencia a estos tipos de proyectos como "Proyectos de máquina estándar".

Un juego de datos de control, también denominado registro de control, que se programa en el bloque de programa de arranque notifica a la CPU cuántos módulos faltan en la instalación real en comparación con la configuración o qué módulos están ubicados en los diferentes slots en comparación con la configuración. El control de configuración no afecta la asignación de parámetros de los módulos.

El control de configuración ofrece la flexibilidad de variar la instalación mientras pueda derivar la configuración real de la configuración máxima de dispositivos en STEP 7.

Para activar el control de configuración y estructurar el juego de datos de control necesario, proceda del siguiente modo:

1. Opcionalmente, restablezca los ajustes de fábrica de la CPU para asegurarse de que en la CPU no haya ningún juego de datos de control incompatible.
2. Seleccione la CPU en la configuración de dispositivos de STEP 7.
3. Desde el nodo de control de configuración en las propiedades de la CPU, seleccione la casilla de verificación "Permitir la reconfiguración del dispositivo con el programa de usuario".

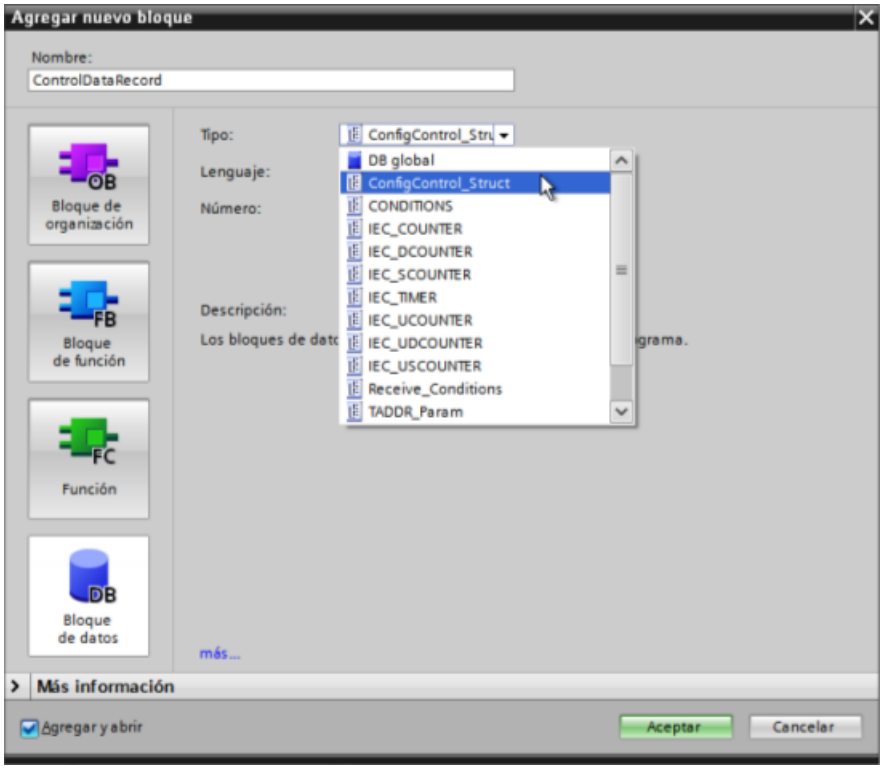


4. Cree un tipo de datos PLC que contenga el juego de datos de control. Configúrelo como una estructura que incluya cuatro USint para la información del control de configuración y otros USint que se correspondan con los slots de una configuración de dispositivos máxima en S7-1200, tal y como sigue:

ConfigControl_Struct

	Nombre	Tipo de datos	Valor predet.	Comentario
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. Cree un bloque de datos del tipo de datos PLC que ha creado.



6. En este bloque de datos configure Block_length, Block_ID, versión y subversión tal y como se muestra abajo. Configure los valores para los slots sobre la base de su presencia o ausencia y posición en la instalación real:
   - 0: el módulo configurado no está presente en la configuración real. (El slot está vacío.)
   - 1 a 9, 101 a 103: la posición de slot real para el slot configurado
   - 255: la configuración de dispositivos STEP 7 no incluye un módulo en este slot.

Nota

Control de configuración no disponible para HSC y PTO en la Signal Board

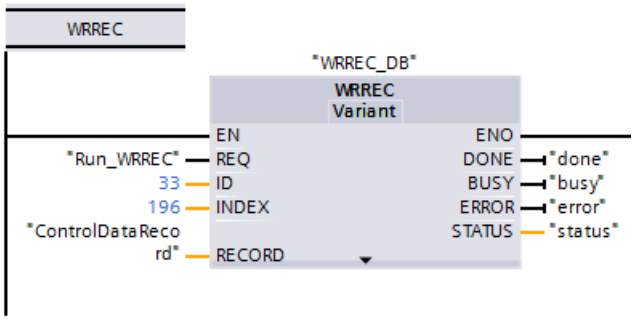
Si en la CPU dispone de una Signal Board configurada para HSC o PTO, no la desactive con un "0" en el slot_1 del juego de datos del control de configuración. Los dispositivos HSC y PTO configurados de la CPU son obligatorios en relación con el control de configuración.

ControlDataRecord

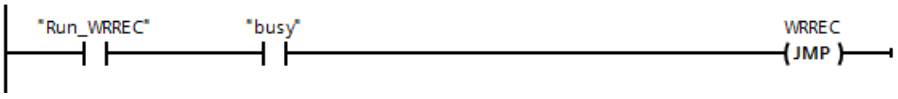
	Nombre	Tipo de datos	Valor de arranque	Comentario
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/ Actual annex ...
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

Consulte Ejemplo de control de configuración (Página 149) para obtener una explicación de cómo asignar los valores de slot.

7. En el OB de arranque debe llamar la instrucción WRREC (Escribir registro) ampliada para transferir el registro o juego de datos de control que ha creado al índice 196 del ID de hardware 33. Use una etiqueta y una instrucción JMP (saltar) para esperar a que la instrucción WRREC finalice.
   Segmento 1:



Segmento 2:



Nota

El control de configuración no actúa hasta que la instrucción WRREC transfiere el juego de datos de control en el OB de arranque. Si se ha activado el control de configuración y la CPU no dispone del juego de datos de control, pasará al estado operativo STOP cuando salga del estado operativo ARRANQUE. Asegúrese de programar el OB de arranque para transferir el juego de datos de control.

Disposición de los módulos

La tabla siguiente muestra la asignación del número de slot:

Slot	Módulos
1	Signal Board o Communication Board (tarjeta adjunta a la CPU)
De 2 a 9	Módulos de señales
De 101 a 103	Módulos de comunicación

Juego de datos de control

Un juego de datos de control 196 contiene la asignación de slots y representa la configuración real tal y como se muestra abajo:

Byte	Elemento	Valor	Explicación
0	Longitud de bloque	16	Encabezado
1	ID de bloque	196	
2	Versión	5	
3	Subversión	0	

Byte	Elemento	Valor	Explicación
4	Asignación de la tarjeta adjunta a la CPU	Tarjeta adjunta real, 0 o 255*	Elemento de control Describe en cada elemento qué slot real del dispositivo está asignado al slot configurado.
5	Asignación del slot configurado 2	Slot real, 0 o 255*	
...	...	...	
12	Asignación del slot configurado 9	Slot real, 0 o 255*	
13	Asignación del slot configurado 101	Slot real o 255*	A diferencia de los módulos de señales, el slot real para los módulos de comunicación físicamente presentes debe ser el mismo que el slot configurado.
14	Asignación del slot configurado 102	Slot real o 255*	
15	Asignación del slot configurado 103	Slot real o 255*	

***Valores de slot:**

0: el módulo configurado no está presente en la configuración real. (El slot está vacío.)

1 a 9, 101 a 103: la posición real del slot para el slot configurado

255: la configuración de dispositivos STEP 7 no incluye un módulo en este slot.

Nota

Alternativa a la creación de un tipo de variable PLC

Como alternativa a la creación de un tipo de variable PLC existe la posibilidad de crear un bloque de datos directamente con todos los elementos de estructura de un juego de datos de control. Incluso podrían configurarse varias estructuras en el bloque de datos que sirvan como diferentes configuraciones del juego de datos de control. Cualquier implementación es una forma efectiva de transferir el juego de datos de control durante el arranque.

Reglas

Tenga en cuenta las reglas siguientes:

- El control de configuración no soporta cambios de posición de módulos de comunicación. Tampoco es posible utilizar el control de configuración para desactivar los módulos de comunicación. Las posiciones de slots en los juegos de datos de control para los slots 101 a 103 deben corresponderse con la instalación real. Si no ha configurado un módulo para el slot en la configuración de dispositivos, introduzca 255 para esta posición del slot en el juego de datos de control. Si ha configurado un módulo para el slot, introduzca el slot configurado como slot real para esta posición del slot.
- Los módulos de periferia F no soportan el control de configuración. Las posiciones de slots para registros de control en un módulo de periferia F deben coincidir con la posición de slots configurada en el módulo de periferia F. Si se intenta mover o borrar un módulo de periferia F configurado por medio del juego de datos de control, todos los módulos de periferia F realmente instalados emitirán un error de parametrización y rechazarán el cambio.
- No puede haber slots vacíos (no usados) integrados entre los slots ocupados (usados). Así, por ejemplo, si la configuración real tiene un módulo en el slot 4, la configuración real también tiene que tener módulos en los slots 2 y 3. En correspondencia, si la configuración real dispone de un módulo de comunicación en el slot 102, la configuración real también tiene que tener un módulo en el slot 101.

- Si se activa el control de configuración, la CPU no está preparada para el funcionamiento sin un juego de datos de control. La CPU pasa del estado operativo ARRANQUE a STOP si un OB de arranque no transfiere un juego de datos de control válido. En este caso, la CPU no inicializa la periferia centralizada e introduce la causa del estado STOP en el búfer de diagnóstico.
- La CPU guarda un juego de datos de control transferido correctamente en la memoria remanente, lo que significa que no es necesario escribir de nuevo el juego de datos de control 196 durante un rearranque si no ha cambiado la configuración.
- Cada slot real solo debe estar presente una vez en el juego de datos de control.
- Solo es posible asignar un slot real a un slot configurado.

Nota

Modificar una configuración

La escritura de un juego de datos de control con una configuración modificada provoca la reacción automática siguiente de la CPU: borrado total seguido de un arranque con la configuración modificada.

Como resultado de esta reacción, la CPU borra el juego de datos de control original y guarda en la memoria remanente el juego de datos de control nuevo.

Comportamiento durante el funcionamiento

Para la visualización online y para la visualización en el búfer de diagnóstico (módulo en buen estado o módulo defectuoso), STEP 7 utiliza la configuración del dispositivo y no la configuración real divergente.

Ejemplo: Un modulo emite datos de diagnóstico. Este módulo está configurado en el slot 4 pero en realidad está insertado en el slot 3. La vista online indica que el slot 4 configurado es defectuoso. En la configuración real, el módulo del slot 3 señaliza un error mediante su indicador LED.

Si se han configurado módulos como ausentes en el juego de datos de control (0 entradas), el sistema de automatización se comporta de la forma siguiente:

- Los módulos designados como ausentes en el juego de datos de control no suministran diagnósticos y su estado siempre es correcto. El estatus del valor es correcto.
- El acceso de escritura directo a las salidas o el acceso de escritura a la memoria imagen de proceso de las salidas que no están presentes no tienen efecto; la CPU no informa de errores de acceso.
- El acceso de lectura directo a las entradas o el acceso de lectura a la memoria imagen de proceso de las entradas que no están presentes resultan en un valor "0" para cada entrada; la CPU no informa de errores de acceso.
- La escritura de un registro en un módulo que no está presente no tiene efecto; la CPU no informa de errores.
- El intento de leer un registro de un módulo que no está presente provoca un error porque la CPU no puede devolver un registro válido.

Mensajes de error

La CPU devuelve los mensajes de error siguientes si se produce un error durante la escritura del juego de datos de control:

Código de error	Significado
16#80B1	Longitud no válida; la información de longitud en el juego de datos de control no es correcta.
16#80B5	Parámetros de control de configuración no asignados
16#80E2	El registro ha sido transferido en un contexto de OB erróneo. El registro debe ser transferido al OB de arranque.
16#80B0	El tipo de bloque (byte 2) del juego de datos de control no es igual a 196.
16#80B8	Error de parámetros; el módulo señaliza parámetros no válidos, por ejemplo: • El juego de datos de control intenta modificar la configuración de un módulo de comunicación o una tarjeta adjunta de comunicación. La configuración real para los módulos de comunicación y una tarjeta adjunta de comunicación debe ser igual a la configuración de STEP 7. • El valor asignado para un slot no configurado en el proyecto de STEP 7 no es igual a 255. • El valor asignado para un slot configurado está fuera de rango. • La configuración asignada tiene un slot interno vacío, por ejemplo, el slot n está asignado y el slot n-1 no está asignado.

6.4.3 Ejemplo de control de configuración

Este ejemplo describe una configuración consistente en una CPU y tres módulos E/S. El módulo del slot 3 no está presente en la primera instalación real, de forma que se usa el control de configuración para "ocultarlo".

En la segunda instalación la aplicación incluye el módulo que inicialmente había sido ocultado, pero ahora lo incluye en el último slot. Un registro de control modificado ofrece información sobre las asignaciones de los slots de los módulos.

Ejemplo: instalación real con un módulo configurado pero no usado

La configuración del dispositivo contiene todos los módulos que pueden estar presentes en una instalación real (configuración máxima). En este caso, el módulo que ahora está en el slot 3 en la configuración del dispositivo no está presente en la instalación real.

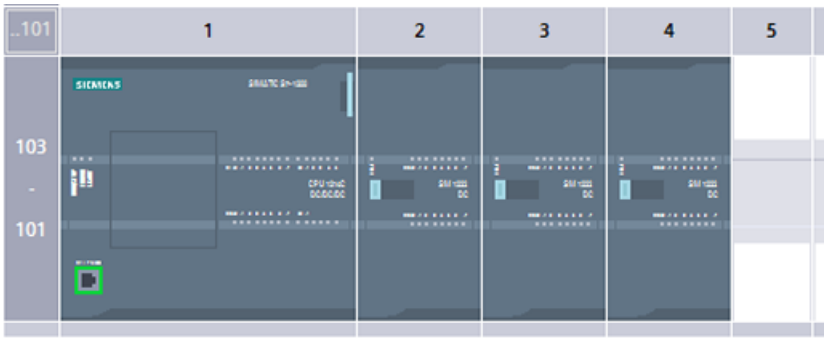


Figura 6-1 Configuración del dispositivo para una instalación máxima con tres módulos de señales



Figura 6-2 Instalación real con el módulo configurado en el slot 3 ausente y el módulo configurado para el slot 4 en el slot 3

Para indicar la ausencia del módulo que falta debe configurarse el slot 3 en el registro de control con 0.

ControlDataRecord

		Nombre	Tipo de datos	Valor de arranque	Comentario
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 ...
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

Ejemplo: instalación real con el módulo agregado posteriormente en un slot diferente

En el segundo ejemplo, el módulo del slot 3 de la configuración del dispositivo está presente en la instalación real pero está en el slot 4.

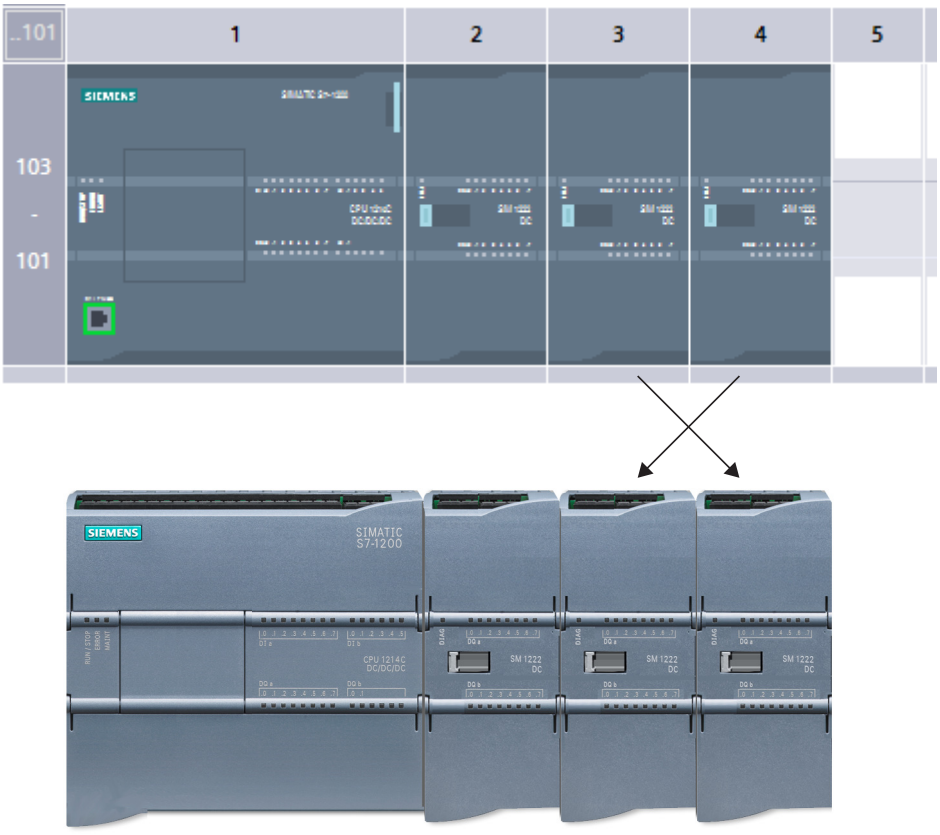


Figura 6-3 Configuración del dispositivo comparado con la instalación real con módulos intercambiados en los slots 3 y 4

Para correlacionar la configuración del dispositivo con la instalación real, edite el registro de control para asignar los módulos a las posiciones correctas de los slots.

ControlDataRecord

	Nombre	Tipo de datos	Valor de arranque	Comentario
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 ...
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 Cambiar un dispositivo

Existe la posibilidad de cambiar el tipo de dispositivo de una CPU o un módulo configurado. En la configuración de dispositivos, haga clic con el botón derecho del ratón en el dispositivo y elija el comando "Cambiar dispositivo" del menú contextual. En el cuado de diálogo que aparece, navegue hasta la CPU o el módulo que desee reemplazar y selecciónelo. El cuadro de diálogo Cambiar dispositivo muestra información sobre la compatibilidad entre ambos dispositivos.

Encontrará indicaciones sobre el cambio de dispositivos entre diferentes versiones de CPU en Sustitución de una CPU V3.0 por una CPU V4.x (Página 1448).

6.6 Configurar el funcionamiento de la CPU

6.6.1 Vista general

Para configurar los parámetros operativos de la CPU, selecciónela en la vista de dispositivos (recuadro azul alrededor de la CPU) y utilice la ficha "Propiedades" de la ventana de inspección.

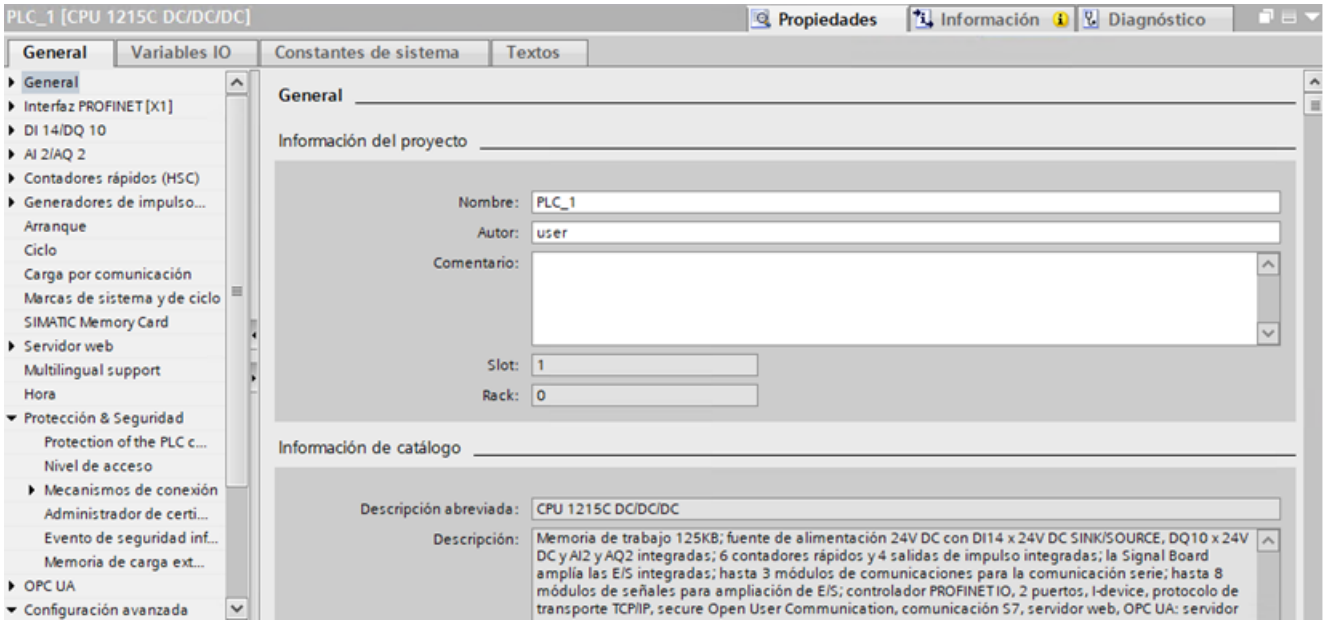


Tabla 6-2 Propiedades de la CPU

Propiedad	Descripción
Interfaz PROFINET	Permite configurar la dirección IP de la CPU y la sincronización horaria
DI, DO y AI	Permite configurar la reacción de las E/S locales (integradas) digitales y analógicas (por ejemplo, tiempos de filtro de entradas digitales y reacción de las salidas digitales ante una parada de la CPU).
Contadores rápidos (Página 539) y generadores de impulsos (Página 471)	Permite habilitar y configurar los contadores rápidos (HSC) y generadores de impulsos utilizados para las operaciones de tren de impulsos (PTO) y modulación del ancho de impulsos (PWM) Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con la PWM o con instrucciones de Motion Control), las direcciones de salida correspondientes se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa de usuario. Si el programa de usuario escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

Propiedad	Descripción
Arranque (Página 71)	**Arranque tras POWER ON:** permite configurar la reacción de la CPU a una transición de OFF a ON, p. ej. el arranque en estado operativo STOP o la transición a RUN tras un arranque en caliente
	Compatibilidad de hardware soportada: Configura la estrategia de sustitución para todos los componentes del sistema (SM, SB, CM, CP y CPU): • Permitir sustituto aceptable • Permitir cualquier sustituto (valor predeterminado) Cada módulo contiene a nivel interno requisitos de compatibilidad de sustitución según el número de E/S, la compatibilidad eléctrica y otros puntos de comparación que procedan. Por ejemplo, un SM de 16 canales podría ser un sustituto aceptable de un SM de 8 canales, pero un SM de 8 canales no sería un sustituto aceptable para un SM de 16 canales. Si selecciona "Permitir sustituto aceptable", STEP 7 aplica las reglas de sustitución; en caso contrario, STEP 7 permite cualquier sustitución.
	Tiempo de asignación de los parámetros para E/S descentralizadas: configura un tiempo máximo (valor predeterminado: 60000 ms) para las E/S descentralizadas que deben conmutarse online. (Los CMs y CPs reciben la alimentación y los parámetros de comunicación de la CPU durante el arranque. El tiempo de asignación permite que las E/S conectadas al CM o CP cambien a online). La CPU pasa a RUN en cuanto las E/S descentralizadas están online, independientemente del tiempo de asignación. Si las E/S descentralizadas no se han llevado online dentro de este tiempo, la CPU pasará igualmente a RUN sin las E/S descentralizadas. **Nota:** Si se utiliza un CM 1243-5 (maestro PROFIBUS) en la configuración, no debe ajustarse este parámetro por debajo de 15 segundos (15000 ms) con el fin de garantizar que el módulo pueda pasar a online.
	Los OB deben poder interrumpirse: Configura si la ejecución del OB (para todos los OB) en la CPU debe poder interrumpirse o no (Página 87)
Ciclo (Página 91)	Permite definir un tiempo de ciclo máximo o un tiempo de ciclo mínimo fijo
Carga de comunicación	Permite asignar el porcentaje del tiempo de la CPU que debe dedicarse a las tareas de comunicación
Marcas de sistema y de ciclo (Página 95)	Habilita un byte para funciones de "memoria de sistema" y un byte para funciones de "marca de ciclo" (en las que cada bit cambia entre on y off con una frecuencia predefinida)
Servidor web (Página 843)	Habilita y configura la funcionalidad del servidor web
Hora	Selecciona la zona horaria y configura el horario de verano/invierno
Soporte multilingüe (Página 158)	Asigna un idioma del proyecto al servidor web que se usa para mostrar textos de entrada del búfer de diagnóstico para cada uno de los idiomas de visualización posibles de la interfaz de usuario del servidor web.
Protección (Página 163)	Permite ajustar la protección de lectura/escritura y las contraseñas para acceder a la CPU.
Control de configuración (Página 142)	Permite realizar una configuración de dispositivo maestro que puede controlarse para diferentes configuraciones reales de dispositivos
Recursos de conexión (Página 591)	Ofrece un resumen de los recursos de conexión de comunicación que están disponibles para la CPU y del número de recursos de conexión que se han configurado
Vista general de las direcciones	Ofrece un resumen de las direcciones E/S que se han configurado para la CPU

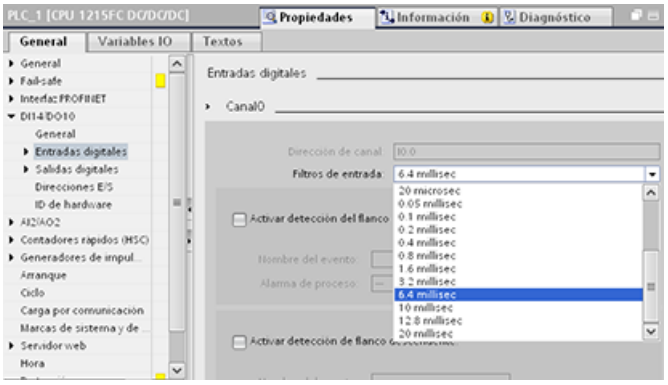
6.6.2 Configuración de tiempos de filtro de entradas digitales

Los filtros de entradas digitales protegen el programa para que no responda a cambios rápidos no deseados de las señales de entradas, ya que pueden producirse a causa de rebotes de contactos o ruido eléctrico. El tiempo de filtro predeterminado de 6,4 ms bloquea las transiciones no deseadas de los contactos mecánicos habituales. Las diferentes E/S de la aplicación pueden requerir tiempos de filtro más cortos para detectar y responder a las entradas de sensores rápidos o tiempos de filtro más largos para bloquear el rebote de contacto lento o el ruido de impulsos más largo.

Un tiempo de filtro de entrada de 6,4 ms implica que un solo cambio de señal, de "0" a "1" o de "1" a "0", debe prolongarse durante aproximadamente 6,4 ms para detectarlo, y que no se detecte un solo impulso alto o bajo inferior a 6,4 ms. Si una señal de entrada conmuta entre "0" y "1" más rápidamente que el tiempo de filtro, el valor de entrada puede modificarse en el programa de usuario cuando la duración acumulada de los nuevos impulsos de valor frente a los impulsos de valor anteriores supere el tiempo de filtro.

El filtro de entradas digitales funciona del siguiente modo:

- Cuando la entrada es "1", el contador se incrementa y se detiene en el tiempo de filtro. El bit de imagen del proceso cambia de "0" a "1" cuando el contaje alcanza el tiempo de filtro.
- Cuando se introduce "0", el contador baja y se detiene en el valor "0". El bit de imagen del proceso cambia de "1" a "0" cuando el contaje alcanza el valor "0".
- Si la entrada cambia en ambos sentidos, el contador aumentará a veces y disminuirá a veces. La memoria imagen de proceso cambiará cuando la acumulación neta de los contajes alcance el tiempo de filtro o "0".
- Una señal que cambie rápidamente con más "0" que "1" finalmente pasará al valor "0" y, si hay más "1" que "0", la memoria imagen de proceso finalmente cambiará a "1".



Cada entrada tiene una sola configuración de filtro, aplicable a todos los usos: entradas de proceso, interrupciones, captura de impulsos y entradas de HSC. Para configurar los tiempos de filtro de entrada, seleccione "Entradas digitales".

El tiempo de filtro predeterminado para las entradas digitales es de 6,4 ms. El tiempo de filtro se selecciona en la lista desplegable para filtros de entrada. Los tiempos de filtro válidos van de 0,1 us a 20,0 ms.

⚠ ADVERTENCIA

Riesgos asociados con los cambios en el tiempo de filtro para el canal de entrada digital

Si se reajusta el tiempo de filtro de un canal de entrada digital, es posible que el nuevo valor de entrada de nivel "0" deba permanecer a "0" durante 20 ms para que el filtro esté totalmente operativo ante nuevas entradas. Durante este tiempo, puede que no se detecten o no se cuenten los eventos de pulsación breve "0" cuya duración sea inferior a 20 ms.

Este cambio de los tiempos de filtro puede originar un funcionamiento inesperado de los equipos o del proceso, lo que puede causar la muerte o lesiones graves al personal y/o daños a los equipos.

Para asegurar que un tiempo de filtro nuevo tenga efecto inmediato, desconecte y vuelva a conectar la CPU.

Configurar tiempos de filtro para las entradas digitales utilizadas como HSC

Modifique el tiempo de filtro de las entradas que utilice como contadores rápidos (HSC) a un valor apropiado para no perder recuentos.

Siemens recomienda los ajustes siguientes:

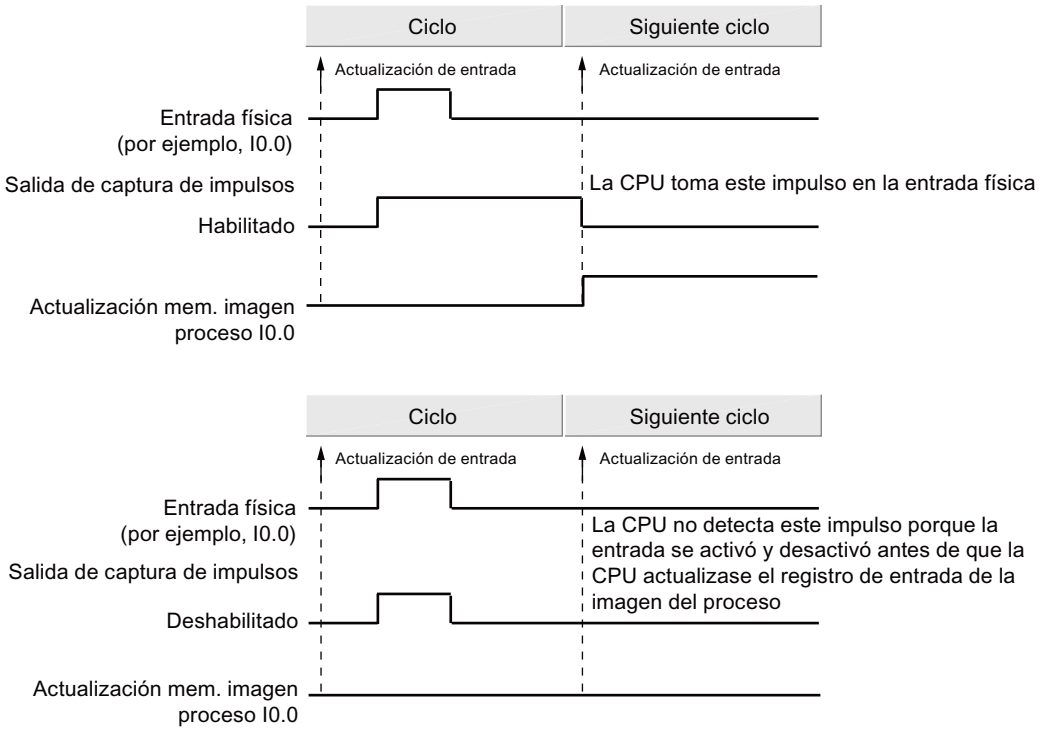
Tipo de HSC	Tiempo de filtro recomendado
1 MHz	0,1 microsegundos
100 kHz	0,8 microsegundos
30 kHz	3,2 microsegundos

6.6.3 Capturar impulsos

La CPU S7-1200 ofrece una función de toma o captura de impulsos para puntos de entradas digitales. Esta función permite detectar impulsos altos o bajos de tan corta duración que no se registrarían en todos los casos, cuando la CPU lee las entradas digitales al comienzo del ciclo.

Si habilita la captura de impulsos para una entrada, se iniciará un cambio en el estado de la entrada y se mantendrá hasta la siguiente actualización de ciclo de la entrada. De ese modo se garantiza que un impulso que dura un corto periodo de tiempo pueda detectarse y retenerse hasta que la CPU lea las entradas.

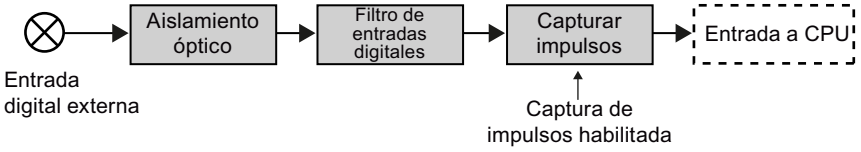
Las siguientes figuras muestran el funcionamiento básico de la CPU S7-1200 con y sin habilitar la captura de impulsos:



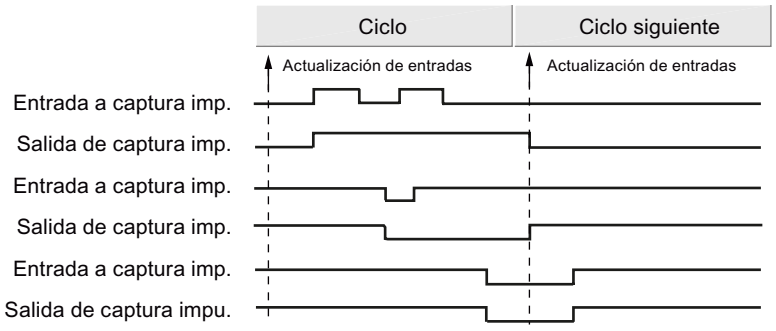
Nota

Dado que la función de toma de impulsos se aplica a la entrada después de pasar a través del filtro de entrada, hay que ajustar el tiempo de filtro de entrada para que el filtro no elimine el impulso.

La figura siguiente muestra un diagrama de bloques de una entrada digital:



La figura siguiente muestra la reacción de una función de captura de impulsos a diversas condiciones de entrada. Si en un ciclo dado se presenta más de un impulso, se lee solo el primer impulso. Si se presentan varios impulsos en un ciclo, se recomienda utilizar los eventos de interrupción de flanco ascendente/descendente:



6.7 Configurar el soporte multilingüe

La configuración del soporte multilingüe permite asignar uno de los dos idiomas del proyecto a cada idioma de la interfaz de usuario para el servidor web (Página 843) del S7-1200. También existe la posibilidad de no configurar un idioma del proyecto para un idioma de la interfaz de usuario.

¿Qué es un idioma del proyecto?

El idioma del proyecto es el idioma empleado por el TIA Portal para mostrar textos de proyecto definidos por el usuario como comentarios de red y bloque.

Los idiomas del proyecto se seleccionan en el TIA Portal con el comando de menú **Herramientas > Idiomas del proyecto** para el proyecto seleccionado en el árbol del proyecto.

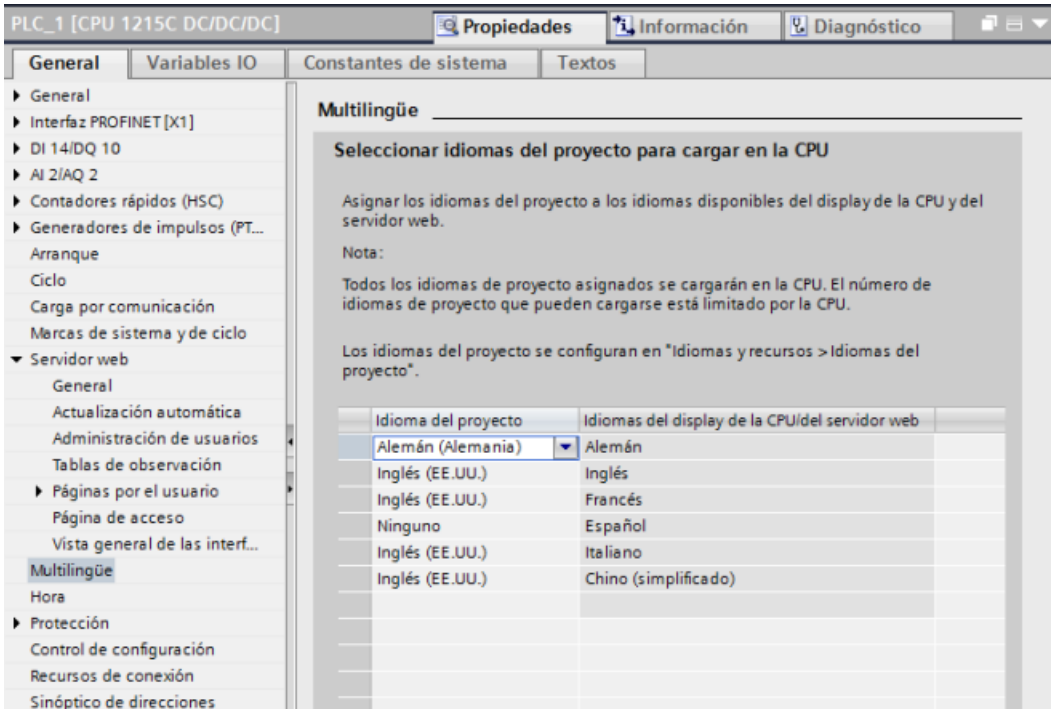
Seguidamente, es posible configurar textos de usuario como comentarios de red y bloque en cada idioma del proyecto con el comando de menú **Herramientas > Textos del proyecto**. Cuando se cambia el idioma de la interfaz de usuario del TIA Portal, los comentarios de red, comentarios de bloque y otros textos del proyecto multilingüe se muestran en el idioma del proyecto correspondiente. El idioma de la interfaz de usuario del TIA Portal se ajusta desde el comando de menú **Opciones > Ajustes** del idioma del proyecto.

También existe la posibilidad de configurar los idiomas y los textos del proyecto en el nodo del árbol del proyecto **Idiomas y recursos**.

El servidor web puede emplear uno o dos de los idiomas del proyecto de STEP 7 para mostrar los avisos del búfer de diagnóstico.

Correspondencia entre el idioma del proyecto y el idioma de la interfaz de usuario en el servidor web

El servidor web soporta los mismos idiomas de interfaz de usuario que el TIA Portal; sin embargo, solo soporta dos idiomas del proyecto como máximo. Es posible configurar el servidor web para que utilice uno de los dos idiomas del proyecto para entradas de texto del búfer de diagnóstico en función del idioma de la interfaz de usuario del servidor web. Estos ajustes se configuran en las propiedades "Soporte multilingüe" de la configuración de dispositivos de la CPU. (Los comentarios de red y de bloque así como otros textos multilingües no son visibles desde el servidor web.)



En las propiedades Soporte multilingüe, los idiomas de la interfaz de usuario de la derecha no pueden editarse. Son idiomas predefinidos que están disponibles tanto para el TIA Portal como para las interfaces de usuario del servidor web. El ajuste "Asignar idioma del proyecto" es configurable y puede ser uno de los dos idiomas de proyecto configurados, aunque también puede ser "Ninguno". Puesto que la CPU S7-1200 solo soporta dos idiomas del proyecto, no es posible configurar que el idioma del proyecto sea el mismo que el idioma de la interfaz de usuario para todos los idiomas de interfaz soportados.

En la configuración inferior, el servidor web muestra entradas del búfer de diagnóstico (Página 866) en alemán si la interfaz de usuario del servidor web está en alemán, no muestra texto para eventos del búfer de diagnóstico si la interfaz de usuario del servidor web está en español y muestra entradas del búfer de diagnóstico en inglés para el resto de idiomas.

6.8 Protección y seguridad

6.8.1 Uso del asistente de seguridad para establecer los ajustes de seguridad del PLC

El asistente de seguridad de TIA Portal V17 o superior le ofrece un lugar central para configurar sus ajustes de seguridad PLC. Cuando se inserta una CPU S7-1200 V4.5 o superior (Página 138) en el proyecto, TIA Portal inicia el asistente de seguridad.

El asistente de seguridad consta de cuatro partes:

- Protección de datos confidenciales del PLC
- Modo para comunicación PG/PC y HMI
- Protección de acceso al PLC
- Vista general

Al hacer clic en el botón "Finalizar" del asistente, STEP 7 guarda los ajustes realizados en el asistente para el proyecto. Si hace clic en "Cancelar", STEP 7 no guarda los cambios. Los cambios realizados en el asistente solo afectan al proyecto STEP 7.

Protección de datos confidenciales del PLC

La función "Protección de datos de configuración confidenciales del PLC" permite proteger cada CPU de forma individual en el proyecto. Utilice el asistente de seguridad para activar esta protección y establecer una contraseña para proteger los datos de configuración confidenciales del PLC.

- Si el dispositivo dispone de esta contraseña, TIA Portal le solicitará en la primera descarga que introduzca la contraseña de protección de datos de configuración confidenciales del PLC.
- Si el dispositivo ya dispone de esta contraseña de protección, la contraseña deben coincidir en el proyecto de STEP 7 y en el dispositivo. Si la contraseña no coincide, no podrá descargar el proyecto a la CPU. Deberá borrar la contraseña para la protección de datos de configuración confidenciales del PLC o establecerla de modo que coincida con la contraseña en el dispositivo.

La protección de datos de configuración confidenciales del PLC también se puede configurar desde la configuración de dispositivo de la CPU (Página 161).

Modo para comunicación PG/PC y HMI

El modo de comunicación PG/PC y HMI permite utilizar un certificado de comunicación del PLC (Página 586) para proteger la comunicación entre la CPU y otros dispositivos:

- Programadoras (PG) como TIA Portal y SIMATIC Automation Tool
- HMI

En el asistente, seleccione "Permitir solo la comunicación PG/PC y HMI segura" para permitir solo la comunicación segura.

Si necesita comunicarse con dispositivos que no soportan la comunicación segura, deseleccione "Permitir solo la comunicación PG/PC y HMI segura". Esta selección permite a su PLC comunicarse utilizando o bien comunicación segura o comunicación legacy (Página 586).

También puede configurar el modo de comunicación PG/PC y HMI (Página 166) en los mecanismos de comunicación de la configuración de dispositivo de la CPU.

Protección de acceso al PLC

El asistente de seguridad también permite establecer las contraseñas de nivel de acceso (Página 163) para la CPU. Se trata de la misma configuración de nivel de acceso que se encuentra en la configuración del dispositivo. El asistente de seguridad ofrece acceso según su conveniencia. TIA Portal V17 y las CPU S7-1200 V4.5 han mejorado el cifrado de estas contraseñas.

Vista general

La selección general del asistente de seguridad muestra los ajustes realizados en las áreas siguientes:

- Protección de datos confidenciales del PLC (Página 161)
- Modo para comunicación PG/PC y HMI (Página 166)
- Protección de acceso al PLC (Página 163)

Revise los ajustes realizados y use el botón Atrás si quiere hacer cambios. Cuando esté satisfecho con los ajustes, haga clic en "Finalizar". STEP 7 guarda los ajustes en el proyecto.

Iniciar el asistente de seguridad desde la configuración de dispositivos de la CPU

También puede iniciar el asistente de seguridad para una CPU V4.5 o superior desde la configuración de dispositivos de la CPU en el área Protección & seguridad.

6.8.2 Protección de datos de configuración confidenciales del PLC

La función "Protección de datos de configuración confidenciales del PLC" permite proteger la configuración de cada CPU del proyecto. Desde la sección Protección & seguridad de la configuración de dispositivos, es posible activar esta protección y establecer una contraseña para la protección de datos de configuración confidenciales del PLC.

Si configura la protección de datos de configuración confidenciales del PLC, observe lo siguiente:

- Si el dispositivo dispone de esta contraseña, TIA Portal le solicitará en la primera descarga que introduzca la contraseña de protección de datos de configuración confidenciales del PLC.
- Si el dispositivo ya dispone de esta contraseña de protección, la contraseña deben coincidir en el proyecto de STEP 7 y en el dispositivo. Si la contraseña no coincide, no podrá descargar el proyecto a la CPU. Deberá borrar la contraseña para la protección de datos de configuración confidenciales del PLC o establecerla de modo que coincida con la contraseña en el dispositivo. Se puede activar o borrar la contraseña en el dispositivo desde Online y diagnóstico (Página 1205).

Asistente de seguridad

También puede utilizar el asistente de seguridad (Página 160) para activar esta función y establecer la contraseña. El asistente de seguridad se inicia cuando se inserta por primera vez una CPU V4.5 o superior. También puede iniciar el asistente de seguridad desde el área Protección & Seguridad de la configuración de dispositivos.

Ventajas de proteger los datos de configuración confidenciales del PLC

Las CPU V4.5 y superiores con TIA Portal V17 ofrecen protección individual para cada CPU. La protección de datos de configuración confidenciales del PLC contribuye a aumentar la seguridad de almacenamiento del proyecto en cada PLC.

Funcionamiento de la protección

La protección funciona de forma similar a una cerradura y una llave. El usuario activa la protección de datos de configuración confidenciales del PLC en TIA Portal y establece una contraseña para la protección. La descarga del proyecto establece la contraseña "Protección de datos de configuración confidenciales del PLC" en la CPU. La CPU debe disponer de la contraseña de una descarga de proyecto o de una SIMATIC Memory Card (Página 130) para poder leer los archivos de proyecto. Los archivos de proyecto contienen los datos de configuración confidenciales.



Cifrado:

El usuario activa la "protección de datos de configuración confidenciales del PLC" en TIA Portal y establece una contraseña. Esta contraseña es el medio para proteger los datos de configuración confidenciales del PLC.

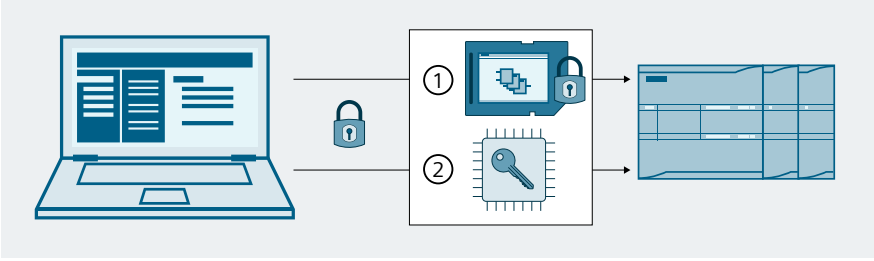


Descifrado:

El usuario descarga el proyecto a la CPU o alternativamente carga el proyecto desde una SIMATIC Memory Card (Página 130).

A continuación, la CPU puede leer (desbloquear) los archivos de proyecto.

Por ello, el cifrado de la contraseña en el proyecto y el descifrado subsiguiente en la CPU ofrecen un alto grado de protección para los datos de configuración confidenciales del PLC.



① Proyecto con datos confidenciales protegidos por contraseña en la SIMATIC Memory Card de la CPU.

② Información clave, generada a partir de la contraseña, en el área de memoria de la CPU que permite el acceso a los datos de configuración confidenciales protegidos.

Nota

Responsabilidad de las CPU

Si ha configurado y descargado en una CPU la protección para los datos de configuración confidenciales del PLC, elimine de forma segura la CPU cuando la ponga fuera de servicio.

Si ha configurado y descargado la protección para los datos de configuración confidenciales del PLC en una CPU y más adelante descarga un proyecto en la CPU en el que la versión de firmware de la CPU en el proyecto STEP 7 sea anterior a V4.5, la CPU aún contendrá la protección cifrada. Si pone fuera de servicio esta CPU, elimínela de forma segura.

Eliminar de forma segura las CPU puestas fuera de servicio impide que terceros tengan acceso a los datos de configuración confidenciales protegidos.

Herramientas online

Cuando la CPU está online, también puede establecer, borrar o modificar la contraseña para la "protección de datos de configuración confidenciales del PLC" desde las herramientas online y de diagnóstico (Página 1205).

Información adicional

Para más información sobre la funcionalidad y la implementación, consulte el capítulo "Comunicación segura" en el sistema de información de TIA Portal.

Consulte también

Sustituir una CPU con protección de datos de configuración confidenciales del PLC (Página 1447)

6.8.3 Protección por nivel de acceso para la CPU

La CPU ofrece diferentes niveles de seguridad para restringir el acceso a determinadas funciones. Al configurar el nivel de protección y la contraseña de una CPU, se limitan las funciones y áreas de memoria accesibles sin introducir una contraseña.

Todo nivel permite acceder a ciertas funciones sin introducir una contraseña. El ajuste predeterminado para la CPU es "Sin acceso (protección competa)". Para utilizar cualquier nivel de protección de acceso debe proporcionar una contraseña para ese nivel.

Si la contraseña se introduce a través de una red, esto no afecta la protección por contraseña de la CPU. La protección por contraseña no es aplicable a la ejecución de las instrucciones del programa de usuario incluyendo las funciones de comunicación. Si se introduce la contraseña correcta se puede acceder a todas las funciones del nivel.

El nivel de protección de la CPU no restringe la comunicación entre PLCs (mediante instrucciones de comunicación en los bloques lógicos).

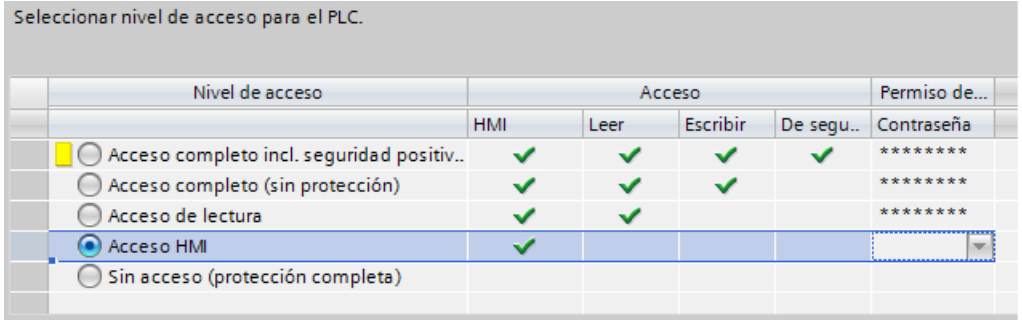
Tabla 6-3 Niveles de protección de la CPU

Nivel de protección	**Restricciones de acceso**
Acceso completo incl. seguridad positiva (sin protección)	Permite el acceso completo a una CPU F sin protección por contraseña.
Acceso completo (sin protección)	Permite el acceso completo a una CPU estándar sin protección por contraseña.
Acceso de lectura	Permite el acceso de solo lectura a la configuración hardware y a los bloques sin introducir una contraseña. Es posible cargar la configuración hardware y los bloques en la programadora. Además se dispone de acceso a HMI y a los datos de diagnóstico. Es posible visualizar resultados de comparaciones offline/online, cambiar el modo de operación (RUN/STOP) y ajustar la hora. No se pueden descargar bloques ni una configuración hardware a la CPU. Las actualizaciones de firmware no son posibles.
Acceso HMI	Solo permite el acceso HMI
Sin acceso (protección competa)	No permite el acceso sin contraseña. Solo es posible ver los datos de identificación, por ejemplo, "Dispositivos accesibles".

Tenga en cuenta que puede ajustar una dirección IP (temporal) de emergencia (Página 808) para la CPU en cualquier nivel de seguridad.

Las contraseñas distinguen entre mayúsculas y minúsculas. Para configurar el nivel de protección y las contraseñas, proceda del siguiente modo:

1. Seleccione la CPU en la "Configuración de dispositivos".
2. Seleccione la pestaña "Propiedades" en la ventana de inspección.
3. Elija la propiedad "Protección & seguridad" para seleccionar el nivel de acceso e introducir contraseñas.



La versión V4.5 de la CPU S7-1200 soporta el almacenamiento mejorado de contraseñas de nivel de acceso. Al cambiar una CPU S7-1200 a V4.5, el botón "Actualizar cifrado de contraseñas" actualiza el formato de almacenamiento de contraseñas de nivel de acceso ya existentes.

Cuando se descarga esta configuración a la CPU, el usuario dispone de acceso HMI y puede acceder a funciones HMI sin una contraseña. Para leer datos o comparar los bloques lógicos

offline/online, el usuario debe introducir la contraseña configurada para "Acceso de lectura" o la contraseña para "Acceso completo (sin protección)". Para escribir datos, el usuario debe introducir la contraseña configurada para "Acceso completo (sin protección)". Para tener pleno acceso a las CPU de seguridad, el usuario tiene que introducir la contraseña configurada para "Acceso completo incl. seguridad positiva (sin protección)".

⚠ ADVERTENCIA

Acceso no autorizado a una CPU protegida

Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU protegidos por contraseña e IDs de usuario de servidor web (Página 848) con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.

6.8.4 Configurar mecanismos de conexión

6.8.4.1 Configurar el mecanismo de acceso para interlocutores remotos

Para acceder a interlocutores remotos con instrucciones PUT/GET, el usuario también debe disponer de permisos.

Por defecto, la opción "Permitir acceso vía comunicación PUT/GET" está desactivada. En ese caso, el acceso de lectura y escritura a los datos de la CPU solo es posible para conexiones de

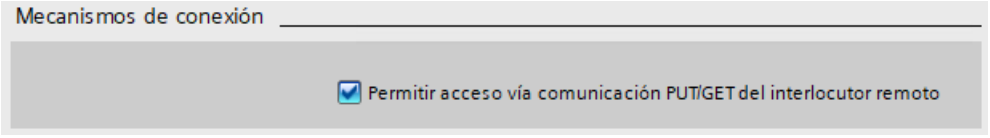
comunicación que requieren configuración o programación tanto para la CPU local como para el interlocutor. El acceso mediante instrucciones BSEND/BRCV es posible, por ejemplo.

En consecuencia, las conexiones en las que la CPU local tan solo hace de servidor (es decir, en la CPU local no se ha configurado ni programado la comunicación con el interlocutor) no son posibles mientras la CPU está en funcionamiento, por ejemplo:

- Acceso PUT/GET, FETCH/WRITE o FTP mediante módulos de comunicaciones
- Acceso PUT/GET desde otras CPU S7
- Acceso HMI mediante comunicación PUT/GET

Si desea permitir el acceso a datos de la CPU desde el lado del cliente, es decir, que no desea restringir los servicios de comunicación de la CPU, siga los siguientes pasos:

1. Configure un nivel de acceso protegido cualquiera excepto "Sin acceso (protección competa)".
2. Active la casilla de verificación "Permitir acceso vía comunicación PUT/GET".



Al descargar la configuración en la CPU, esta permitirá la comunicación PUT/GET desde interlocutores remotos

6.8.4.2 Activar la comunicación PG/PC y HMI segura y crear certificados

Utilice los "Mecanismos de conexión" en la configuración de dispositivo de la CPU para configurar si la CPU acepta solo la comunicación segura, o acepta tanto la comunicación segura como la antigua (legacy). La comunicación segura utiliza certificados X.509 a través de TLS (Transport Layer Security) 1.3. La CPU utiliza estos certificados para establecer una comunicación segura entre la CPU y los clientes. Los clientes incluyen:

- TIA Portal
- SIMATIC Automation Tool
- HMI

En el asistente, seleccione "Permitir solo la comunicación PG/PC y HMI segura" para desactivar la comunicación PG/PC y HMI antigua (legacy).

También puede crear sus propios certificados. Haga clic en "..." junto a "Certificado de comunicación del PLC" para agregar un certificado nuevo para la CPU o para seleccionar un certificado existente. Consulte el tema "Crear/renovar certificados" en el sistema de información de TIA Portal para más detalles sobre los parámetros de configuración de certificados.

Las secciones de configuración de "Protección y seguridad" en la configuración de dispositivos ofrecen ayuda en pantalla sobre sus opciones de seguridad. Estas secciones también incluyen enlaces a temas del sistema de información de TIA Portal para cada tarea de configuración y los conceptos de seguridad relacionados.

Comunicación legacy

Si necesita comunicarse con un dispositivo que no soporta la comunicación segura, deseleccione "Permitir solo la comunicación PG/PC y HMI segura". Esta selección permite a su PLC comunicarse utilizando o bien comunicación segura o comunicación legacy (Página 586).

TIA Portal V17 y posteriores ajusta por defecto el nivel más alto de comunicación segura, pero por motivos de puesta en marcha se puede forzar al TIA Portal a que use comunicación PG/PC legacy seleccionando "Utilizar solo comunicación PG/PC legacy" en el menú online.

Asistente de seguridad

También puede utilizar el asistente de seguridad (Página 160) para configurar la comunicación PG/PC y HMI para CPU V4.5 y superiores.

6.8.5 Memoria de carga externa

También es posible impedir copias de la memoria de carga interna en la memoria de carga externa (SIMATIC Memory Card) Para impedir copias de la memoria de carga interna en la memoria de carga externa, proceda del siguiente modo:

1. Desde la configuración de dispositivos de la CPU en STEP 7 seleccione "Protección" en las Propiedades generales.
2. En la sección "Memoria de carga externa" seleccione "Desactivar la copia de la memoria de carga interna en la memoria de carga externa".

Consulte también el apartado Insertar una Memory Card en la CPU (Página 120) para obtener una descripción de cómo esta propiedad afecta la inserción de una Memory Card en la CPU.

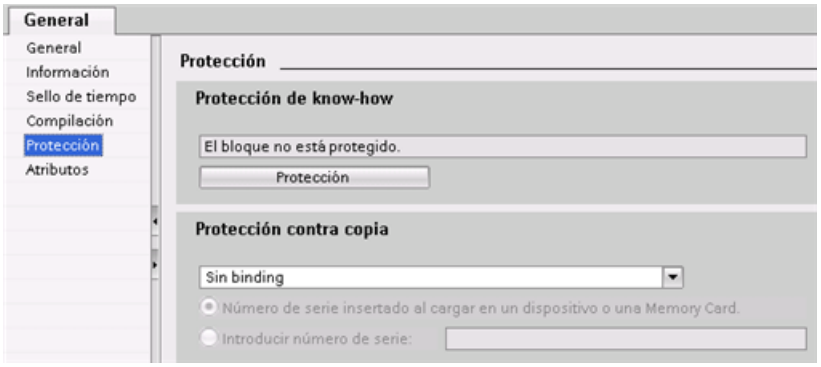
6.8.6 Protección de know-how

La protección de know-how impide el acceso no autorizado a uno o más bloques lógicos (OB, FB, FC o DB) del programa. Es posible crear una contraseña para limitar el acceso al bloque lógico. La protección por contraseña impide que el bloque lógico sea leído o modificado sin autorización. Si no se introduce la contraseña, solo es posible leer la siguiente información del bloque lógico:

- Título, comentario y propiedades del bloque
- Parámetros de transferencia (IN, OUT, IN_OUT, Return)
- Estructura de llamadas del programa
- Variables globales en las referencias cruzadas (sin información acerca de la ubicación); las variables locales se ocultan

Si un bloque se configura para que tenga protección de "know-how", no será posible acceder al código del bloque si no se introduce la contraseña.

Utilice la Task Card "Propiedades" del bloque lógico para configurar la protección de know-how de dicho bloque. Después de abrir el bloque lógico, seleccione "Protección" en "Propiedades".



1. En las propiedades del bloque lógico, haga clic en el botón "Protección" para que aparezca el cuadro de diálogo "Protección de know-how".
2. Haga clic en el botón "Definir" para introducir la contraseña.



Después de introducir y confirmar la contraseña, haga clic en "Aceptar".



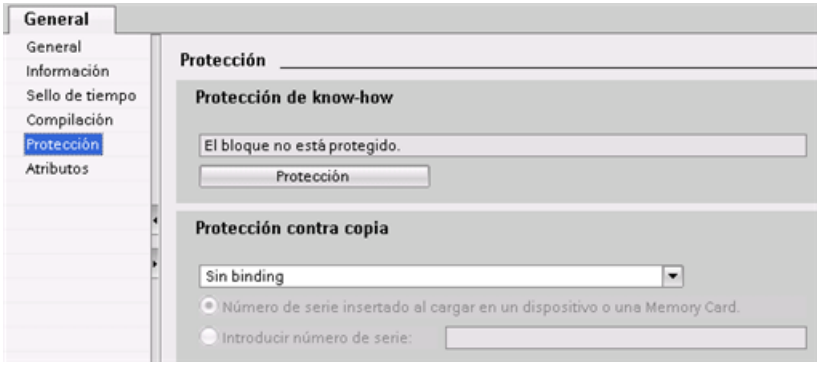
6.8.7 Protección anticopia

Una función de protección adicional permite enlazar bloques de programa para el uso con una Memory Card o CPU específica. Esta función se usa especialmente para proteger la propiedad intelectual. Al enlazar un bloque de programa a un dispositivo específico, solo se permite usar dicho programa o bloque lógico con una CPU o Memory Card determinada. Esta función permite distribuir un programa o bloque lógico electrónicamente (por Internet o mediante un correo electrónico) o bien enviando una tarjeta de memoria. La protección contra copia está disponible para OBs (Página 178), FBs (Página 181) y FCs (Página 180). La CPU S7-1200 soporta tres tipos de protección de bloques:

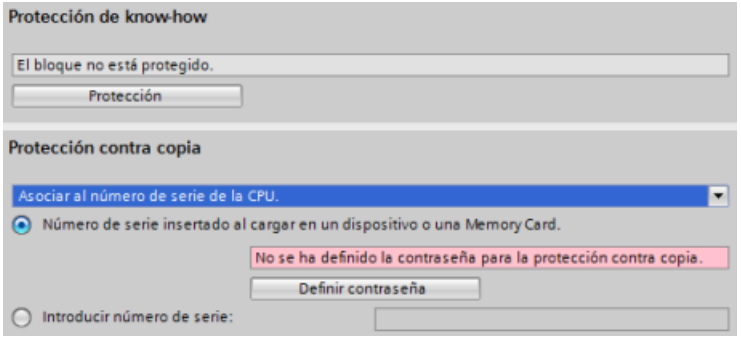
- Enlace al número de serie de una CPU
- Enlace al número de serie de una Memory Card
- Enlace dinámico con una contraseña obligatoria

Utilice la Task Card "Propiedades" del bloque lógico para enlazar el bloque con una CPU o Memory Card determinada.

1. Después de abrir el bloque lógico, seleccione "Protección".



2. En la opción "Protección contra copia" de la lista desplegable, seleccione el tipo de protección contra copia que desea usar.



3. Para enlazar al número de serie de una CPU o Memory Card, elija la opción de insertar el número de serie cuando realice la descarga o introduzca el número de serie para la Memory Card o CPU.

Nota

El número de serie distingue entre mayúsculas y minúsculas.

Para un enlace dinámico con contraseña obligatoria, defina la contraseña que debe usar para descargar o copiar el bloque.
Si, posteriormente, descarga (Página 199) un bloque con un enlace dinámico, debe introducir la contraseña para poder descargar el bloque. Tenga en cuenta que la contraseña de protección contra copia y la contraseña de protección de know-how (Página 167) son dos contraseñas diferentes.

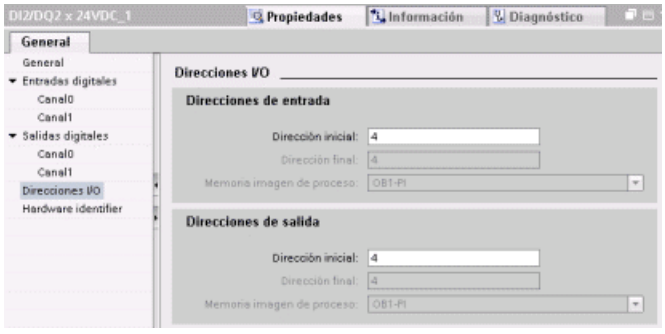
6.9 Configurar los parámetros de los módulos

Para configurar los parámetros operativos de un módulo, selecciónelo en la vista de dispositivos y utilice la ficha "Propiedades" de la ventana de inspección.

Configurar un módulo de señales (SM) o una Signal Board (SB)

La configuración de dispositivos de los módulos de señales y las Signal Boards ofrecen la posibilidad de configurar lo siguiente:

- E/S digitales: Puede configurar las entradas para detectar flancos ascendentes o descendentes (asociando cada una de ellas a un evento y una alarma de proceso) o para la "captura de impulsos" (para que permanezcan activadas tras un impulso momentáneo) hasta la siguiente actualización de la memoria imagen de proceso de las entradas. Las salidas pueden congelarse o utilizar un valor sustitutivo.
- E/S analógicas: Es posible configurar los parámetros de las distintas entradas, tales como el tipo de medición (tensión o intensidad), el rango y el alisamiento, así como habilitar el diagnóstico de rebase por defecto o por exceso. Las salidas analógicas ponen a disposición parámetros tales como el tipo de salida (tensión o intensidad) y de diagnóstico, p. ej. cortocircuito (para salidas de tensión) o diagnóstico de límite superior/inferior. En el diálogo Propiedades no se configuran rangos de entradas y salidas analógicas en unidades de ingeniería. Estas deben procesarse en la lógica del programa de la forma descrita en "Procesamiento de valores analógicos (Página 105)".
- Direcciones de E/S: El usuario configura la dirección inicial del conjunto de entradas y salidas del módulo. También puede asignar las entradas y salidas a una memoria imagen parcial del proceso (IPP0, IPP1, IPP2, IPP3, IPP4) o bien puede actualizar automáticamente o no usar una memoria imagen parcial de proceso. Consulte "Ejecución del programa de usuario" (Página 67) para encontrar una explicación sobre la memoria imagen de proceso o las memorias imagen parciales de proceso.



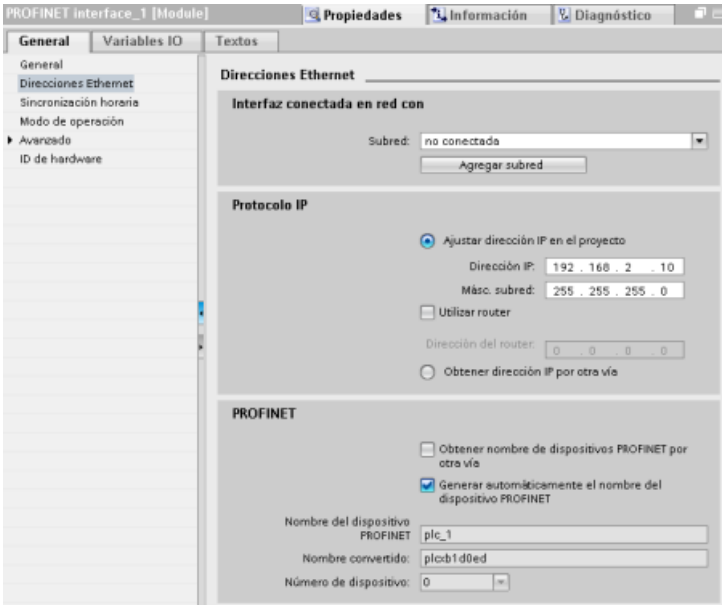
Configurar una interfaz de comunicación (CM, CP o CB)

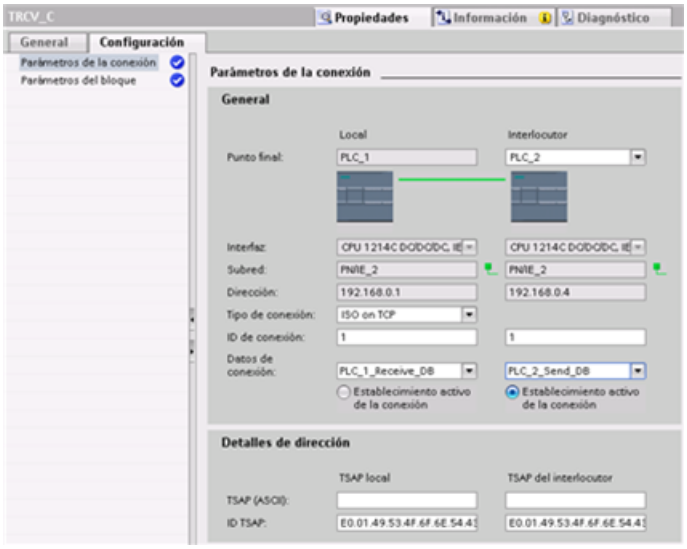
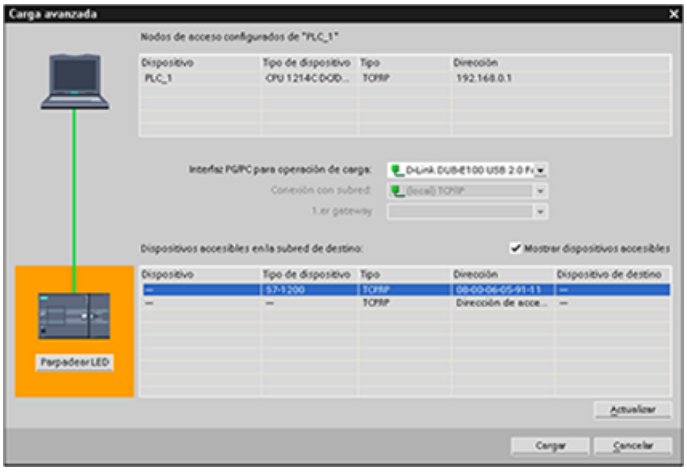
Dependiendo del tipo de interfaz de comunicación, se configuran los parámetros del segmento.



6.10 Configurar la CPU para la comunicación

El S71200 se ha diseñado para solucionar las tareas de comunicación en redes, soportando redes tanto sencillas como complejas. El S7-1200 incorpora herramientas que facilitan la comunicación con otros aparatos (p. ej. impresoras y balanzas) que utilizan sus propios protocolos de comunicación.

	Utilice la "Vista de red" de la "Configuración de dispositivos" para crear las conexiones de red entre los dispositivos del proyecto. Tras crear la conexión de red, utilice la ficha "Propiedades" de la ventana de inspección para configurar los parámetros de la red. Encontrará más información en "Crear una conexión de red" (Página 596).
	En la ventana de propiedades, seleccione la entrada de configuración "Direcciones Ethernet". STEP 7 muestra el cuadro de diálogo de configuración de direcciones Ethernet, en el que el proyecto de software se asocia a la dirección IP de la CPU que lo recibirá. Nota: la CPU S7-1200 no tiene dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente. Para más información, consulte el apartado "Asignar direcciones IP (Internet Protocol)" (Página 599).

	Para los protocolos de Ethernet TCP, ISO on TCP y UDP, utilice las "Propiedades" de la instrucción (TSEND_C, TRCV_C o TCON) para configurar las conexiones "local/interlocutor". La figura muestra las "Propiedades de conexión" de la ficha "Configuración" en el caso de una conexión ISO on TCP. Consulte "Configurar la vía de conexión local/interlocutor" (Página 596) para más información.
	Tras finalizar la configuración, cargue el proyecto en la CPU. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo. Para más información, consulte "Comprobar la red PROFINET" (Página 607).

Nota

Para establecer una conexión con la CPU, la tarjeta de interfaz de red (NIC) y la CPU deben estar en la misma clase de red y en la misma subred. Es posible configurar la tarjeta de interfaz de red para que coincida con la dirección IP predeterminada de la CPU o bien cambiar la dirección IP de la CPU para que coincida con la clase de red y la subred de la tarjeta de interfaz de red.

Para más información sobre cómo conseguirlo, consulte el apartado "Asignar direcciones IP (Internet Protocol)" (Página 599).

6.11 Sincronización horaria

El objetivo de la sincronización horaria de los relojes en tiempo real es tener un reloj maestro que sincronice todos los demás relojes locales. El reloj maestro sincroniza los relojes locales inicialmente y los resincroniza periódicamente para evitar los efectos de la desviación con el paso del tiempo.

En el caso del S7-1200 y sus componentes básicos locales, solo la CPU y algunos de los módulos CP tienen relojes en tiempo real que pueden requerir una sincronización. El reloj en tiempo real de la CPU se puede configurar para que se sincronice con un reloj maestro externo. El reloj maestro externo puede proporcionar la hora utilizando un servidor NTP o un CP del rack local del S7-1200 que esté conectado a un sistema SCADA que incorpore un reloj maestro.

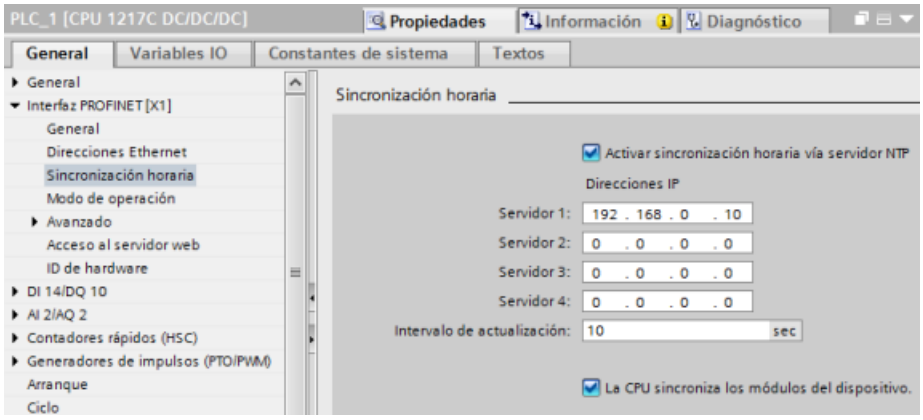
Consulte los CP S7-1200 (https://support.industry.siemens.com/cs/es/es/ps) en el soporte de producto de Siemens Industry Online Support si desea más información sobre todos los CP S7-1200 que soportan la función de sincronización horaria.

Ajustar el reloj en tiempo real

Hay tres formas de ajustar el reloj en tiempo real en la CPU S7-1200:

- Con el servidor NTP (Página 610)
- Uso de STEP 7
- Desde el programa de usuario
- Con un panel HMI

La sincronización horaria de los módulos CP con el reloj de la CPU se configura seleccionando la casilla de verificación "La CPU sincroniza los módulos del dispositivo", tal como se muestra:



En el ajuste predeterminado no están activadas ni la sincronización horaria con el servidor NTP ni la sincronización horaria de los relojes de CP con el reloj de la CPU.

La sincronización horaria del reloj de la CPU se configura independientemente de la sincronización horaria de los relojes de los módulos CP. En consecuencia, es posible activar la sincronización horaria de los relojes de los CP por la CPU si el reloj de la CPU se ajusta siguiendo uno de los métodos indicados arriba.

Se puede seleccionar el intervalo de actualización utilizando el servidor NTP. En el ajuste predeterminado, el intervalo de actualización del servidor NTP es de 10 segundos.

Cuando se activa la sincronización horaria en un módulo, STEP 7 pide que se seleccione la casilla de verificación "La CPU sincroniza los módulos del dispositivo" si todavía no se ha seleccionado en el cuadro de diálogo "Sincronización horaria" de la CPU. STEP 7 también emite una

advertencia si se ha configurado más de un reloj maestro para la sincronización horaria (p. ej. si se ha activado la sincronización horaria en más de un CP o tanto en la CPU como en un módulo).

Nota

La activación de la sincronización horaria en un CP hace que el CP ajuste el reloj de la CPU.

Si selecciona "La CPU sincroniza los módulos del dispositivo" en el cuadro de diálogo "Sincronización horaria" de la CPU, la CPU será el reloj maestro. Los módulos CP se sincronizan entonces con el reloj de la CPU.

Nota

Configure únicamente una fuente horaria para la CPU. Si la CPU recibe sincronizaciones horarias de más de una fuente (servidor NTP y módulo CP, por ejemplo) podrían darse conflictos al actualizar la hora. Las sincronizaciones horarias de varias fuentes podrían afectar las instrucciones y los eventos que se basan en la hora.

Principios básicos de programación 7

7.1 Directrices para diseñar un sistema PLC

Al diseñar un sistema PLC es posible seleccionar entre diferentes métodos y criterios. Las directrices generales siguientes pueden aplicarse a un gran número de proyectos. Por supuesto que es necesario respetar las directrices corporativas y las prácticas usuales aprendidas y aplicadas.

Tabla 7-1 Directrices para diseñar un sistema PLC

Pasos recomendados	Tareas
Dividir el proceso o máquina	Divida el proceso o máquina en secciones independientes. Estas secciones determinan los límites entre los controladores e influyen en las especificaciones funcionales y la asignación de recursos.
Crear las especificaciones funcionales	Describa el funcionamiento de cada una de las secciones del proceso o máquina, tales como las entradas y salidas, la descripción funcional de la operación, los estados que deben adoptarse antes de que puedan entrar en acción los actuadores (como p. ej. electroválvulas, motores o accionamientos), la descripción de la interfaz de operador y cualquier interfaz con otras secciones del proceso o máquina.
Diseñar los circuitos de seguridad	Determine los equipos que puedan requerir cableado fijo por motivos de seguridad. Recuerde que los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez un arranque inesperado o cambios de funcionamiento de la maquinaria. El funcionamiento inesperado o incorrecto de la maquinaria puede causar lesiones corporales o daños materiales considerables. Por tanto, prevea dispositivos de protección electromecánicos (que funcionen independientemente del PLC) para evitar las condiciones no seguras. Las siguientes tareas deben incluirse en el diseño de circuitos de seguridad: • Definir el funcionamiento erróneo o inesperado de los actuadores que pudiera resultar peligroso. • Definir las condiciones que garanticen un funcionamiento seguro y determinar cómo detectar estas condiciones, independientemente del PLC. • Definir cómo el PLC y los módulos de ampliación deben influir en el proceso al conectarse y desconectarse la alimentación eléctrica, así como al detectarse errores. Utilice esta información sólo para proyectar el funcionamiento normal y el funcionamiento anormal esperado. Por motivos de seguridad, no conviene fiarse del supuesto más favorable. • Prever dispositivos de parada de emergencia manual o dispositivos de protección electromecánicos que impidan el funcionamiento peligroso, independientemente del PLC. • Proporcionar información de estado apropiada desde los circuitos independientes al PLC para que el programa y las interfaces de operador dispongan de la información necesaria. • Definir otros requisitos adicionales de seguridad para el funcionamiento seguro del proceso.
Planificar la seguridad del sistema	Determine qué nivel de protección (Página 160) necesita para el acceso al proceso. Puede proteger por contraseña las CPU y los bloques de programa frente al acceso no autorizado.

Pasos recomendados	Tareas
Determinar las estaciones de operador	Según los requisitos de las especificaciones funcionales, cree los siguientes dibujos de las estaciones de operador: • Dibujo general de la ubicación de todas las estaciones de operador con respecto al proceso o máquina • Dibujo de la disposición mecánica de los dispositivos de la estación de operador, p. ej. display, interruptores y lámparas • Esquemas eléctricos con las E/S asociadas del PLC y los módulos de señales
Crear los dibujos de configuración	Según los requisitos de las especificaciones funcionales, cree dibujos de configuración de los equipos de control: • Dibujo general de la ubicación de todos los PLCs con respecto al proceso o máquina • Dibujo de la disposición mecánica de todos los PLCs y módulos de E/S, incluyendo los armarios y otros equipos. • Esquemas eléctricos de todos los PLCs y módulos de E/S, incluyendo los números de referencia de los dispositivos, las direcciones de comunicación y las direcciones de E/S.
Crear una lista de nombres simbólicos	Cree una lista de los nombres simbólicos correspondientes a las direcciones absolutas. Incluya no sólo las E/S físicas, sino también los demás elementos (p. ej. los nombres de variables) que se utilizarán en el programa.

7.2 Estructurar el programa de usuario

Al crear el programa de usuario para las tareas de automatización, las instrucciones del programa se insertan en bloques lógicos:

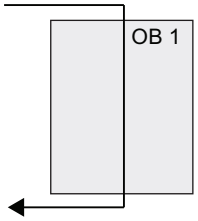
- Un bloque de organización (OB) reacciona a un evento específico en la CPU y puede interrumpir la ejecución del programa de usuario. El bloque predeterminado para la ejecución cíclica del programa de usuario (OB 1) ofrece la estructura básica para el programa de usuario. Si se incluyen otros OBs en el programa, éstos interrumpen la ejecución del OB 1. Los demás OBs ejecutan funciones específicas, tales como tareas de arranque, procesamiento de alarmas y tratamiento de errores, o ejecución de un código de programa específico en determinados intervalos.
- Un bloque de función (FB) es una subrutina que se ejecuta cuando se llama desde otro bloque lógico (OB, FB o FC). El bloque que efectúa la llamada transfiere parámetros al FB e identifica un bloque de datos determinado (DB) que almacena los datos de la llamada o instancia específica de este FB. La modificación del DB instancia permite a un FB genérico controlar el funcionamiento de un conjunto de dispositivos. Por ejemplo, un solo FB puede controlar varias bombas o válvulas con diferentes DBs de instancia que contienen los parámetros operativos específicos de cada bomba o válvula.
- Una función (FC) es una subrutina que se ejecuta cuando se llama desde otro bloque lógico (OB, FB o FC). La FC no tiene un DB instancia asociado. El bloque que efectúa la llamada transfiere los parámetros a la FC. Los valores de salida de la FC deben escribirse en una dirección de la memoria o en un DB global.

Seleccionar el tipo de estructura del programa de usuario

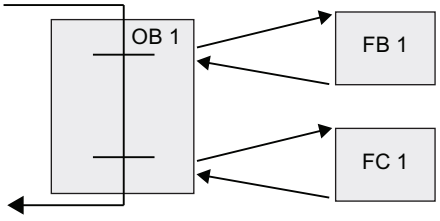
Según los requisitos de la aplicación, es posible seleccionar una estructura lineal o modular para crear el programa de usuario:

- Un programa lineal ejecuta todas las instrucciones de la tarea de automatización de forma secuencial, es decir, una tras otra. Generalmente, el programa lineal deposita todas las instrucciones del programa en el OB encargado de la ejecución cíclica del programa (OB 1).
- Un programa modular llama bloques de función específicos que ejecutan determinadas tareas. Para crear una estructura modular, la tarea de automatización compleja se divide en tareas subordinadas más pequeñas, correspondientes a las funciones tecnológicas del proceso. Cada bloque lógico provee el segmento del programa para cada tarea subordinada. El programa se estructura llamando uno de los bloques lógicos desde otro bloque.

Estructura lineal:



Estructura modular:



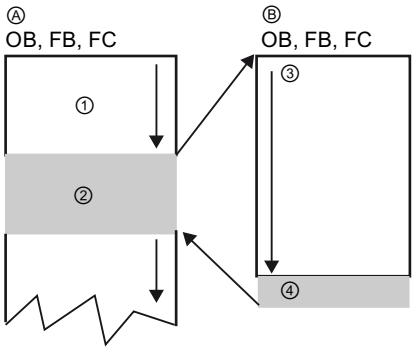
Creando bloques lógicos genéricos que pueden reutilizarse en el programa de usuario, es posible simplificar el diseño y la implementación del programa de usuario. La utilización de bloques lógicos genéricos ofrece numerosas ventajas:

- Es posible crear bloques lógicos reutilizables para tareas estándar, tales como el control de una bomba o motor. También es posible almacenar estos bloques lógicos genéricos en una librería, de manera que puedan ser utilizados por diferentes aplicaciones o soluciones.
- El programa de usuario puede dividirse en componentes modulares para las tareas funcionales, facilitando así su comprensión y gestión. Los componentes modulares ayudan no sólo a estandarizar el diseño del programa, sino que también pueden facilitar y agilizar la actualización o modificación del código del programa.
- La creación de componentes modular simplifica la depuración del programa. Dividiendo el programa completo en segmentos de programa modulares, es posible comprobar las funciones de cada bloque lógico a medida que se va desarrollando.
- La creación de componentes modulares para las distintas funciones tecnológicas permite simplificar y reducir el tiempo de puesta en marcha de la aplicación.

7.3 Utilizar bloques para estructurar el programa

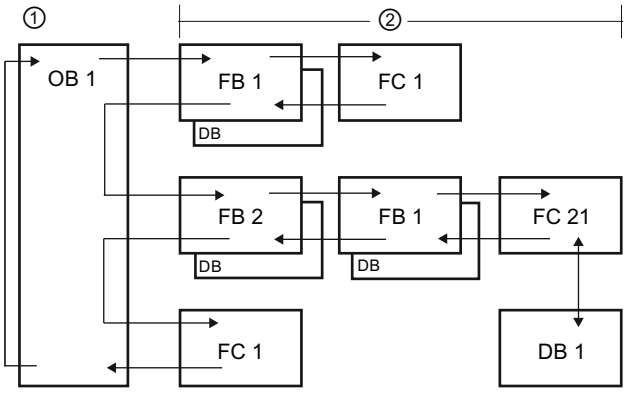
Diseñando bloques FB y FC que ejecuten tareas genéricas, se crean bloques lógicos modulares. El programa se estructura luego, de manera que otros bloques lógicos llamen estos bloques modulares reutilizables. El bloque que efectúa la llamada transfiere los parámetros específicos del dispositivo al bloque llamado.

Cuando un bloque lógico llama otro bloque lógico, la CPU ejecuta el código del programa en el bloque llamado. Una vez finalizada la ejecución del bloque llamado, la CPU reanuda la ejecución del bloque que ha efectuado la llamada. El procesamiento continúa con la ejecución de la instrucción siguiente a la llamada de bloque.



A Bloque que llama
B Bloque llamado (o que interrumpe)
① Ejecución del programa
② Instrucción o evento que inicia la ejecución de otro bloque
③ Ejecución del programa
④ Fin del bloque (regresa al bloque que llama)

Las llamadas de bloque pueden anidarse para crear una estructura más modular. En el ejemplo siguiente, la profundidad de anidamiento es 3: El OB de ciclo de programa más 3 niveles de llamadas de bloques lógicos.



① Inicio del ciclo
② Profundidad de anidamiento

Nota: La profundidad de anidamiento máxima es de seis. Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

7.3.1 Bloque de organización (OB)

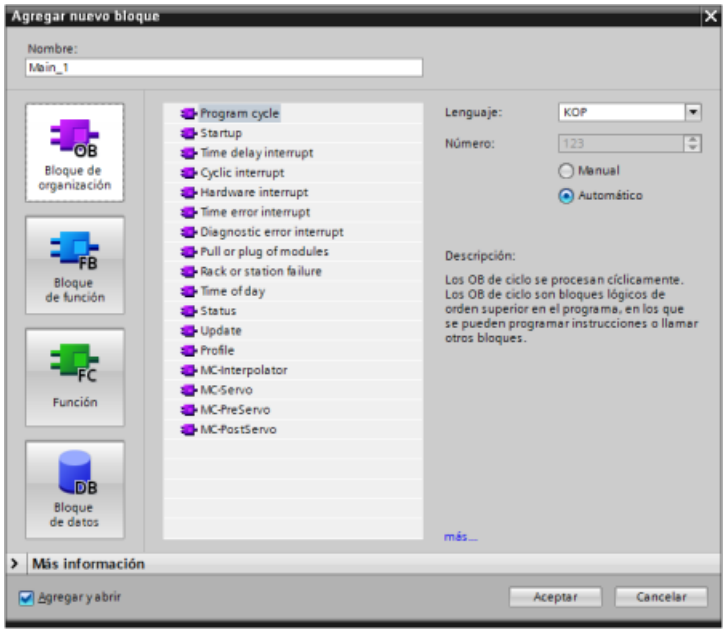
Los bloques de organización permiten estructurar el programa. Estos bloques sirven de interfaz entre el sistema operativo y el programa de usuario. Los OBs son controlados por eventos. Un evento, p. ej. una alarma de diagnóstico o un intervalo, hace que la CPU ejecute un OB. Algunos OB tienen eventos de arranque y comportamiento en arranque predefinidos.

El OB de ciclo contiene el programa principal. Es posible incluir más de un OB de ciclo en el programa de usuario. En el estado operativo RUN, los OB de ciclo se ejecutan en el nivel de prioridad más bajo y pueden ser interrumpidos por todos los demás tipos de eventos. El OB de arranque no interrumpe el OB de ciclo, puesto que la CPU ejecuta el OB de arranque antes de pasar al estado operativo RUN.

Tras finalizar el procesamiento de los OB de ciclo, la CPU vuelve a ejecutarlos inmediatamente. Esta ejecución cíclica es el tipo de procesamiento "normal" que se utiliza para los controladores lógicos programables. En numerosas aplicaciones, el programa de usuario entero está contenido en un solo OB de ciclo.

Es posible crear otros OB para ejecutar funciones específicas, tales como el procesamiento de alarmas y el tratamiento de errores, o la ejecución de un código de programa específico en determinados intervalos. Estos OB interrumpen la ejecución de los OB de ciclo.

Utilice el diálogo "Agregar nuevo bloque" para crear OB nuevos en el programa de usuario.



El procesamiento de alarmas siempre está controlado por eventos. Cuando ocurre un evento, la CPU interrumpe la ejecución del programa de usuario y llama el OB configurado para procesar ese evento. Una vez finalizada la ejecución del OB de alarma, la CPU reanuda la ejecución del programa de usuario en el punto de interrupción.

La CPU determina el orden de procesamiento de los eventos de alarma por prioridad. Se pueden asignar múltiples eventos de alarma a la misma clase de prioridad. Para más información, consulte los temas relativos a bloques de organización (Página 75) y a la ejecución del programa de usuario (Página 67).

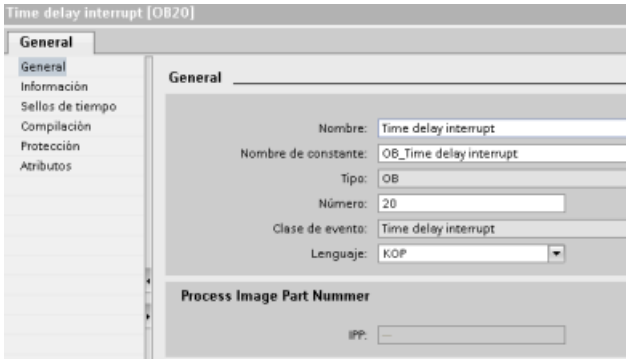
Creación de OB adicionales

Puede crear varios OB para el programa de usuario, incluso para los eventos de OB correspondientes a los OB de ciclo y de arranque. Use el cuadro de diálogo "Agregar nuevo bloque" para crear un OB e introduzca un nombre para el OB.

Si se crean varios OBs de ciclo de programa, la CPU ejecutará cada uno de ellos en una secuencia numérica, comenzando con el OB con el número menor (p. ej. OB 1).: Ejemplo: Tras finalizar el primer OB de ciclo (p. ej. OB 1), la CPU ejecuta el OB de ciclo con el siguiente número más alto.

Configuración de las propiedades de un OB

Las propiedades de un OB pueden modificarse. Por ejemplo, se puede configurar el número de OB o el lenguaje de programación.



Nota

Recuerde que se puede asignar un número de memoria imagen parcial de proceso a un OB que se corresponda con IPP0, IPP1, IPP2, IPP3 o IPP4. Si introduce un número para la memoria imagen parcial de proceso, la CPU crea la memoria imagen parcial en cuestión. Consulte el tema "Ejecución del programa de usuario (Página 67)" para obtener una explicación sobre memorias imagen parciales de proceso.

7.3.2 Función (FC)

Una función (FC) es un bloque lógico que, por lo general, realiza una operación específica en un conjunto de valores de entrada. La FC almacena los resultados de esta operación en posiciones de memoria. Por ejemplo, las FC se utilizan para ejecutar operaciones estándar y reutilizables (como cálculos matemáticos) o funciones tecnológicas (como para controles individuales que utilizan lógica de bits). Una FC también se puede llamar varias veces en diferentes puntos de un programa. Esto facilita la programación de tareas que se repiten con frecuencia.

Una FC no tiene ningún bloque de datos instancia asociado (DB). La FC usa la pila de datos locales para los datos temporales utilizados para calcular la operación. Los datos temporales no se almacenan. Para almacenar los datos de forma permanente es preciso asignar el valor de salida a una posición de memoria global, p. ej. el área de marcas o un DB global.

7.3.3 Bloque de función (FB)

Un bloque de función (FB) es un bloque lógico que utiliza un bloque de datos instancia para sus parámetros y datos estáticos. Los FBs tienen una memoria variable ubicada en un bloque de datos (DB) o DB "instancia". El DB instancia ofrece un bloque de memoria asociado a esa instancia (o llamada) del FB y almacena datos una vez que haya finalizado el FB. Es posible asociar distintos DBs de instancia a diferentes llamadas del FB. Los DBs instancia permiten utilizar un FB genérico para controlar varios dispositivos. El programa se estructura de manera que un bloque lógico llame un FB y un DB instancia. La CPU ejecuta luego el código del programa en ese FB y almacena los parámetros del bloque y los datos locales estáticos en el DB instancia. Cuando finaliza la ejecución del FB, la CPU regresa al bloque lógico que ha llamado el FB. El DB instancia conserva los valores de esa instancia del FB. Estos valores están disponibles para las llamadas posteriores al bloque de función, bien sea en el mismo ciclo o en otros ciclos.

Bloques lógicos reutilizables con memoria asociada

Por lo general, los FBs se utilizan para controlar tareas o dispositivos cuya operación no finaliza dentro de un ciclo. Para almacenar los parámetros operativos de manera que sea posible acceder rápidamente a ellos de un ciclo a otro, todo FB del programa de usuario tiene uno o más DBs instancia. Cuando se llama un FB, se especifica también un DB instancia que contiene los parámetros del bloque y los datos locales estáticos de esa llamada o "instancia" del FB. El DB instancia conserva estos valores una vez finalizada la ejecución del FB.

Si el FB se diseña para realizar tareas de control genéricas, es posible reutilizarlo para varios dispositivos, seleccionando diferentes DB instancia para las distintas llamadas del FB.

Un FB guarda los parámetros de entrada, salida, entrada y salida y estáticos en un DB de instancia.

También puede modificar y cargar la interfaz de bloque de función en estado operativo RUN (Página 1229).

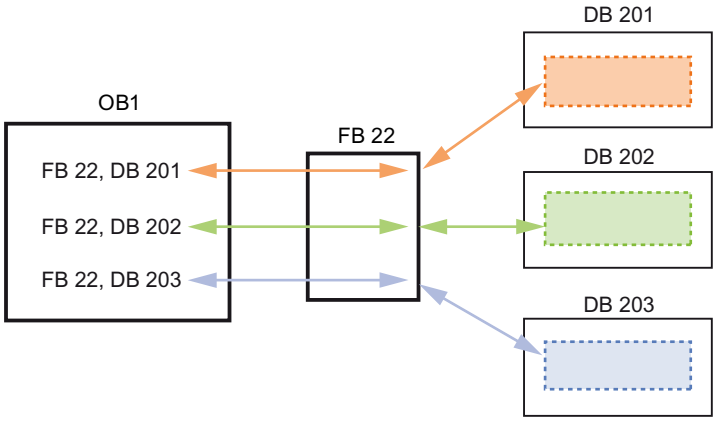
Asignar el valor inicial en el DB instancia

El DB instancia almacena un valor predeterminado y un valor iniciar para cada parámetro. El valor inicial proporciona el valor que debe utilizarse cuando se ejecuta el FB. Posteriormente, el valor inicial puede modificarse durante la ejecución del programa de usuario.

La interfaz del FB también dispone de una columna "Valor predeterminado" que permite asignar un nuevo valor inicial a los parámetros mientras se escribe el código del programa. Este valor predeterminado en el FB se transfiere posteriormente al valor inicial en el DB instancia asociado. Si no se asigna ningún valor inicial nuevo a un parámetro en la interfaz del FB, el valor predeterminado del DB instancia se copia en el valor inicial.

Utilizar un solo FB con DBs

La figura siguiente muestra un OB que llama un FB tres veces, utilizando un bloque de datos diferente para cada llamada. Esta estructura permite que un FB genérico controle varios dispositivos similares (p. ej. motores), asignando un bloque de datos instancia diferente a cada llamada de los distintos dispositivos. Cada DB instancia almacena los datos (p. ej. velocidad, tiempo de aceleración y tiempo de operación total) de un dispositivo en particular.



En este ejemplo, el FB 22 controla tres dispositivos diferentes. El DB 201 almacena los datos operativos del primer dispositivo, el DB 202, los del segundo y, el DB 203, los del tercero.

7.3.4 Bloque de datos (DB)

Los bloques de datos (DB) se crean en el programa de usuario para almacenar los datos de los bloques lógicos. Todos los bloques del programa de usuario pueden acceder a los datos en un DB global. En cambio, un DB instancia almacena los datos de un bloque de función (FB) específico.

Los datos almacenados en un DB no se borran cuando finaliza la ejecución del bloque lógico asociado. Hay dos tipos de DBs, a saber:

- Un DB global almacena los datos de los bloques lógicos en el programa. Cualquier OB, FB o FC puede acceder a los datos en un DB global.
- Un DB instancia almacena los datos de un FB específico. La estructura de los datos en un DB instancia refleja los parámetros (Input, Output e InOut) y los datos estáticos del FB. (La memoria temporal del FB no se almacena en el DB instancia.)

Nota

Aunque el DB instancia refleja los datos de un FB específico, cualquier bloque lógico puede acceder a los datos en un DB instancia.

También puede modificar y cargar los bloques de datos en estado operativo RUN (Página 1229).

Bloques de datos de solo lectura

Un DB se puede configurar de manera que sea de sólo lectura:

1. Haga clic con el botón derecho del ratón en el DB en el árbol del proyecto y seleccione "Propiedades" del menú contextual.
2. En el cuadro de diálogo "Propiedades", seleccione "Atributos".
3. Seleccione la opción "Bloque de datos protegido contra escritura en el dispositivo" y haga clic en "Aceptar".

Bloques de datos optimizados y estándar

También es posible configurar que el acceso al bloque de datos sea optimizado. Si el bloque no está optimizado, se considerará un bloque de datos estándar Un DB estándar es compatible con las herramientas de programación clásicas de STEP 7 y las CPU S7-300 y S7-400 clásicas. Los bloques de datos con acceso optimizado no tienen una estructura fija definida. Los elementos de datos solo contienen un nombre simbólico en la declaración y no tienen una dirección fija en el bloque. La CPU almacena los elementos automáticamente en el área de memoria disponible del bloque de modo que no haya lagunas de memoria, lo cual permite optimizar el uso de la capacidad de memoria.

Para establecer el acceso optimizado de un bloque de datos, proceda del siguiente modo:

1. Amplíe la carpeta de bloques de programa en el árbol del proyecto STEP 7.
2. Haga clic con el botón derecho del ratón en el bloque de datos y seleccione "Propiedades" en el menú contextual.
3. Para los atributos, seleccione "Acceso optimizado al bloque".

Tenga en cuenta que el acceso al bloque optimizado es el valor predeterminado de los nuevos bloques de datos. Si deselecciona "Acceso optimizado al bloque", el bloque utiliza el acceso estándar.

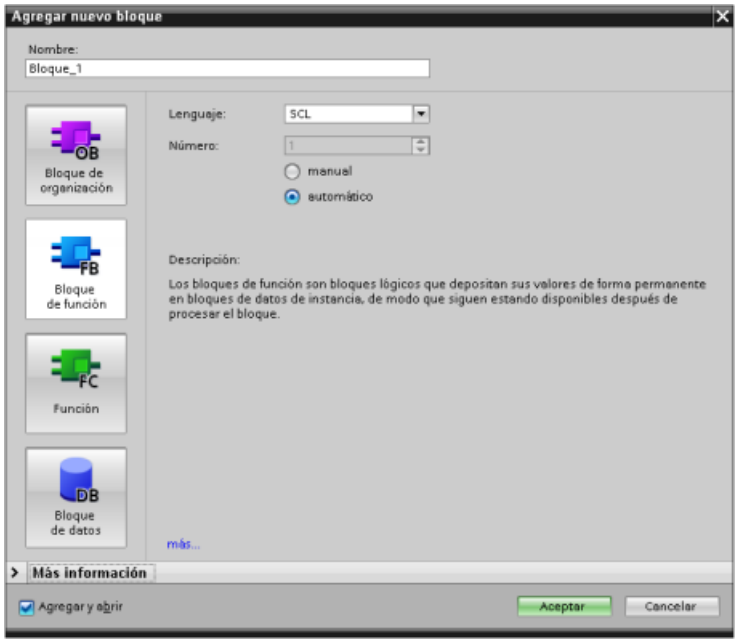
Nota

Tipo de acceso al bloque para un FB y su DB de instancia

Si el ajuste para el FB es "Acceso optimizado al bloque", asegúrese de que el ajuste del DB de instancia para dicho FB sea también "Acceso optimizado al bloque". De igual modo, si no se ha seleccionado "Acceso optimizado al bloque" para el FB porque el FB es de acceso estándar, asegúrese de que el DB de instancia también sea estándar, es decir, sin acceso optimizado al bloque.

Si los tipos de acceso al bloque no son compatibles, los cambios en los valores del parámetro IN/OUT del FB desde un HMI podrían perderse mientras se ejecuta el FB.

7.3.5 Crear bloques lógicos reutilizables



Utilice el diálogo "Agregar nuevo bloque" en "Bloques de programa" en el árbol del proyecto para crear OBs, FBs, FCs y DBs globales.

Al crear un bloque lógico se selecciona el lenguaje de programación para el bloque. El lenguaje de un DB no se selecciona, puesto que este solo almacena datos.

Activando la casilla de verificación "Agregar y abrir" (predeterminada) se abre el bloque lógico en la vista del proyecto.

Se pueden guardar en librerías objetos que se desea reutilizar. A cada proyecto hay conectada una librería de proyecto. Además de la librería del proyecto se puede crear un número indeterminado de librerías globales, que pueden utilizarse en varios proyectos. Dado que las librerías son compatibles entre sí, sus elementos pueden copiarse y moverse de una librería a otra.

Las librerías se utilizan, por ejemplo, para crear plantillas para bloques que primero se copian en la librería del proyecto y a continuación se desarrollan en él. Finalmente los bloques se copian de la librería del proyecto a una librería global. La librería global puede ser accesible para otros compañeros que trabajen en el mismo proyecto. Ellos utilizan los bloques y los adaptan a sus necesidades individuales, si es necesario.

Para más detalles sobre operaciones con librerías, consulte los temas relativos a librerías de la Ayuda en pantalla de STEP 7.

7.3.6 Transferir parámetros a bloques

Los bloques de función (FB) y las funciones (FC) tienen tres tipos de interfaz diferentes:

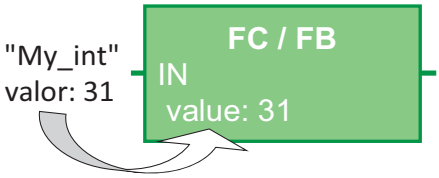
- IN
- IN/OUT
- OUT

Los FBs y las FCs reciben parámetros por los tipos de interfaz IN e IN/OUT. Los bloques procesan los parámetros y devuelven valores al emisor a través de los tipos de interfaz IN/OUT y OUT.

El programa de usuario transfiere parámetros utilizando uno de los dos métodos.

Llamada por valor

Cuando el programa de usuario transfiere un parámetro a una función mediante "llamada por valor", copia el valor del parámetro real al parámetro de entrada del bloque para el tipo de interfaz IN. Esta operación requiere memoria adicional para el valor copiado.



Cuando el programa de usuario llama el bloque, copia los valores.

Llamada por referencia

Cuando el programa de usuario transfiere un parámetro a una función mediante "llamada por referencia", hace referencia a la dirección del parámetro real para el tipo de interfaz IN/OUT y no copia el valor. Esta operación no requiere memoria adicional.



Cuando el programa de usuario llama el bloque, hace referencia a la dirección de los parámetros reales.

Nota

Por regla general, use el tipo de interfaz IN/OUT para las variables estructuradas (por ejemplo, ARRAY, STRUCT y STRING) con el fin de evitar ampliar innecesariamente la memoria de datos requerida.

Optimización del bloque y transferencia de parámetros

El programa de usuario transfiere los parámetros de la FC mediante "llamada por valor" para tipos de datos simples (como INT, DINT y REAL). Los tipos de datos complejos (como STRUCT, ARRAY y STRING) los transfiere mediante "llamada por referencia".

Por norma general, el programa de usuario transfiere parámetros del FB en el bloque de datos de instancia (DB) asociado al FB:

- El programa de usuario transfiere los tipos de datos simples (como INT, DINT y REAL) mediante "llamada por valor" copiando los parámetros al DB de instancia o de él.
- El programa de usuario copia tipos de datos complejos (como STRUCT, ARRAY y STRING) al DB de instancia o de él para tipos de parámetros IN y OUT.
- El programa de usuario transfiere tipos de datos complejos mediante "llamada por referencia" para el tipo de interfaz IN/OUT.

Los DBs pueden crearse como "optimizados" o "estándar" (no optimizados). Los bloques de datos optimizados son más compactos que los no optimizados. Es decir, el orden de los elementos de datos dentro del DB es diferente para DBs optimizados y no optimizados. Consulte la sección "Bloques optimizados" de la Guía de programación S7 para S7-1200/1500, STEP 7 (TIA Portal), 03/2014 (https://support.industry.siemens.com/cs/ww/es/view/81318674) para encontrar información sobre bloques optimizados.

Los FBs y las FCs se crean para procesar datos optimizados y no optimizados. Es posible seleccionar la casilla de verificación "Acceso optimizado al bloque" como uno de los atributos del bloque. El programa de usuario optimiza bloques de programa de forma predeterminada y estos esperan que los datos transferidos al bloque estén en formato optimizado.

Cuando el programa de usuario transfiere un parámetro complejo (por ejemplo STRUCT) a una función, el sistema comprueba el ajuste de optimización del bloque de datos que contiene la estructura y el ajuste de optimización del bloque de programa. Si se optimizan tanto el bloque de datos como la función, el programa de usuario transfiere STRUCT mediante "llamada por referencia". Lo mismo es válido si ni el bloque de datos ni la función están optimizados.

Sin embargo, si la optimización es diferente para la función y el bloque de datos (es decir, si se optimiza un bloque y el otro no), STRUCT debe convertirse al formato que espera la función. Así, por ejemplo, si se selecciona la opción "no optimizado" para el bloque de datos y "optimizado" para la función, un STRUCT en el bloque de datos debe convertirse a un formato optimizado antes de que la función pueda procesarlo. El sistema realiza esta conversión realizando una copia de STRUCT y convirtiéndola al formato optimizado que espera la función.

Resumiendo, cuando el programa de usuario transfiere un tipo de datos complejo (por ejemplo STRUCT) a una función como parámetro IN/OUT, la función espera que lo transfiera mediante "llamada por referencia":

- Si se selecciona "optimizado" o "no optimizado" tanto para el bloque de datos que contiene STRUCT como para la función, el programa de usuario transfiere los datos mediante "llamada por referencia".
- Si el bloque de datos y la función no se configuran con el mismo ajuste de optimización (uno está optimizado y el otro no), el sistema debe realizar una copia de STRUCT antes de transferirlo a la función. Como el sistema tiene que hacer esta copia de la estructura, esto convierte la "llamada por referencia" en "llamada por valor" en realidad.

Efecto de los ajustes de optimización en programas de usuario

La copia del parámetro puede causar un problema en un programa de usuario si un HMI o un OB de alarma modifica elementos de la estructura. Así, por ejemplo, hay un parámetro IN/OUT de una función (por lo general transferido mediante "llamada por referencia"), pero los ajustes de optimización del bloque de datos y de la función difieren:

1. Cuando el programa de usuario está listo para llamar la función, el sistema debe realizar una copia de la estructura para cambiar el formato de los datos de forma que coincidan con la función.
2. El programa de usuario llama la función con una referencia a la copia de la estructura.
3. Se produce un OB de alarma mientras se ejecuta la función y este OB de alarma cambia un valor en la estructura original.
4. La función finaliza y, puesto que la estructura es un parámetro IN/OUT, el sistema copia de nuevo los valores en la estructura original con el formato original.

La consecuencia de realizar la copia de la estructura para cambiar el formato es que se pierden los datos escritos por el OB de alarma. Lo mismo puede suceder cuando se escribe un valor con un HMI. El HMI puede interrumpir el programa de usuario y escribir un valor de la misma forma que un OB de alarma.

Hay varias formas de solucionar este problema:

- La mejor solución es que los ajustes de optimización del bloque de programa y del bloque de datos sean los mismos cuando se utilizan tipos de datos complejos (por ejemplo STRUCT). De este modo el programa de usuario siempre transfiere los parámetros mediante "llamada por referencia".
- Otra solución es que un OB de alarma o un HMI no modifiquen un elemento directamente en la estructura. El OB o HMI puede modificar otra variable y, posteriormente, el usuario puede copiar esta variable en la estructura en un punto concreto del programa de usuario.

7.4 Principios básicos de la coherencia de datos

La CPU conserva la coherencia de datos de todos los tipos de datos simples (p. ej. Word o DWord) y de todas las estructuras definidas por el sistema (p. ej. IEC_TIMERS o DTL). La lectura o escritura de valores no se puede interrumpir. (Por ejemplo, la CPU protege el acceso a un valor de palabra doble (DWord) hasta que se hayan leído o escrito los cuatro bytes de la palabra doble.) Para garantizar que los OBs de ciclo y de alarma no puedan escribir simultáneamente en la misma posición de memoria, la CPU no ejecuta un OB de alarma hasta que no haya finalizado la operación de lectura o escritura en el OB de ciclo.

Si el programa de usuario comparte varios valores entre un OB de ciclo y un OB de alarma en la memoria, dicho programa debe garantizar asimismo que estos valores se modifiquen o lean de forma coherente. Utilice las instrucciones DIS_AIRT (deshabilitar tratamiento de eventos de alarma) y EN_AIRT (habilitar tratamiento de eventos de alarma) del OB de ciclo de programa para proteger cualquier acceso a los valores compartidos.

- Inserte una instrucción DIS_AIRT en el bloque lógico para garantizar que un OB de alarma no pueda ejecutarse durante la operación de lectura o escritura.
- Inserte las instrucciones que leen o escriben los valores que podrían ser modificados por un OB de alarma.
- Inserte una instrucción EN_AIRT al final de la secuencia para cancelar la instrucción DIS_AIRT y permitir la ejecución del OB de alarma.

Una petición de comunicación de un dispositivo HMI o de otra CPU puede interrumpir asimismo la ejecución del OB de ciclo de programa. Las peticiones de comunicación también pueden causar problemas relacionados con la coherencia de datos. La CPU garantiza que las instrucciones del programa de usuario lean y escriban coherentemente los tipos de datos simples. Puesto que las comunicaciones interrumpen el programa de usuario de forma periódica, no es posible garantizar que el dispositivo HMI actualice simultáneamente varios valores en la CPU. Por ejemplo, los valores visualizados en la pantalla de un HMI podrían provenir de diferentes ciclos de la CPU.

Las instrucciones PtP (punto a punto), PROFINET (p. ej. TSEND_C y TRCV_C), instrucciones E/S descentralizadas para PROFINET (Página 369) y instrucciones E/S descentralizadas para PROFIBUS (Página 369) transfieren búferes de datos que podrían interrumpirse. La coherencia de datos de los búferes debe asegurarse evitando operaciones de lectura y escritura en los búferes, tanto en el OB de ciclo como en un OB de alarma. Si es necesario modificar los valores

de los búferes para estas instrucciones en un OB de alarma, utilice una instrucción DIS_AIRT para retardar las alarmas (un OB de alarma o una alarma de comunicación de un HMI u otra CPU) hasta que se ejecute una instrucción EN_AIRT.

Nota

La utilización de la instrucción DIS_AIRT retarda el procesamiento de los OBs de alarma hasta que se ejecuta la instrucción EN_AIRT, lo que afecta la latencia de alarmas (tiempo que transcurre desde un evento hasta que se ejecuta el OB de alarma) del programa de usuario.

7.5 Lenguaje de programación

STEP 7 ofrece los lenguajes de programación estándar siguientes para S7-1200:

- KOP (esquema de contactos) es un lenguaje de programación gráfico. Su representación se basa en esquemas (Página 188) de circuitos.
- FUP (diagrama de funciones) es un lenguaje de programación que se basa en los símbolos lógicos gráficos empleados en el álgebra (Página 189) booleana.
- SCL (structured control language) es un lenguaje de programación de alto nivel basado en texto (Página 189).

Al crear un bloque lógico, se debe seleccionar el lenguaje de programación que empleará dicho bloque.

El programa de usuario puede emplear bloques lógicos creados con cualquiera de los lenguajes de programación.

7.5.1 Esquema de contactos (KOP)

Los elementos de un esquema de circuitos, tales como los contactos normalmente cerrados y normalmente abiertos, así como las bobinas, se combinan para formar segmentos.



Para crear la lógica de operaciones complejas, es posible insertar ramas para los circuitos paralelos. Las ramas paralelas se abren hacia abajo o se conectan directamente a la barra de alimentación. Las ramas se terminan hacia arriba.

KOP ofrece instrucciones con cuadros para numerosas funciones, p. ej. matemáticas, temporizadores, contadores y transferencia.

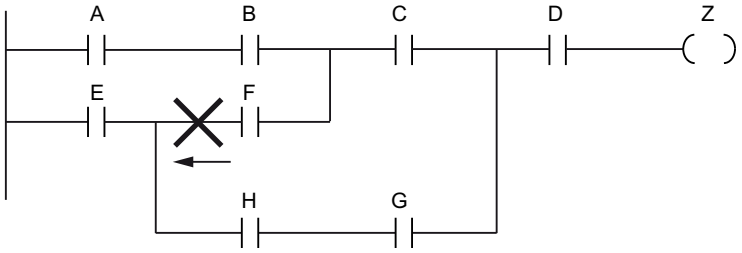
STEP 7 no limita el número de instrucciones (filas y columnas) de un segmento KOP.

Nota

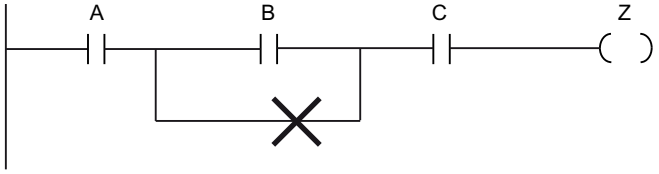
Todo segmento KOP debe terminar con una bobina o cuadro.

Tenga en cuenta las reglas siguientes al crear segmentos KOP:

- No se permite programar ramas que puedan ocasionar un flujo invertido de la corriente.

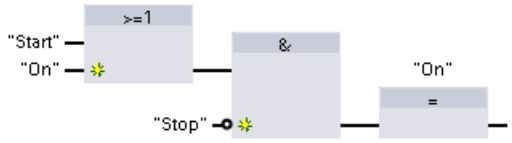


- No se permite programar ramas que causen cortocircuitos.



7.5.2 Diagrama de funciones (FUP)

Al igual que KOP, FUP es un lenguaje de programación gráfico. La representación de la lógica se basa en los símbolos lógicos gráficos del álgebra booleana.



Para crear la lógica de operaciones complejas, inserte ramas paralelas entre los cuadros.

Las funciones matemáticas y otras operaciones complejas pueden representarse directamente en combinación con los cuadros lógicos.

STEP 7 no limita el número de instrucciones (filas y columnas) de un segmento FUP.

7.5.3 SCL

El lenguaje de control estructurado (SCL, Structured Control Language) es un lenguaje de programación de alto nivel basado en PASCAL para las CPU de SIMATIC S7. SCL soporta la estructura de bloques de STEP 7 (Página 177). Su proyecto puede incluir bloques de programa en uno de los tres lenguajes de programación: SCL, KOP y FUP.

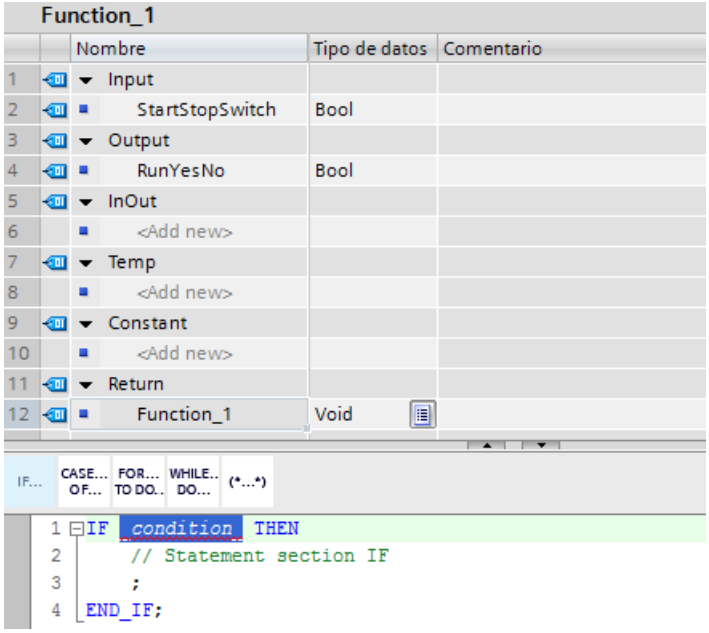
Las instrucciones de SCL emplean operadores de programación estándar, p. ej. para asignación (:=) o para funciones matemáticas (+ para la suma, - para la resta, * para la multiplicación y / para la división). SCL también utiliza operaciones de control de programa PASCAL estándar, tales como IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO y RETURN. Es posible utilizar cualquier referencia PASCAL para elementos sintácticos del lenguaje de programación SCL. Muchas otras instrucciones de SCL, como los temporizadores y contadores, se corresponden con las instrucciones de KOP y FUP. Para más información sobre determinadas instrucciones, véanse las instrucciones específicas de las secciones correspondientes a Instrucciones básicas (Página 209) e Instrucciones avanzadas (Página 325).

7.5.3.1 Editor de programas SCL

Es posible designar cualquier tipo de bloque (OB, FB o FC) para utilizar el lenguaje de programación en el momento de crear el bloque. STEP 7 cuenta con un editor de programas SCL que incluye los elementos siguientes:

- Sección de interface para definir los parámetros del bloque lógico
- Sección de código para el código del programa
- Árbol de instrucciones que contiene las instrucciones SCL soportadas por la CPU

El código SCL para la instrucción se introduce directamente en la sección de código. El editor incluye botones para las construcciones y comentarios de código habituales. Para obtener instrucciones más complejas, basta con arrastrar las instrucciones SCL del árbol de instrucciones y soltarlas en el programa. También se puede emplear cualquier editor de texto para crear un programa SCL y, a continuación, importar el archivo a STEP 7.



En la sección de interfaz del bloque lógico SCL se pueden declarar los tipos de parámetros siguientes:

- Input, Output, InOut y Ret_Val: estos parámetros definen las variables de entrada, las variables de salida y el valor de retorno del bloque lógico. El nombre de la variable introducida en este punto se emplea de forma local durante la ejecución del bloque lógico. Normalmente, no se emplea el nombre de variables globales en la tabla de variables.
- Static (solo FB; la figura de arriba corresponde a una FC): el bloque lógico utiliza variables estáticas para almacenar resultados intermedios estáticos en el bloque de datos de instancia. El bloque retiene datos estáticos hasta que se sobrescriben, lo cual puede ocurrir después de varios ciclos. Los nombres de los bloques, que este bloque llama como multiinstancia, también se almacenan en los datos locales estáticos.
- Temp: estos parámetros son variables temporales que se emplean durante la ejecución del bloque lógico.
- Constant: son valores constantes con nombre para el bloque lógico.

Si se llama el bloque lógico SCL desde otro bloque lógico, los parámetros del bloque lógico SCL aparecen como entradas o salidas.



En este ejemplo, las variables de "Start" y "On" (de la tabla de variables de proyecto) se corresponden con "StartStopSwitch" y "RunYesNo" en la tabla de declaración del programa SCL.

7.5.3.2 Expresiones y operaciones de SCL

Construir una expresión SCL

Una expresión SCL es una fórmula para calcular un valor. La expresión consiste en operandos y operadores (p. ej. *, /, + o -). Los operandos pueden ser variables, constantes o expresiones.

La evaluación de la expresión se desarrolla en un orden determinado, que está definido por los siguientes factores:

- Cada operador tiene una prioridad predefinida y la operación con mayor prioridad se ejecuta en primer lugar.
- Si tienen la misma prioridad, los operadores se procesan en una secuencia de izquierda a derecha.
- Se utilizan paréntesis para designar una serie de operadores que deben evaluarse conjuntamente.

El resultado de una expresión se puede utilizar para asignar un valor a una variable utilizada por el programa, como una condición que debe ser considerada en una instrucción de control, o como parámetros para otra instrucción SCL o para llamar un bloque lógico.

Tabla 7-2 Operadores en SCL

Tipo	Operación	Operador	Prioridad
Paréntesis	(*Expresión*)	(,)	1
Matemáticas	Alimentación	**	2
	Signo (más unario)	+	3
	Signo (menos unario)	-	3
	Multiplicación	*	4
	División	/	4
	Modulo	MOD	4
	Suma	+	5
	Resta	-	5
Comparación	Menor	<	6
	Menor o igual	<=	6
	Mayor	>	6
	Mayor o igual	>=	6
	Igual	=	7
	Diferente	<>	7
Lógica con bits	Negación (unaria)	NOT	3
	Combinación Y lógica	Y o &	8
	Combinación lógica O-exclusiva	XOR	9
	Combinación lógica O	O	10
Parametrización	Parametrización	:=	11

Como lenguaje de programación de alto nivel, SCL utiliza instrucciones estándar para tareas básicas:

- Instrucción de asignación: :=
- Funciones matemáticas: +, -, * y /
- Direccionamiento de variables globales: "<nombre de variable>" (nombre de la variable o nombre del bloque de datos entre comillas dobles)
- Direccionamiento de variables locales: #<nombre de variable> (nombre de la variable precedido del símbolo "#")

Los ejemplos siguientes muestran diferentes expresiones para usos distintos:

`"C" := #A+#B;`	Asigna la suma de dos variables locales a una variable
`"Data_block_1".Tag := #A;`	Asignación a una variable de bloque de datos
`IF #A > #B THEN "C" := #A;`	Condición de la instrucción IF-THEN
`"C" := SQRT (SQR (#A) + SQR (#B));`	Parámetros de la instrucción SQRT

Los operadores aritméticos pueden procesar varios tipos de datos numéricos. El tipo de datos resultante lo determina el tipo de datos de los operandos más significativos. Por ejemplo, una

operación de multiplicación que emplea un operando INT y un operando REAL da como resultado un valor REAL.

Instrucciones de control

Una instrucción de control es un tipo de expresión SCL especializada, que realiza las siguiente tareas:

- Ramificación del programa
- Repetición de secciones del código del programa SCL
- Salto a otras partes del programa SCL
- Ejecución condicionada

Las instrucciones de control de SCL incluyen IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO y RETURN.

Una instrucción única generalmente ocupa una línea de código. Es posible introducir varias instrucciones en una línea o dividir una instrucción en varias líneas de código para que el código resulte más comprensible. Los separadores (como tabulaciones, saltos de línea o espacios adicionales) se ignoran durante la comprobación de sintaxis. Una instrucción END termina la instrucción de control.

Los siguientes ejemplos muestran una instrucción de control FOR-TO-DO. (Ambas formas de codificación son válidas sintácticamente.)

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;
FOR x := 0 TO max DO
     sum := sum + value(x);
END_FOR;
```

Una instrucción de control también puede llevar una etiqueta. Una etiqueta se inserta colocando dos puntos (:) al comienzo de la instrucción:

```
Etiqueta: <instrucción>;
```

La ayuda en línea de STEP 7 ofrece una completa referencia del lenguaje de programación SCL.

Condiciones

Una condición es una expresión de comparación o una expresión lógica cuyo resultado es del tipo BOOL (con el valor TRUE o bien FALSE). El ejemplo siguiente muestra condiciones de diferentes tipos:

`#Temperatura > 50`
`#Contador <= 100`
`#CHAR1 < 'S'`
	Expresión de relación

`(#Alpha <> 12) AND NOT #Beta`	Comparación y expresión lógica
`5 + #Alpha`	Expresión aritmética

Una condición puede utilizar expresiones aritméticas:

- La condición de la expresión es TRUE si el resultado es cualquier valor diferente a cero.
- La condición de la expresión es FALSE si el resultado es igual a cero.

Llamar otros bloques lógicos desde el programa SCL

Para llamar otro bloque lógico en el programa de usuario, basta con introducir el nombre (o dirección absoluta) del FB o la FC que contiene los parámetros. En el caso de un FB, también hay que indicar un DB de instancia para llamarlo con el FB.

`<Nombre de DB> (lista de parámetros)`	Llamar como instancia única
`<#Nombre de instancia> (lista de parámetros)`	Llamar como multiinstancia
`"MyDB"(MyInput:=10, MyInOut:="Tag1");`	
`<Nombre de FC> (Lista de parámetros)`	Llamada estándar
`<Operando>:=<Nombre de FC> (Lista de parámetros)`	Llamar en una expresión
`"MyFC"(MyInput:=10, MyInOut:="Tag1");`	

También se pueden arrastrar bloques desde el árbol de navegación al editor del programa SCL y completar la asignación de parámetros.

Adición de comentarios de bloque al código SCL

Puede incluir un comentario de bloque en el código SCL mediante la inclusión del texto del comentario entre (* y *). Puede incluir tantas líneas de comentario como desee entre (* y *). El bloque de programa SCL puede incluir varios comentarios de bloque. A efectos de programación, el editor de SCL incluye un botón de comentarios de bloque junto con las instrucciones de control habituales:



Direccionamiento

Igual que en el caso de KOP y FUP, SCL permite utilizar variables (direccionamiento simbólico) o direcciones absolutas en el programa de usuario. SCL también permite utilizar una variable como índice de matriz.

Direccionamiento absoluto

`%I0.0` `%MB100`	Debe colocar el símbolo "%" delante de las direcciones absolutas. Sin el símbolo "%", STEP 7 genera un error de variable no definido durante el tiempo de compilación.

Direccionamiento simbólico

`"PLC_Tag_1"`	Variable en la tabla de variables PLC
`"Data_block_1".Tag_1`	Variable en un bloque de datos
`"Data_block_1".MyArray[#i]`	Elemento de una matriz de bloque de datos

7.5.3.3 Direccionamiento indexado con instrucciones PEEK y POKE

SCL ofrece instrucciones PEEK y POKE que permiten leer de bloques de datos, E/S o memoria, o escribir en ellos. Hay que proporcionar parámetros de offsets de byte, o de bit, específicos para la operación.

Nota

Para usar las instrucciones PEEK y POKE con bloques de datos deben usarse bloques de datos estándar (no optimizados). Nótese que las instrucciones PEEK y POKE solo transfieren datos. Desconocen los tipos de datos que hay en las direcciones.

```
PEEK(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_);
```

Lee el byte referenciado por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo de referencia al bloque de datos:

```
%MB100 := PEEK(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

Ejemplo de referencia a la entrada IB3:

```
%MB100 := PEEK(area:=16#81,
dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_);
```

Lee la palabra referenciada por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MW200 := PEEK_WORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_);
```

Lee la doble palabra referenciada por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MD300 := PEEK_DWORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_,
          bitOffset:=_in_);
```

Lee un booleano referenciado por el bitOffset y el byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
dbNumber:=1, byteOffset:=#ii,
bitOffset:=#j);
```

```
POKE(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     value:=_in_);
```

Escribe el valor (Byte, Word o DWord) en el byteOffset referenciado del bloque de datos, E/S o área de memoria referenciados.

Ejemplo de referencia al bloque de datos:

```
POKE(area:=16#84, dbNumber:=2,
byteOffset:=3, value:="Tag_1");
```

Ejemplo de referencia a la salida QB3:

```
POKE(area:=16#82, dbNumber:=0,
byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     bitOffset:=_in_,
     value:=_in_);
```

Escribe el valor booleano en el bitOffset y byteOffset referenciados del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
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_);
```

Escribe el número de "recuento" de bytes empezando por el offset de bytes referenciado del bloque de datos, E/S o área de memoria de origen referenciados en el byteOffset referenciado del bloque de datos, E/S o área de memoria de destino referenciados.

Ejemplo:

```
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);
```

Para las instrucciones PEEK y POKE, se aplican los valores siguientes de los parámetros "area", "area_src" y "area_dest". Para otras áreas distintas de los bloques de datos, el parámetro dbNumber debe ser 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

7.5.4 EN y ENO para KOP, FUP y SCL

Determinar el "flujo de corriente" (EN y ENO) para una instrucción

Algunas instrucciones (p. ej. matemáticas y de transferencia) proporcionan parámetros para EN y ENO. Estos parámetros se refieren al flujo de corriente en KOP y FUP y determinan si la instrucción se ejecuta en ese ciclo. SCL también permite activar el parámetro ENO para un bloque lógico.

- EN (Enable In) es una entrada booleana. Debe haber flujo de corriente (EN = 1) en esta entrada para que la instrucción de cuadro pueda ejecutarse. Si la entrada EN de un cuadro KOP se conecta directamente a la barra de potencia izquierda, la instrucción se ejecutará siempre.
- ENO (Enable Out) es una salida booleana. Si el cuadro tiene flujo de corriente en la entrada EN y ejecuta su función sin errores, la salida ENO transfiere el flujo de corriente (ENO = 1) al siguiente elemento. Si se detecta un error en la ejecución del cuadro, el flujo de corriente se interrumpe (ENO = 0) en la instrucción de cuadro que ha generado el error.

Tabla 7-3 Operandos para EN y ENO

Editor de programas	Entradas/salidas	Operandos	Tipo de datos
KOP	EN, ENO	Flujo de corriente	Bool
FUP	EN	I, I:P, Q, M, DB, Temp, flujo de corriente	Bool
	ENO	Flujo de corriente	Bool
SCL	EN[1]	TRUE, FALSE	Bool
	ENO[2]	TRUE, FALSE	Bool

[1] El uso de EN solo está disponible para bloques FB.

[2] El uso de ENO con el bloque lógico SCL es opcional. Hay que configurar el compilador SCL de forma que active ENO cuando acabe el bloque lógico.

Configuración de SCL para definir ENO

Para configurar el compilador SCL para que active ENO proceda del siguiente modo:

1. Elija el comando "Configuración" del menú "Opciones".

2. Expanda las propiedades "Programación PLC" y seleccione "SCL (Structured Control Language)".

3. Seleccione la opción "Activar ENO automáticamente".

Uso de ENO en el código del programa

Asimismo, puede usar ENO en el código del programa, por ejemplo asignando ENO a una variable PLC o evaluando ENO en un bloque local.

Ejemplos:

```
"MyFunction"
  ( IN1 := … ,
    IN2 := … ,
```

```
    OUT1 => #myOut,
    ENO => #statusFlag ); // Variable de PLC statusFlag que contiene
el valor de ENO

"MyFunction"
  ( IN1 := …
    IN2 := … ,
    OUT1 => #myOut,
    ENO => ENO );  // La marca de estado de bloque de "MyFunction"
                   // se almacena en el bloque local

IF ENO = TRUE THEN
    // Ejecutar código solo si MyFunction devuelve ENO como TRUE
```

Efecto de los parámetros Ret_Val o Status sobre ENO

Algunas instrucciones, como las de comunicación o de conversión de cadena, ofrecen un parámetro de salida que incluye información relacionada con el procesamiento de la instrucción. Así, por ejemplo, algunas instrucciones ofrecen un parámetro Ret_Val (valor de retorno), que suele ser un tipo de datos Int que incluye información de estado en un rango de -32768 a +32767. Otras instrucciones disponen de un parámetro Status que suele ser un tipo de datos Word que almacena información de estado en un rango de valores hexadecimales de 16#0000 a 16#FFFF. El valor numérico almacenado en un parámetro Ret_Val o Status determina el estado de ENO para dicha instrucción.

- Ret_Val: un valor de 0 a 32767 pone típicamente ENO = 1 (o TRUE). Un valor de -32768 a -1 pone típicamente ENO = 0 (o FALSE). Para evaluar Ret_Val, cambie la representación a hexadecimal.
- Status: un valor de 16#0000 a 16#7FFF pone típicamente ENO = 1 (o TRUE). Un valor de 16#8000 a 16#FFFF pone típicamente ENO = 0 (o FALSE).

Las instrucciones cuya ejecución dura más de un ciclo suelen disponer de un parámetro Busy (Bool) para indicar que la instrucción está activa pero aún no se ha completado su ejecución. Dichas instrucciones también suelen incluir un parámetro Done (Bool) y un parámetro Error (Bool). Done indica que la instrucción se ha completado sin errores y Error indica que la instrucción se ha completado con una condición de error.

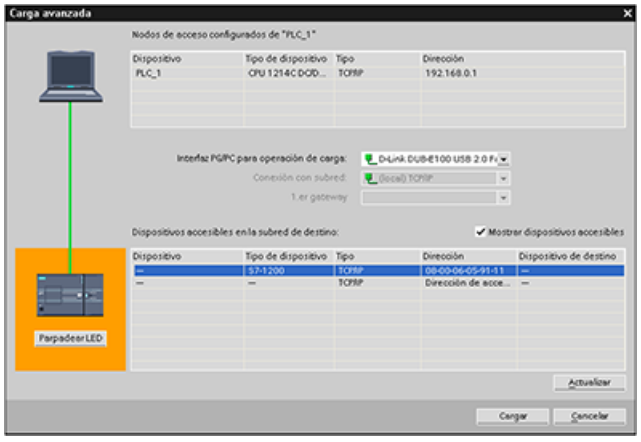
- Cuando Busy = 1 (o TRUE), ENO = 1 (o TRUE).
- Cuando Done = 1 (o TRUE), ENO = 1 (o TRUE).
- Cuando Error = 1 (o TRUE), ENO = 0 (o FALSE).

Consulte también

OK (comprobar validez) y NOT_OK (comprobar invalidez) (Página 233)

7.6 Cargar los elementos del programa en la CPU

Los elementos del proyecto se pueden cargar desde la programadora a la CPU. Cuando se descarga un proyecto, la CPU almacena el programa de usuario (OB, FC, FB y DB) en la memoria de carga interna o, en caso de haber una SIMATIC Memory Card, en la memoria de carga externa (la tarjeta).



El proyecto se puede cargar de la programadora en la CPU desde cualquiera de las ubicaciones siguientes:

- Árbol del proyecto: Haga clic con el botón derecho del ratón en el elemento de programa deseado y elija el comando "Cargar en dispositivo" del menú contextual.
- Menú online: Elija el comando "Cargar en dispositivo".
- Barra de herramientas: Haga clic en el botón "Cargar en dispositivo".
- Configuración de dispositivos: Haga clic con el botón derecho del ratón en la CPU y seleccione los elementos que deben descargarse.

Tenga en cuenta que si ha aplicado el enlace dinámico con una contraseña obligatoria (Página 168) a cualquiera de los bloques del programa, debe introducir la contraseña de los bloques protegidos para descargarlos. Si ha configurado este tipo de protección contra copia para varios bloques, debe introducir la contraseña para cada uno de los bloques protegidos para descargarlos.

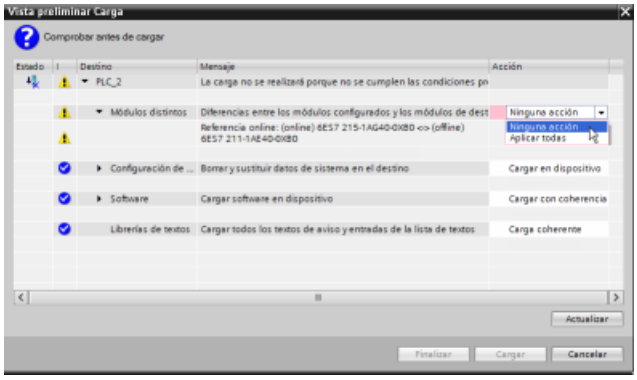
Nota

Cuando se carga un programa no se borran ni se cambian los valores existentes en la memoria remanente. Si desea borrar la memoria remanente antes de realizar una carga, restablezca la configuración de fábrica de la CPU antes de cargar el programa.

También puede cargar un proyecto de panel para los paneles HMI Basic (Página 32) desde el TIA Portal a la Memory Card de la CPU S7-1200.

Descarga cuando la CPU configurada es diferente de la CPU conectada

STEP 7 y el S7-1200 permiten la descarga si la CPU conectada tiene capacidad suficiente para almacenar una descarga de la CPU configurada, de acuerdo con los requisitos de memoria del proyecto y la compatibilidad de las E/S. La configuración y el programa de una CPU pueden descargarse en una CPU de mayor capacidad, por ejemplo, de una CPU 1211C DC/DC/DC a una CPU 1215C DC/DC/DC, porque las E/S son compatibles y la memoria es suficiente. En este caso, la operación de descarga muestra una advertencia, "Diferencias entre los módulos configurados y los módulos de destino (online)", junto con las referencias y versiones de firmware, en el cuadro de diálogo "Vista preliminar". Hay que elegir entre "Ninguna acción" si no desea que se inicie la descarga o "Aplicar todas" si desea iniciar la descarga:



Nota

Cuando se pasa a online (Página 1201) después de descargar la CPU configurada en una CPU conectada distinta, el proyecto de la CPU configurada se ve en el árbol del proyecto con indicadores de estado online. En cambio, en la vista Online y diagnóstico se muestra el tipo de módulo CPU conectado realmente.

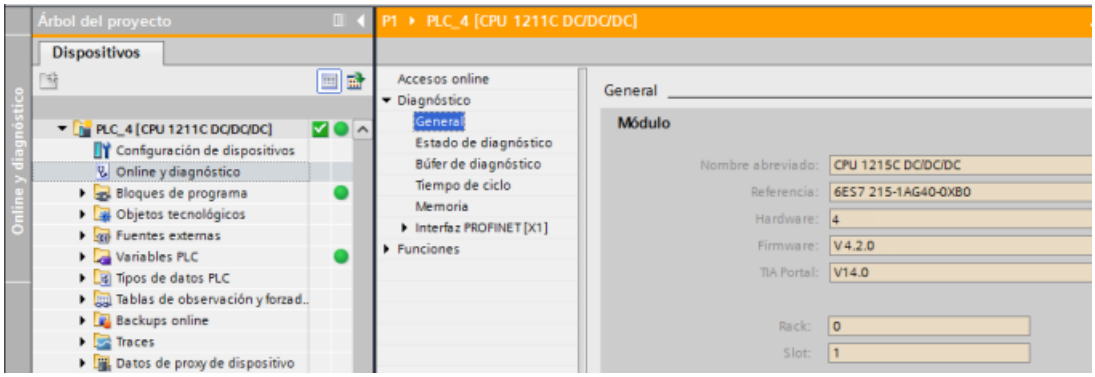


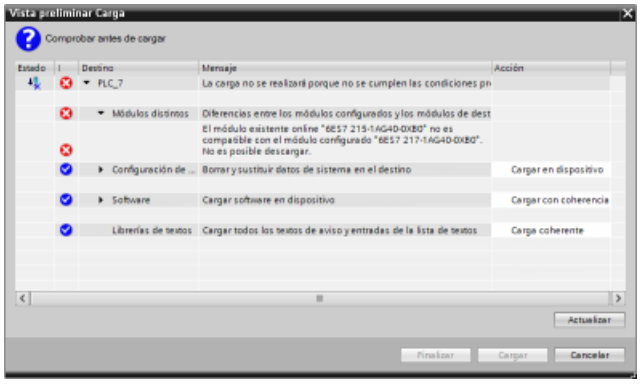
Figura 7-1 Vista online cuando la CPU configurada es diferente de la CPU conectada

Evidentemente, es posible cambiar el dispositivo (Página 152) en la configuración de dispositivos, de modo que la CPU configurada tenga el mismo tipo de módulo que la CPU conectada. El cuadro de diálogo "Cambiar dispositivo" ofrece todos los detalles de compatibilidad cuando se intenta cambiar un dispositivo.

STEP 7 y el S7-1200 prohíben la descarga si la CPU conectada no tiene capacidad suficiente para almacenar una descarga de la CPU configurada; así, por ejemplo, no es posible descargar la configuración hardware y el programa en los casos siguientes:

- De una CPU 1215C DC/DC/DC a una CPU 1212C DC/DC/DC debido a una memoria de trabajo insuficiente
- De una CPU 1211C DC/DC/relé a una CPU 1211C DC/DC/DC debido a diferencias en las E/S
- De una CPU 1217C DC/DC/DC a cualquier CPU 1211C, CPU 1212C, CPU 1214C o CPU 1215C debido a las salidas de 1,5 V DC de la CPU 1217C
- De una CPU 1214C V4.2.x a una CPU 1214C V4.0 debido a la incompatibilidad de las versiones de firmware

En estos casos, el cuadro de diálogo "Vista preliminar" muestra un error:



CPU con protección de datos de configuración confidenciales del PLC

Si ha configurado Protección de datos de configuración confidenciales del PLC (Página 161) recuerde lo siguiente:

- Si el dispositivo no dispone de la contraseña para la protección de datos de configuración confidenciales del PLC, TIA Portal le solicitará en la primera descarga que introduzca la contraseña de protección de datos de configuración confidenciales del PLC.
- Si el dispositivo ya dispone de la contraseña de protección de datos de configuración confidenciales del PLC, las contraseñas del proyecto de STEP 7 y del dispositivo deben coincidir. Si las contraseñas no coinciden, no podrá descargar el proyecto a la CPU. Deberá borrar la contraseña para la protección de datos de configuración confidenciales del PLC o establecerla de modo que coincida con la contraseña del dispositivo. También es posible establecer o borrar la contraseña para la protección de datos de configuración confidenciales del PLC desde Online y diagnóstico (Página 1205).

Recuperación de una descarga fallida

Si la descarga falla, la ficha de información de la ventana de inspección muestra el motivo. El búfer de diagnóstico también ofrece información. Después de una descarga fallida, proceda del siguiente modo para poder llevar a cabo una descarga de forma correcta:

1. Corrija el problema tal como se describe en el mensaje de error.
2. Vuelva a intentar la descarga.

En contadas ocasiones, la descarga se produce pero la desconexión y conexión posterior de la CPU falla. En este caso, se mostrará un error en el búfer de diagnóstico del tipo siguiente:

- 16# 02:4175 -- error de CPU: error de evaluación de la Memory Card: versión desconocida o incompatible del tipo de tarjeta actual con la descripción de la configuración de la CPU: No ha terminado ninguna función de la Memory Card o se ha cancelado, se ha ajustado un nuevo bloqueo de arranque: ..- falta la Memory Card, es del tipo erróneo, tiene un contenido erróneo o está protegida

Si esto ocurriera y los intentos sucesivos también fallaran, habrá que borrar la memoria de carga interna o externa:

1. Si se utiliza una memoria de carga interna, restablezca la configuración de fábrica de la CPU.
2. Si se utiliza una SIMATIC Memory Card, extráigala y elimine su contenido (Página 125) antes de volver a insertarla.
3. Descargue la configuración hardware y el software.

Consulte también

Sincronizar la CPU online y el proyecto offline (Página 202)

7.7 Sincronizar la CPU online y el proyecto offline

Cuando se descargan bloques de proyecto en la CPU, esta puede detectar si los bloques o las variables han cambiado en la CPU online desde la última descarga. En estos casos, la CPU ofrece la posibilidad de sincronizar los cambios. Esto significa que los cambios en la CPU online pueden cargarse en el proyecto antes de descargar el proyecto en la CPU. Los cambios en la CPU online pueden deberse a varios factores:

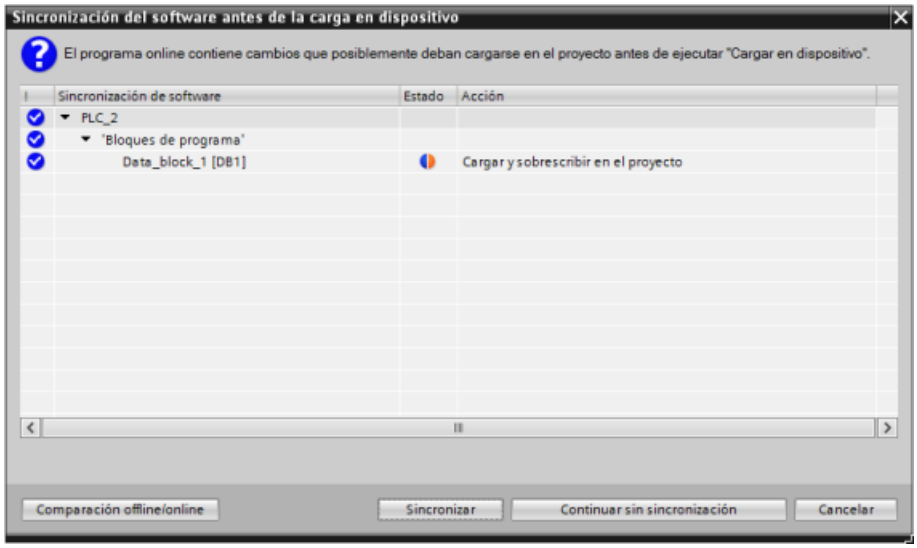
- Cambios en los valores iniciales de las variables de bloques de datos en tiempo de ejecución, p. ej. por parte de la instrucción WRIT_DBL (Página 518) o debido a la carga de una receta
- Una descarga de un proyecto "secundario" (un proyecto distinto del que originó la última descarga) en el que se den una o varias de las condiciones siguientes:
  - La CPU online incluye bloques de programa que no existen en el proyecto.
  - Las variables de bloques de datos o los atributos de bloques son diferentes entre el proyecto offline y la CPU online.
  - En la CPU online hay variables PLC que no existen en el proyecto offline.

Nota

Si se editan bloques o variables en el proyecto que se utilizaron para la última descarga no es necesario hacer selecciones sobre la sincronización. STEP 7 y la CPU detectan que los cambios en el proyecto offline son más recientes que la CPU online e inician una operación de descarga normal.

Opciones de sincronización

Cuando se descarga un proyecto en la CPU, se visualiza el cuadro de diálogo de sincronización si STEP 7 detecta que los bloques de datos o las variables de la CPU online son más recientes que los valores del proyecto. Así, por ejemplo, si el programa S7 ha ejecutado WRIT_DBL y ha cambiado un valor de arranque de una variable en Data_block_1, STEP 7 muestra el siguiente cuadro de diálogo de configuración al iniciar una descarga:



Este cuadro de diálogo lista los bloques de programa en los que hay diferencias. Desde este cuadro de diálogo existen las opciones siguientes:

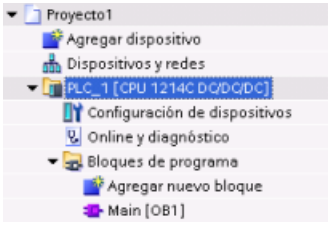
- Comparación online/offline: Al hacer clic en este botón, STEP 7 muestra los bloques de programa, bloques de sistema, objetos tecnológicos, variables PLC y tipos de datos PLC del proyecto en comparación con la CPU online (Página 1213). Se puede hacer clic en cada objeto para ver un análisis detallado de las diferencias, incluidos los sellos de tiempo. Esta información puede utilizarse para decidir lo que debe hacerse con las diferencias entre la CPU online y el proyecto.
- Sincronizar: Al hacer clic en este botón, STEP 7 carga los bloques de programa, variables y otros objetos de la CPU online en el proyecto. Seguidamente puede continuar con la descarga del programa, a menos que la ejecución del programa haya provocado que se pierda de nuevo la sincronización con la CPU.
- Continuar sin sincronización: Al hacer clic en este botón, STEP 7 descarga el proyecto en la CPU.
- Cancelar: Al hacer clic en este botón se cancela la operación de descarga.

7.8 Cargar desde la CPU online

También se pueden copiar bloques de programa de una CPU online o una Memory Card conectada a la programadora.

Prepare el proyecto offline para los bloques de programa copiados:

1. Agregue un dispositivo CPU que coincida con la CPU online.
2. Expanda el nodo de la CPU una vez de manera que se vea la carpeta "Bloques de programa".



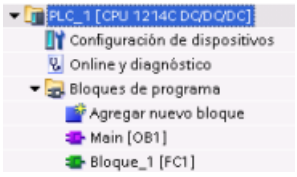
Para cargar los bloques de programa desde la CPU online al proyecto offline, siga estos pasos:





1. Haga clic en la carpeta "Bloques de programa" del proyecto offline.
2. Haga clic en el botón "Establecer conexión online".
3. Haga clic en el botón "Cargar".
4. Confirme la decisión en el cuadro de diálogo Cargar (Página 1201).

Una vez se ha completado la carga, STEP 7 muestra todos los bloques de programa cargados en el proyecto.



7.8.1 Comparar la CPU online con la CPU offline

El editor de comparación (Página 1213) de STEP 7 permite buscar diferencias entre los proyectos online y offline. Esto resulta útil antes de cargar desde la CPU.

7.9 Depurar y comprobar el programa

7.9.1 Vigilar y modificar datos de la CPU

Tal y como muestra la tabla siguiente, los valores de la CPU online pueden vigilarse y modificarse.

Tabla 7-4 Vigilar y modificar datos con STEP 7

Editor	Vigilar	Modificar	Forzado permanente
Tabla de observación	Sí	Sí	No
Tabla de forzado permanente	Sí	No	Sí

Editor	Vigilar	Modificar	Forzado permanente
Editor de programas	Sí	Sí	No
Tabla de variables	Sí	No	No
Editor de DB	Sí	No	No



Vigilar con una tabla de observación



Vigilar con el editor KOP.

Véase el capítulo "Online y diagnóstico" para más información sobre cómo vigilar y modificar datos de la CPU (Página 1215).

7.9.2 Tablas de observación y tablas de forzado

Las "tablas de observación" se utilizan para observar y forzar los valores del programa de usuario que se está ejecutando en la CPU online. Es posible crear y guardar diferentes tablas de observación en el programa para soportar distintos entornos de test. Esto permite reproducir los tests durante la puesta en marcha, o bien para fines de servicio y mantenimiento.

Una tabla de observación permite observar e interactuar con la CPU mientras ésta ejecuta el programa de usuario. Es posible ver o cambiar los valores no sólo de las variables de los bloques lógicos y bloques de datos, sino también de las áreas de memoria de la CPU, incluyendo las entradas y salidas (I y Q), entradas de periferia (I:P), marcas (M) y bloques de datos (DB).

La tabla de observación permite habilitar las salidas físicas (Q:P) de una CPU en estado operativo STOP. Por ejemplo, es posible asignar valores específicos a las salidas al comprobar el cableado de la CPU.

STEP 7 también dispone de una tabla para "forzar permanentemente" una variable a un valor concreto. Para más información sobre el forzado permanente, consulte el apartado Forzado permanente de valores en la CPU (Página 1222) del capítulo "Online y diagnóstico".

Nota

Los valores de forzado permanente se guardan en la CPU y no en la tabla de visualización.

No se puede forzar permanentemente una entrada (o dirección "I"). No obstante, sí que es posible forzar permanentemente una entrada de periferia. Para forzar permanentemente una entrada de periferia, agregue ":P" a la dirección (por ejemplo: "On:P").

STEP 7 también ofrece la capacidad de trazar y registrar las variables de programa en función de las condiciones de disparo (Página 1234).

7.9.3 Referencia cruzada para mostrar la utilización

La ventana de inspección muestra referencias cruzadas sobre cómo un objeto seleccionado se utiliza en todo el proyecto, p. ej. en el programa de usuario, la CPU y los dispositivos HMI. La ficha "Referencias cruzadas" muestra las instancias en las que se está utilizando un objeto seleccionado y los objetos que las utilizan. La ventana de inspección también incluye bloques que solo están disponibles online en las referencias cruzadas. Para mostrar las referencias cruzadas, elija el comando "Mostrar referencias cruzadas". (En la vista del proyecto encontrará las referencias cruzadas en el menú "Herramientas".)

Nota

No es necesario cerrar el editor para ver la información de las referencias cruzadas.

Las entradas de la tabla de referencias cruzadas pueden clasificarse. La lista de referencias cruzadas proporciona una vista general del uso de direcciones de memoria y variables en el programa de usuario.

- Al crear y cambiar un programa se genera una vista general de los operandos, variables y llamadas de bloque utilizados.
- Desde las referencias cruzadas se puede saltar directamente a la ubicación de los operandos y variables.
- Durante un test de programa o el tratamiento de errores se indica qué parte de la memoria se está procesando por qué comando y en qué bloque, qué variable se está utilizando en qué pantalla y qué bloque es llamado por qué otro bloque.

Tabla 7-5 Elementos de la referencia cruzada

Columna	Descripción
Objeto	Nombre del objeto que utiliza los objetos del nivel inferior o que es utilizado por éstos.
Número	Número de utilizaciones
Ubicación (lugar de uso)	Cada una de las ubicaciones, por ejemplo, un segmento
Propiedad	Propiedades especiales de objetos referenciados, por ejemplo, los nombres de variables en declaraciones multiinstancia.
como	Muestra información adicional sobre el objeto, como p. ej. si un DB instancia se utiliza como plantilla o como multiinstancia
Acceso	Tipo de acceso: el acceso al operando puede ser un acceso de lectura (R) y/o de escritura (W)
Dirección	Dirección del operando
Tipo	Información del tipo y el lenguaje utilizados para crear el objeto
Ruta	Ruta del objeto en el árbol del proyecto

Dependiendo de los productos instalados, la tabla de referencias cruzadas mostrará columnas adicionales o diferentes.

7.9.4 Estructura de llamadas para ver la jerarquía de llamadas

La estructura de llamadas describe la jerarquía de llamadas del bloque dentro del programa de usuario. Proporciona una vista general de los bloques utilizados, las llamadas a otros bloques, la relación entre bloques, los datos necesarios para cada bloque y el estado de los bloques. Es posible abrir el editor de programación y editar bloques desde la estructura de llamadas.

Al visualizar la estructura de llamadas se dispone de una lista de los bloques utilizados en el programa de usuario. STEP 7 resalta el primer nivel de la estructura de llamadas y muestra los bloques que no son llamados por ningún otro bloque del programa. El primer nivel de la estructura de llamadas muestra los OBs y todas las FCs, los FBs y DBs que no son llamados por ningún OB. Si un bloque lógico llama otro bloque, el bloque llamado se muestra en forma de sangría debajo del bloque invocante. La estructura de llamadas solo muestra aquellos bloques que son llamados por un bloque lógico.

Dentro de la estructura de llamadas se pueden visualizar selectivamente solo bloques que causan conflictos. Los conflictos se dan en las siguientes condiciones:

- Bloques que ejecutan llamadas con sello de tiempo anterior o posterior
- Bloques que llaman un bloque con interfaz forzada
- Bloques que utilizan una variable con dirección y/o tipo de datos forzado
- Bloques que no son llamados ni directa ni indirectamente por ningún OB
- Bloques que llaman un bloque inexistente o no disponible

Se pueden agrupar varias llamadas de bloque y bloques de datos. Una lista desplegable muestra los enlaces a las diferentes ubicaciones de llamada.

También es posible realizar una comprobación de coherencia para ver conflictos de sello de tiempo. Cambiando el sello de tiempo de un bloque durante o tras la generación del programa se pueden provocar conflictos, lo que, a su vez, provoca incoherencias en los bloques que llaman y que son llamados.

- La mayoría de los conflictos de sello de tiempo y de interfaz pueden corregirse volviendo a compilar los bloques lógicos.
- Si la compilación no depura las incoherencias, utilice el enlace de la columna "Detalles" para ir a la fuente del problema en el editor de programación. De ese modo se pueden eliminar las incoherencias manualmente.
- Los bloques marcados en rojo deben volver a compilarse.

Instrucciones básicas

8

8.1 Operaciones lógicas con bits

8.1.1 Operaciones lógicas con bits

KOP y FUP resultan muy efectivos para procesar lógica booleana. Por otro lado, aunque SCL resulta especialmente efectivo para la computación matemática compleja y para estructuras de control de proyectos, se puede utilizar para la lógica booleana.

Contactos KOP

Tabla 8-1 Contactos normalmente abiertos y normalmente cerrados

KOP	SCL	Descripción

	`IF in THEN`
	`  Statement;`
	`ELSE`
	`  Statement;`
	`END_IF;`
		Contactos normalmente abiertos y normalmente cerrados: Los contactos se pueden conectar a otros contactos, creando así una lógica combinacional propia. Si el bit de entrada indicado utiliza el identificador de memoria I (entrada) o Q (salida), el valor de bit se lee de la memoria imagen de proceso. Las señales de los contactos físicos del proceso controlado se cablean con los bornes de entrada del PLC. La CPU consulta las señales de entrada cableadas y actualiza continuamente los valores de estado correspondientes en la memoria imagen de proceso de las entradas.
		La lectura directa de una entrada física se realiza introduciendo ":P" después del offset I (p. ej. "%I3.4:P"). En una lectura inmediata, los valores de datos de bit se leen directamente de la entrada física y no de la memoria imagen de proceso. La lectura inmediata no actualiza la memoria imagen de proceso.

	`IF NOT (in) THEN`
	`  Statement;`
	`ELSE`
	`  Statement;`
	`END_IF;`	

Tabla 8-2 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Bool	Bit asignado

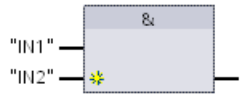
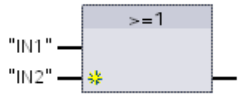
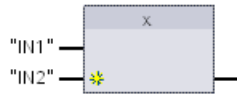
- El contacto normalmente abierto se cierra (ON) cuando el valor de bit asignado es igual a 1.
- El contacto normalmente cerrado se cierra (ON) cuando el valor de bit asignado es igual a 0.
- Los contactos conectados en serie crean segmentos lógicos Y.
- Los contactos conectados en paralelo crean segmentos lógicos O.

Cuadros Y, O y O-exclusiva en FUP

En la programación FUP, los segmentos de los contactos KOP se transforman en segmentos de cuadros Y (&), O (>=1) y O-exclusiva (x), en los que pueden indicarse valores de bit para las entradas y salidas de los cuadros. También es posible interconectar cuadros lógicos y crear combinaciones lógicas propias. Tras colocar un cuadro en el segmento, es posible arrastrar la función "Insertar entrada" desde la barra de herramientas "Favoritos" o desde el árbol de instrucciones y soltarla en el lado de entrada del cuadro para agregar entradas adicionales. También se puede hacer clic con el botón derecho del ratón en el conector de entrada del cuadro y seleccionar "Insertar entrada".

Es posible conectar las entradas y salidas de los cuadros con un cuadro lógico diferente, o bien introducir una dirección de bit o un nombre simbólico de bit para una entrada no conectada. Cuando se ejecuta el cuadro, los estados actuales de las entradas se aplican a la lógica del cuadro binario y, si se cumplen, la salida del cuadro será verdadera.

Tabla 8-3 Cuadros Y, O y O-exclusiva

FUP	SCL[1]	Descripción
	`out := in1 AND in2;`	Todas las entradas de un cuadro Y tienen que cumplirse para que la salida sea TRUE (verdadera).
	`out := in1 OR in2;`	Una entrada cualquiera de un cuadro O tiene que cumplirse para que la salida sea TRUE (verdadera).
	`out := in1 XOR in2;`	Un número impar de entradas de un cuadro O-exclusiva tiene que cumplirse para que la salida sea TRUE (verdadera).

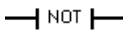
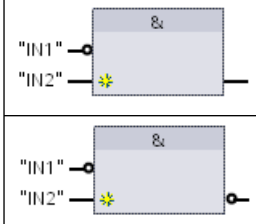
[1] En SCL: el resultado de la operación debe asignarse a una variable para que pueda usarse en otra instrucción.

Tabla 8-4 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN1, IN2	Bool	Bit de entrada

Invertir resultado lógico (NOT)

Tabla 8-5 Invertir RLO (resultado lógico)

KOP	FUP	SCL	Descripción
		**NOT**	En la programación FUP es posible arrastrar la función "Invertir RLO" desde la barra de herramientas "Favoritos" o desde el árbol de instrucciones y soltarla en una entrada o salida para crear un inversor lógico en ese conector del cuadro. El contacto NOT KOP invierte el estado lógico de la entrada de flujo de corriente. • Si no fluye corriente al contacto NOT, hay flujo de corriente en la salida. • Si fluye corriente al contacto NOT, no hay flujo de corriente en la salida.

Bobina de relé y cuadro de asignación

La instrucción "Bobina de salida, relé" escribe un valor en un bit de salida. Si el bit de salida indicado utiliza el identificador de memoria Q, la CPU activa o desactiva el bit de salida en la memoria imagen de proceso, poniendo el bit especificado al correspondiente estado de flujo de corriente. Las señales de salida de los actuadores de control se cablean con los terminales Q de la CPU. En el modo RUN, el sistema de la CPU lee las señales de entrada continuamente, procesa los estados de entrada acorde con la lógica del programa de usuario, y a continuación reacciona aplicando nuevos estados lógicos de salida en la memoria imagen de proceso de las salidas. La CPU transfiere el nuevo estado de las salidas almacenado en la memoria imagen de proceso a los bornes de salida cableados.

Tabla 8-6 Asignación y negar asignación

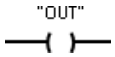
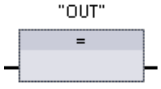
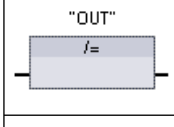
KOP	FUP	SCL	Descripción
		**out := <Expresión booleana>;**	En la programación FUP, las bobinas KOP se transforman en cuadros de asignación (= y /=), en los que se indica una dirección de bit para la salida del cuadro. Es posible conectar las entradas y salidas del cuadro con otros cuadros lógicos, o bien introducir una dirección de bit. La escritura directa en una salida física se indica introduciendo ":P" después del offset Q (p. ej. "%Q3.4:P"). En una escritura inmediata, los valores de datos de bit se escriben en la memoria imagen de proceso de las salidas y directamente en la salida física.
		**out := NOT <Expresión booleana>;**	

Tabla 8-7 Tipos de datos para los parámetros

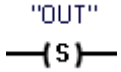
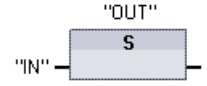
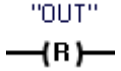
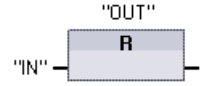
Parámetro	Tipo de datos	Descripción
OUT	Bool	Bit asignado

- Si fluye corriente a través de una bobina de salida o se habilita un cuadro FUP "=", el bit de salida se pone a 1.
- Si no fluye corriente a través de una bobina de salida o no se habilita un cuadro de asignación FUP "=", el bit de salida se pone a 0.
- Si fluye corriente a través de una bobina de salida invertida o se habilita un cuadro FUP "/=", el bit de salida se pone a 0.
- Si no fluye corriente a través de una bobina de salida invertida o no se habilita un cuadro FUP "/=", el bit de salida se pone a 1.

8.1.2 Instrucciones "Activar salida" y "Desactivar salida"

Activar y desactivar 1 bit

Tabla 8-8 Instrucciones S y R

KOP	FUP	SCL	Descripción
		No disponible	Activar salida: Si se activa S (Set) el valor de datos de la dirección OUT se pone a 1. Si S no está activado, OUT no cambia.
		No disponible	Desactivar salida: Si se activa R (Reset), el valor de datos de la dirección de salida OUT se pone a 0. Si no se activa R, no se modifica OUT.

[1] En KOP y FUP: Estas instrucciones pueden disponerse en cualquier posición del segmento.

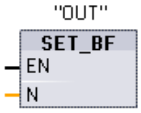
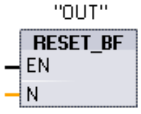
[2] En SCL: Es necesario escribir código para duplicar esta función en la aplicación.

Tabla 8-9 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN (o conectar a contacto/lógica de puerta)	Bool	Dirección de bit que se debe observar
OUT	Bool	Dirección de bit que se debe activar o desactivar

Activar y desactivar mapa de bits

Tabla 8-10 Instrucciones SET_BF y RESET_BF

KOP[1]	FUP	SCL	Descripción
		No disponible	Activar mapa de bits: Cuando se activa SET_BF, el valor de datos 1 se asigna a "n" bits, comenzando en la variable de dirección OUT. Si SET_BF no se activa, OUT no cambia.
		No disponible	Desactivar mapa de bits: RESET_BF escribe un valor de datos 0 en "n" bits, comenzando en la variable de dirección OUT. Si RESET_BF no se activa, OUT no cambia.

1 En KOP y FUP: Estas instrucciones solo se pueden disponer en el extremo derecho de una rama.

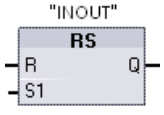
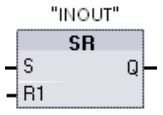
2 En SCL: Es necesario escribir código para duplicar esta función en la aplicación.

Tabla 8-11 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
OUT	Bool	Elemento inicial de un mapa de bits que se debe activar o desactivar (ejemplo: #MyArray[3])
n	Constante (UInt)	Número de bits que deben escribirse

Flipflop de activación/desactivación y flipflop de desactivación/activación

Tabla 8-12 Instrucciones RS y SR

KOP / FUP	SCL	Descripción
	No disponible	Flipflop de desactivación/activación: RS es un flipflop en el que domina la activación. Si las señales de activación (S1) y desactivación (R) son verdaderas, el valor en la dirección INOUT se pone a 1.
	No disponible	Flipflop de activación/desactivación: SR es un flipflop en el que domina la desactivación. Si las señales de activación (S) y desactivación (R1) son verdaderas, el valor en la dirección INOUT se pone a 0.

1 En KOP y FUP: Estas instrucciones solo se pueden disponer en el extremo derecho de una rama.

2 En SCL: Es necesario escribir código para duplicar esta función en la aplicación.

Tabla 8-13 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
S, S1	Bool	Activar entrada; 1 indica dominancia
R, R1	Bool	Desactivar entrada; 1 indica dominancia

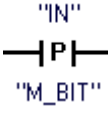
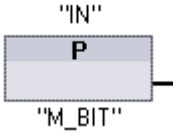
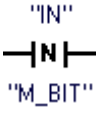
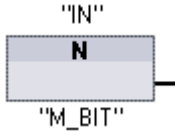
Parámetro	Tipo de datos	Descripción
INOUT	Bool	Variable de bit asignada "INOUT"
Q	Bool	Corresponde al estado del bit "INOUT"

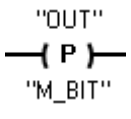
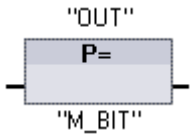
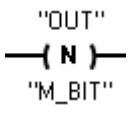
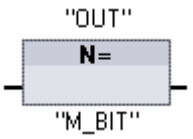
La variable "INOUT" asigna la dirección de bit que se activa o desactiva. La salida opcional Q sigue el estado lógico de la dirección "INOUT".

Instrucción	S1	R	Bit "INOUT"
RS	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	1
	S	**R1**	
SR	0	0	Estado anterior
	0	1	0
	1	0	1
	1	1	0

8.1.3 Instrucciones de flanco ascendente y descendente

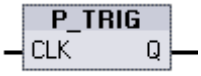
Tabla 8-14 Detección de flanco ascendente y descendente

KOP	FUP	SCL	Descripción
		No disponible [1]	Consultar flanco de señal ascendente de un operando. KOP: El estado de este contacto es TRUE cuando se detecta un flanco ascendente (OFF a ON) en el bit "IN" asignado. El estado lógico del contacto se combina entonces con el estado de entrada del flujo de corriente para activar el estado de salida del flujo de corriente. El contacto P puede disponerse en cualquier posición del segmento, excepto al final de una rama. FUP: El estado lógico de la salida es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el bit de entrada asignado. El cuadro P Solo se puede disponer al comienzo de una rama.
		No disponible [1]	Consultar flanco de señal descendente de un operando. KOP: El estado de este contacto es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el bit de entrada asignado. El estado lógico del contacto se combina entonces con el estado de entrada del flujo de corriente para activar el estado de salida del flujo de corriente. El contacto N puede disponerse en cualquier posición del segmento, excepto al final de una rama. FUP: El estado lógico de la salida es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el bit de entrada asignado. El cuadro N Solo se puede disponer al comienzo de una rama.

KOP	FUP	SCL	Descripción
		No disponible [1]	Activar operando con flanco de señal ascendente. KOP: El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el flujo de corriente que entra a la bobina. El estado de entrada del flujo de corriente atraviesa la bobina como el estado de salida del flujo de corriente. La bobina P puede disponerse en cualquier posición del segmento. FUP: El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco ascendente (OFF a ON) en el estado lógico de la conexión de entrada del cuadro, o bien en la asignación del bit de entrada si el cuadro está ubicado al comienzo de una rama. El estado lógico de la entrada atraviesa el cuadro como el estado lógico de la salida. El cuadro P= puede disponerse en cualquier posición de la rama.
		No disponible [1]	Activar operando con flanco de señal descendente. KOP: El bit asignado "OUT" es TRUE (verdadero) cuando se detecta un flanco descendente (ON a OFF) en el flujo de corriente que entra a la bobina. El estado de entrada del flujo de corriente atraviesa la bobina como el estado de salida del flujo de corriente. La bobina N puede disponerse en cualquier posición del segmento. FUP: El bit asignado "OUT" es TRUE cuando se detecta un flanco descendente (ON a OFF) en el estado lógico de la conexión de entrada del cuadro, o bien en la asignación del bit de entrada si el cuadro está ubicado al comienzo de una rama. El estado lógico de la entrada atraviesa el cuadro como el estado lógico de la salida. El cuadro N= puede disponerse en cualquier posición de la rama.

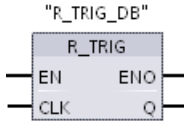
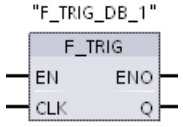
[1] En SCL: Es necesario escribir código para duplicar esta función en la aplicación.

Tabla 8-15 P_TRIG y N_TRIG

KOP / FUP	SCL	Descripción
	No disponible [1]	Consultar flanco de señal ascendente del RLO (resultado lógico). El flujo de corriente o estado lógico de la salida Q es TRUE cuando se detecta un flanco ascendente (OFF a ON) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción P_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción P_TRIG puede disponerse en cualquier posición de la rama, excepto al final.
	No disponible [1]	Consultar flanco de señal descendente del RLO. El flujo de corriente o estado lógico de la salida Q es TRUE cuando se detecta un flanco descendente (ON a OFF) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción N_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción N_TRIG puede disponerse en cualquier posición de la rama, excepto al final.

[1] En SCL: Es necesario escribir código para duplicar esta función en la aplicación.

Tabla 8-16 Instrucciones R_TRIG y F_TRIG

KOP / FUP	SCL	Descripción
	**"R_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);**	Activar variable con flanco de señal ascendente. El DB de instancia asignado se usa para almacenar el estado anterior de la entrada CLK. El flujo de corriente o estado lógico de la salida Q es TRUE cuando se detecta un flanco ascendente (OFF a ON) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción R_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción R_TRIG puede disponerse en cualquier posición de la rama, excepto al final.
	"F_TRIG_DB"(CLK:=_in_, Q=> _bool_out_);	Activar variable con flanco de señal descendente. El DB de instancia asignado se usa para almacenar el estado anterior de la entrada CLK. El flujo de corriente o estado lógico de la salida Q es TRUE cuando se detecta un flanco descendente (ON a OFF) en el estado lógico de CLK (en FUP) o en el flujo de corriente de CLK (en KOP). En KOP, la instrucción F_TRIG no se puede disponer ni al comienzo ni al final de un segmento. En FUP, la instrucción F_TRIG puede disponerse en cualquier posición de la rama, excepto al final.

Para R_TRIG y F_TRIG, cuando inserte la instrucción en el programa, se abre automáticamente el cuadro de diálogo "Opciones de llamada". En este cuadro de diálogo puede asignar si la marca de flanco se almacena en su propio bloque de datos (instancia única) o como una variable local (multiinstancia) en la interfaz de bloque. Si crea un bloque de datos independiente, lo encontrará en el árbol del proyecto de la carpeta "Recursos de programa" en "Bloques de programa > Bloques de sistema".

Tabla 8-17 Tipos de datos para los parámetros (contactos/bobinas P y N, P=, N=, P_TRIG and N_TRIG)

Parámetro	Tipo de datos	Descripción
M_BIT	Bool	Marca en la que se almacena el estado anterior de la entrada
IN	Bool	Bit de entrada cuyo flanco se detecta
OUT	Bool	Bit de salida que indica que se ha detectado un flanco
CLK	Bool	Flujo de corriente o bit de entrada cuyo flanco debe detectarse
Q	Bool	Salida que indica que se ha detectado un flanco

Todas las instrucciones usan un bit de marca (M_BIT: contactos/bobinas P/N, P_TRIG/N_TRIG) o (bit de DB de instancia: R_TRIG, F_TRIG) para almacenar el estado anterior de la señal de entrada observada. Un flanco se detecta comparando el estado de la entrada con el estado anterior. Si

los estados indican un cambio de la entrada en el sentido deseado, se notifica un flanco activando la salida (TRUE). De lo contrario, se desactivará la salida (FALSE).

Nota

Las instrucciones de detección de flancos evalúan los valores de la entrada y de la marca cada vez que se ejecutan, incluyendo la primera ejecución. Los estados iniciales de la entrada y de la marca deben considerarse al diseñar el programa, con objeto de permitir o impedir la detección de flancos en el primer ciclo.

Puesto que la marca debe conservarse desde una ejecución hasta la siguiente, es preciso utilizar un bit unívoco para cada instrucción de detección de flancos. Este bit no se puede utilizar en ninguna otra ubicación del programa. También se debe evitar la memoria temporal y la memoria que pueda ser modificada por otras funciones de sistema, p. ej. una actualización de E/S. Utilice Solo el área de marcas (M), DB global o memoria estática (en un DB de instancia) para las asignaciones de memoria de M_BIT.

8.2 Temporizadores

Las instrucciones con temporizadores se utilizan para crear retardos programados. El número de temporizadores que pueden utilizarse en el programa de usuario está limitado solo por la cantidad de memoria disponible en la CPU. Cada temporizador utiliza una estructura de DB del tipo de datos IEC_Timer de 16 bytes para guardar la información del temporizador especificada encima de la instrucción de cuadro o bobina. STEP 7 crea el DB automáticamente al insertar la instrucción.

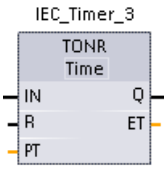
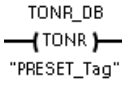
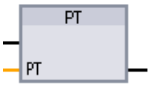
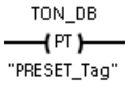
Tabla 8-18 Instrucciones con temporizadores

Cuadros KOP / FUP	Bobinas KOP	SCL	Descripción

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_);`
			El temporizador TP genera un impulso con una duración predeterminada.

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_);`
			El temporizador TON pone la salida Q a ON tras un tiempo de retardo predeterminado.

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_);`
			El temporizador TOF pone la salida Q a OFF tras un tiempo de retardo predeterminado.

Cuadros KOP / FUP	Bobinas KOP	SCL	Descripción
		`"IEC_Timer_0_DB".TONR ( IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);`	El temporizador TONR pone la salida Q a ON tras un tiempo de retardo predeterminado. El tiempo transcurrido se acumula a lo largo de varios periodos de temporización hasta que la entrada R inicializa el tiempo transcurrido.
Solo FUP: 		`PRESET_TIMER( PT:=_time_in_, TIMER:=_iec_timer_in_);`	La bobina PT (Cargar temporizador) carga un nuevo valor de tiempo PRESET (predeterminado) en el IEC_Timer especificado.
Solo FUP: 		`RESET_TIMER( _iec_timer_in_);`	La bobina RT (Inicializar temporizador) inicializa el IEC_Timer especificado.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En los ejemplos SCL, "IEC_Timer_0_DB" es el nombre del DB de instancia.

Tabla 8-19 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Cuadro: IN Bobina: Flujo de corriente	Bool	TP, TON, y TONR: Cuadro: 0=deshabilitar temporizador, 1=habilitar temporizador Bobina: Sin flujo de corriente=deshabilitar temporizador, flujo de corriente=habilitar temporizador TOF: Cuadro: 0=habilitar temporizador, 1=deshabilitar temporizador Bobina: Sin flujo de corriente=habilitar temporizador, flujo de corriente=deshabilitar temporizador
R	Bool	Solo cuadro TONR: 0=Sin inicialización 1= Inicializar el tiempo transcurrido y el bit Q a 0
Cuadro: PT Bobina: "PRESET_variable"	Time	Cuadro o bobina de temporizador: Entrada de tiempo predeterminado
Cuadro: Q Bobina: DBdata.Q	Bool	Cuadro de temporizador: salida de cuadro Q o bit Q en los datos del DB de temporizador Bobina de temporizador: solo se puede direccionar el bit Q en los datos del DB de temporizador
Cuadro: ET Bobina: DBdata.ET	Time	Cuadro de temporizador: salida de cuadro ET (tiempo transcurrido) o valor de tiempo ET en los datos del DB de temporizador Bobina de temporizador: solo se puede direccionar el valor de tiempo ET en los datos del DB de temporizador.

Tabla 8-20 Efecto de los cambios de valores en los parámetros PT e IN

Temporizador	Cambios en los parámetros de cuadro PT e IN y en los parámetros de bobina correspondientes
TP	• Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Un cambio de IN no tiene efecto alguno durante el funcionamiento del temporizador.
TON	• Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Si IN cambia a FALSE durante el funcionamiento del temporizador, este se inicializará y se detendrá.
TOF	• Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador. • Si IN cambia a TRUE durante el funcionamiento del temporizador, este se inicializará y se detendrá.
TONR	• Un cambio de PT no tiene efecto alguno durante el funcionamiento del temporizador, pero sí cuando reanuda el contaje. • Si IN cambia a FALSE durante el funcionamiento del temporizador, este se detendrá pero no se inicializará. Si IN vuelve a cambiar a TRUE, el temporizador comenzará a contar desde el valor de tiempo acumulado.

Los valores PT (tiempo predeterminado) y ET (tiempo transcurrido) se almacenan en los datos de DB IEC_TIMER como enteros dobles con signo que representan milisegundos. Los datos TIME utilizan el identificador T# y pueden introducirse como unidad de tiempo simple ("T#200ms ó 200) y como unidades de tiempo compuestas "T#2s_200ms".

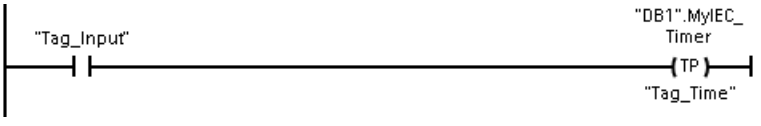
Tabla 8-21 Tamaño y rango del tipo de datos TIME

Tipo de datos	Tamaño	Rangos válidos[1]
TIME	32 bits, almacenados como datos DInt	T#-24d_20h_31m_23s_648ms hasta T#24d_20h_31m_23s_647ms Almacenado como -2.147.483.648 ms hasta +2.147.483.647 ms

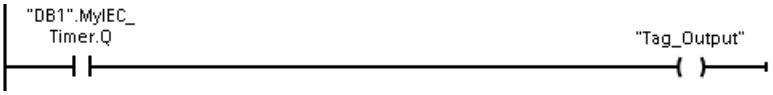
[1] El rango negativo del tipo de datos TIME indicado arriba no puede utilizarse con las instrucciones de temporizador. Los valores PT (tiempo predeterminado) negativos se ponen a cero cuando se ejecuta la instrucción de temporización. ET (tiempo transcurrido) es siempre un valor positivo.

Ejemplo de bobina de temporizador

Las bobinas de temporizador -(TP)-, -(TON)-, -(TOF)- y -(TONR)- deben ser la última instrucción de un segmento KOP. Como se indica en el ejemplo de temporizador, una instrucción de contacto de un segmento posterior evalúa el bit Q en los datos de DB IEC_Timer de una bobina de temporizador. Del mismo modo, hay que direccionar el elemento ELAPSED en los datos de DB IEC_timer si se desea emplear el valor de tiempo transcurrido en el programa.



El temporizador como impulso arranca en una transición de 0 a 1 del valor de bit Tag_Input. El temporizador se ejecuta durante el tiempo especificado por el valor de tiempo Tag_Time.



Mientras el temporizador esté en marcha, el estado de DB1.MyIEC_Timer.Q=1 y el valor de Tag_Output=1. Cuando haya transcurrido el valor de Tag_Time, DB1.MyIEC_Timer.Q=0 y el valor de Tag_Output=0.

Bobinas Inicializar temporizador -(RT)- y predeterminar temporizador -(PT)-

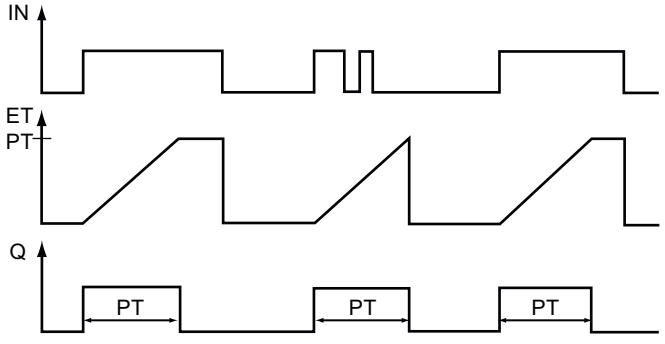
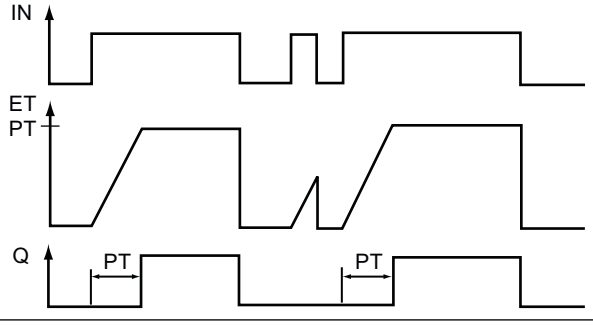
Estas instrucciones de bobina pueden utilizarse con temporizadores de cuadro o bobina y pueden colocarse en una posición central. El estado del flujo de salida de la bobina siempre es el mismo que el estado de entrada de la bobina. Cuando se activa la bobina -(RT)-, el elemento de tiempo transcurrido de los datos de DB IEC_Timer especificados se pone a 0. Cuando la bobina -(PT)- se activa, el elemento de tiempo predeterminado de los datos de DB IEC_Timer especificados se carga con el valor de duración de tiempo asignado.

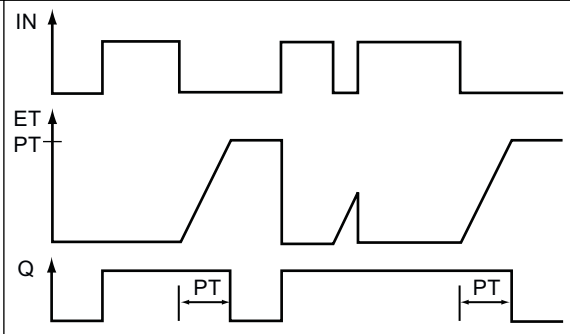
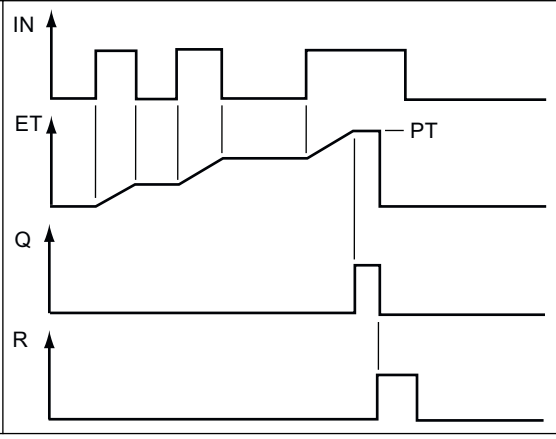
Nota

Si se insertan instrucciones de temporizador en un FB, se puede seleccionar la opción "Bloque de datos multiinstancia". Los nombres de estructura de temporizador pueden ser diferentes con diferentes estructuras de datos, pero los datos del temporizador están contenidos como un bloque de datos único y no requiere un bloque de datos separado para cada temporizador. Esto reduce el tiempo de procesamiento y la memoria de datos necesaria para gestionar los temporizadores. No hay interacción entre las estructuras de datos de los temporizadores en el DB multiinstancia compartido.

Manejo de temporizadores

Tabla 8-22 Tipos de temporizadores IEC

Temporizador	Cronograma
TP: Generar impulso El temporizador TP genera un impulso con una duración predeterminada.	
TON: Retardo al conectar El temporizador TON pone la salida Q a ON tras un tiempo de retardo predeterminado.	

Temporizador	Cronograma
TOF: Retardo al desconectar El temporizador TOF pone la salida Q a OFF tras un tiempo de retardo predeterminado.	
TONR: Acumulador de tiempo El temporizador TONR pone la salida Q a ON tras un tiempo de retardo predeterminado. El tiempo transcurrido se acumula a lo largo de varios periodos de temporización hasta que la entrada R inicializa el tiempo transcurrido.	

Nota

En la CPU no se asigna ningún recurso dedicado a ninguna instrucción de temporizador específica. En lugar de eso, cada temporizador utiliza su estructura de temporizador propia en la memoria de DB y un temporizador de funcionamiento continuo interno de la CPU para la temporización.

Cuando se arranca un temporizador debido a un flanco en la entrada de una instrucción TP, TON, TOF o TONR, el valor del temporizador de funcionamiento continuo interno de la CPU se copia en el miembro START de la estructura del DB asignado a esta instrucción de temporizador. Este valor de arranque permanece inalterado mientras el temporizador continúa funcionando, y se vuelve a utilizar cada vez que el temporizador se actualiza. Cada vez que el temporizador arranca se carga un nuevo valor en la estructura de temporizador del temporizador interno de la CPU.

Cuando se actualiza un temporizador, el valor de arranque arriba descrito se resta del valor actual del temporizador interno de la CPU para determinar el tiempo transcurrido. El tiempo transcurrido se compara entonces con el predeterminado para determinar el estado del bit Q del temporizador. Los miembros ELAPSED y Q se actualizan en la estructura de DB asignada a este temporizador. Recuerde que el tiempo transcurrido se detiene en el valor predeterminado (el temporizador no continúa acumulando tiempo transcurrido una vez que se ha alcanzado el valor predeterminado).

Una actualización de temporizador se realiza única y exclusivamente cuando:

- Se ejecuta una instrucción de temporizador (TP, TON, TOF o TONR)
- El miembro "ELAPSED" de la estructura del DB se referencia directamente con una instrucción
- El miembro "Q" de la estructura de temporizador del DB se referencia directamente con una instrucción

Programación de temporizadores

A la hora de planificar y crear el programa de usuario deben considerarse las siguientes consecuencias del manejo de temporizadores:

- Pueden producirse múltiples actualizaciones de un temporizador en el mismo ciclo. El temporizador se actualiza cada vez que la instrucción de temporizador (TP, TON, TOF, TONR) se ejecuta y cada vez que el miembro ELAPSED o Q de la estructura de temporizador se utiliza como parámetro para otra instrucción ejecutada. Esto es una ventaja si se desea tener el último dato de tiempo (en esencia, una lectura inmediata del temporizador). No obstante, si desea disponer de valores coherentes durante un ciclo del programa, inserte la instrucción de temporizador antes de todas las instrucciones restantes que requieran estos valores, y utilice las variables de las salidas Q y ET de la instrucción de temporizador en lugar de los miembros ELAPSED y Q de la estructura de DB de temporizador.
- Pueden producirse ciclos durante los cuales no se actualice ningún temporizador. Es posible arrancar el temporizador en una función y dejar de llamar la función durante uno o más ciclos. Si no se ejecuta ninguna otra instrucción que referencie los miembros ELAPSED o Q de la estructura de temporizador, el temporizador no se actualizará. No se produce una nueva actualización hasta que la instrucción de temporizador se ejecuta de nuevo o se ejecuta alguna otra instrucción utilizando ELAPSED o Q de la estructura del temporizador como un parámetro.

- Aunque no es común, se puede asignar la misma estructura de DB de temporizador a varias instrucciones de temporizador. En general, para evitar una interacción inesperada, debería utilizarse solo una instrucción de temporizador (TP, TON, TOF, TONR) por estructura de temporizador de DB.
- Las inicializaciones propias de los temporizadores resultan útiles para lanzar acciones que deben producirse periódicamente. Generalmente, los temporizadores autoinicializables se crean colocando un contacto normalmente cerrado que referencie el bit de temporizador situado frente a la instrucción de temporizador. Este segmento de temporizador se suele colocar sobre uno o más segmentos dependientes, que utilizan el bit del temporizador para lanzar acciones. Cuando el temporizador finaliza (el tiempo transcurrido llega al valor predeterminado), el bit de temporizador permanece en ON durante un ciclo, permitiendo que se ejecute la lógica de segmento dependiente controlada por el bit de temporizador. Con la siguiente ejecución del segmento de temporizador, el contacto normalmente cerrado está en desconexión, con lo que el temporizador se reinicia y el bit de temporizador se desactiva. En el siguiente ciclo, el contacto normalmente cerrado está en conexión, de modo que el contacto se inicializa. Cuando cree temporizadores autoinicializables de esta manera, no utilice el miembro "Q" de la estructura de DB de temporizador, ya que el parámetro para el contacto normalmente cerrado está delante de la instrucción de temporizador. En lugar de ello, utilice la variable asociada a la salida "Q" de la instrucción de temporizador. La razón de evitar el acceso al miembro Q de la estructura de DB de temporizador es que ello provoca una actualización del temporizador, y si este se actualiza debido al contacto normalmente cerrado, este inicializará la instrucción de temporizador inmediatamente. La salida Q de la instrucción de temporizador no estará ON para el ciclo en cuestión, y los segmentos dependientes no se ejecutarán.

Retención de los datos de tiempo tras una transición RUN-STOP-RUN o una desconexión y nueva conexión de la CPU

Si una sesión en modo RUN finaliza con el modo STOP o una desconexión y nueva conexión de la CPU y se inicia una nueva sesión en modo RUN, los datos de temporizador guardados en la sesión anterior se pierden, a no ser que la estructura de datos se haya definido como remanente (temporizadores TP, TON, TOF y TONR).

Si se aceptan los ajustes predeterminados del diálogo de opciones de llamada una vez insertada la instrucción en el editor de programas, automáticamente se asignará un DB de instancia **que no puede definirse como remanente**. Para que los datos de temporizador puedan ser remanentes, hay que usar un DB global o un DB multiinstancia.

Asignar un DB global para guardar los datos de temporizador como datos remanentes

Esta opción funciona independientemente de dónde se encuentre el temporizador (OB, FC o FB).

1. Crear un DB global:
   - Haga doble clic en "Agregar nuevo bloque" en el árbol del proyecto
   - Haga clic en el icono del bloque de datos (DB)
   - Elija DB global como tipo
   - Si desea poder definir elementos de datos individuales en este DB como remanentes, asegúrese de que está activada la casilla "Optimizado". La otra opción para el tipo de DB "Estándar - compatible con S7-300/400" Solo permite definir todos los elementos del DB como remanentes o no remanentes.
   - Haga clic en "Aceptar".
2. Agregar estructura(s) de temporizador al DB:
   - En el nuevo DB global, agregue una nueva variable estática utilizando el tipo de datos IEC_Timer.
   - En la columna "Remanencia" compruebe la casilla correspondiente para que la estructura sea remanente.
   - Repita este procedimiento para crear estructuras para todos los temporizadores que desee guardar en ese DB. Es posible insertar cada estructura de temporizador en un DB global único, o insertar múltiples estructuras de temporizador en el mismo DB global. También es posible asignar al DB global otras variables estáticas aparte de los temporizadores. La colocación de múltiples estructuras de temporizador en el mismo DB global permite reducir el número general de bloques.
   - Cambie el nombre de las estructuras de temporizador si lo desea.
3. Abra el bloque de programa para editar la posición en la que desea colocar el temporizador remanente (OB, FC o FB).
4. Sitúe la instrucción de temporizador en la posición deseada.
5. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón Cancelar.
6. En la parte superior de la nueva instrucción de temporizador escriba el nombre (no utilice la función de ayuda para examinar) del DB global y de la estructura de temporizador que ha creado antes (ejemplo: "Bloque_datos_3.Estático_1").

Asignar un DB multiinstancia para guardar los datos de temporizador como datos remanentes

Esta opción solo funciona si el temporizador se coloca en un FB

Esta opción depende de si las propiedades del FB especifican "Acceso optimizado al bloque" (solo permite el direccionamiento simbólico). Para comprobar cómo está configurado el atributo de acceso para un FB existente, haga clic con el botón derecho sobre el FB en el árbol del proyecto, seleccione Propiedades y, a continuación, Atributos.

Si el FB especifica "Acceso optimizado al bloque" (solo permite el direccionamiento simbólico):

1. Abra el FB para editarlo.
2. Sitúe la instrucción de temporizador en la posición deseada dentro del FB.

3. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón de multiinstancia. La opción de multiinstancia solo está disponible si la instrucción se está colocando en un FB.
4. En el diálogo de opciones de llamada, cambie el nombre del temporizador si lo desea.
5. Haga clic en "Aceptar". La instrucción de temporizador aparece en el editor, y la estructura IEC_TIMER aparece en la interfaz de FB, bajo Estático.
6. En caso necesario, abra el editor de interfaz de FB (puede ser necesario hacer clic en la flecha pequeña para expandir la vista).
7. En Estático, localice la estructura de temporizador que se acaba de crear.
8. En la columna Remanencia correspondiente a dicha estructura, cambie la selección a "Remanencia". Cada vez que este FB se llame posteriormente desde otro bloque de programa se creará un DB de instancia con esta definición de interfaz, que contiene la estructura de temporizador marcada como remanente.

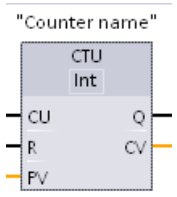
Si el FB no especifica "Acceso optimizado al bloque", el tipo de acceso al bloque es estándar, que es compatible con configuraciones clásicas de S7-300/400 y permite el direccionamiento simbólico y directo. Para asignar una multiinstancia en un FB de acceso estándar al bloque, proceda del siguiente modo:

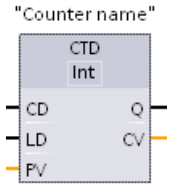
1. Abra el FB para editarlo.
2. Sitúe la instrucción de temporizador en la posición deseada dentro del FB.
3. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón de multiinstancia. La opción de multiinstancia solo está disponible si la instrucción se está colocando en un FB.
4. En el diálogo de opciones de llamada, cambie el nombre del temporizador si lo desea.
5. Haga clic en "Aceptar". La instrucción de temporizador aparece en el editor, y la estructura IEC_TIMER aparece en la interfaz de FB, bajo Estático.
6. Abra el bloque que usará este FB.
7. Sitúe el FB en la posición deseada. De este modo se crea un bloque de datos de instancia para este FB.
8. Abra el bloque de datos de instancia creado al colocar el FB en el editor.
9. En Estático, coloque la estructura de temporizador en cuestión. En la columna Remanencia correspondiente a dicha estructura, cambie la casilla de verificación de modo que la estructura sea remanente.

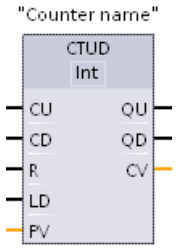
8.3 Contadores

Tabla 8-23 Instrucciones con contadores

KOP / FUP	SCL	Descripción

	`"IEC_Counter_0_DB".CTU(`
	`CU:=_bool_in,`
	`R:=_bool_in,`
	`PV:=_in,`
	`Q=>_bool_out,`
	`CV=>_out);`
		Las instrucciones con contadores se utilizan para contar eventos del programa internos y eventos del proceso externos. Todo contador utiliza una estructura almacenada en un bloque de datos para conservar sus datos. El bloque de datos se asigna al colocar la instrucción de contaje en el editor.
		• CTU es un contador ascendente
		• CTD es un contador descendente
		• CTUD es un contador ascendente/descendente

	`"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] En KOP y FUP: Seleccione el tipo de datos del valor de contaje en la lista desplegable situada debajo del nombre de la instrucción.

[2] STEP 7 crea el DB automáticamente al insertar la instrucción.

[3] En los ejemplos SCL, "IEC_Counter_0_DB" es el nombre del DB de instancia.

Tabla 8-24 Tipos de datos para los parámetros

Parámetro	Tipo de datos[1]	Descripción
CU, CD	Bool	Contaje ascendente o descendente, en incrementos de uno
R (CTU, CTUD)	Bool	Poner a cero el valor del contador
LD (CTD, CTUD)	Bool	Control de carga del valor predeterminado
PV	SInt, Int, DInt, USInt, UInt, UDInt	Valor de contaje predeterminado
Q, QU	Bool	Es verdadero si CV >= PV
QD	Bool	Es verdadero si CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDInt	Valor de contaje actual

[1] El rango numérico de valores de contaje depende del tipo de datos seleccionado. Si el valor de contaje es un entero sin signo, es posible contar hacia atrás hasta cero o hacia delante hasta el límite del rango. Si el valor de contaje es un entero con signo, es posible contar hacia atrás hasta el límite de entero negativo y contar hacia delante hasta el límite de entero positivo.

El número de contadores que pueden utilizarse en el programa de usuario está limitado Solo por la cantidad de memoria disponible en la CPU. Los contadores utilizan la siguiente cantidad de memoria:

- En los tipos de datos SInt o USInt, la instrucción de contaje utiliza 3 bytes.
- En los tipos de datos nt o UInt, la instrucción de contaje utiliza 6 bytes.
- En los tipos de datos DInt o UDInt, la instrucción de contaje utiliza 12 bytes.

Estas instrucciones utilizan contadores por software cuya frecuencia de contaje máxima está limitada por la frecuencia de ejecución del OB en el que están contenidas. El OB en el que se depositan las instrucciones debe ejecutarse con suficiente frecuencia para detectar todas las transiciones de las entradas CU o CD. Para operaciones de contaje rápido, consulte la instrucción CTRL_HSC (Página 539).

Nota

Al colocar instrucciones con contadores en un FB es posible seleccionar la opción de DB multiinstancia. Los nombres de estructura de los contadores pueden diferir en las distintas estructuras, pero los datos de los contadores se encuentran en un DB individual y no requieren un DB propio para cada contador. Esto reduce el tiempo de procesamiento y la memoria de datos necesaria para los contadores. No hay interacción entre las estructuras de datos de los contadores en el DB multiinstancia compartido.

Manejo de los contadores

Tabla 8-25 Operación de CTU (contador ascendente)

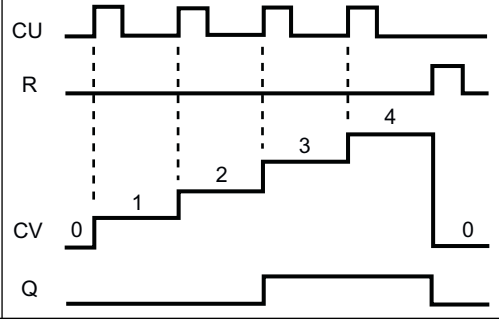
Contador	Operación
El contador CTU incrementa en 1 cuando el valor del parámetro CU cambia de 0 a 1. El cronograma de CTU muestra el manejo con un valor de contaje de entero sin signo (donde PV = 3). • Si el valor del parámetro CV (valor de contaje actual) es superior o igual que el del parámetro PV (valor de contaje predeterminado), el parámetro de salida del contador Q = 1. • Si el valor del parámetro de desactivación R cambia de 0 a 1, el valor de contaje actual se pone a 0.	

Tabla 8-26 Operación de CTD (contador descendente)

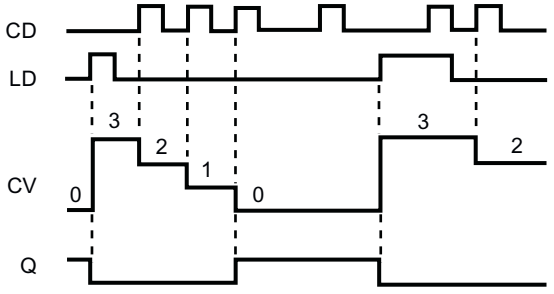
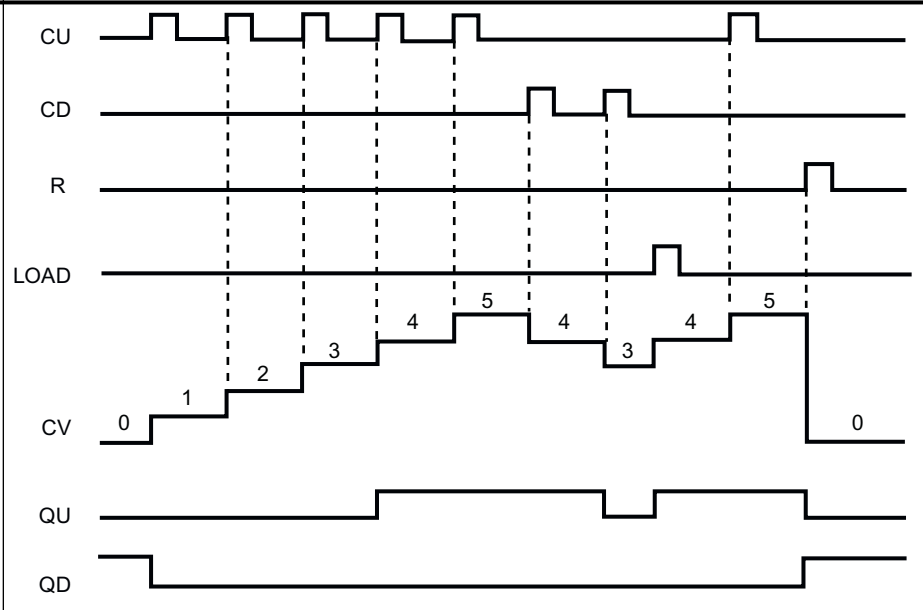
Contador	Operación
El contador CTD decrementa en 1 cuando el valor del parámetro CD cambia de 0 a 1. El cronograma de CTD muestra el manejo con un valor de contaje de entero sin signo (donde PV = 3). • Si el valor del parámetro CV (valor de contaje actual) es inferior o igual a 0, el parámetro de salida del contador Q = 1. • Si el valor del parámetro LOAD cambia de 0 a 1, el valor del parámetro PV (valor predeterminado) se carga en el contador como nuevo CV (valor de contaje actual).	

Tabla 8-27 Operación de CTUD (contador ascendente - descendente)

Contador	Operación
El contador CTUD incrementa o decrementa en 1 en una transición de 0 a 1 de las entradas de contaje ascendente (CU) o descendente (CD). El cronograma muestra el funcionamiento de un contador CTUD con un valor de contaje de entero sin signo (donde PV = 4). • Si el valor del parámetro CV es superior o igual que el del parámetro PV, el parámetro de salida del contador QU = 1. • Si el valor del parámetro CV es inferior o igual a 0, el parámetro de salida del contador QD = 1. • Si el valor del parámetro LOAD cambia de 0 a 1, el valor del parámetro PV se carga en el contador como nuevo CV. • Si el valor del parámetro de reset R cambia de 0 a 1, el valor de contaje actual se pone a 0.	

Remanencia de los datos de contador tras una transición RUN-STOP-RUN o una desconexión y nueva conexión de la CPU

Si una sesión en modo RUN finaliza con el modo STOP o una desconexión y nueva conexión de la CPU y se inicia una nueva sesión en modo RUN, los datos de contador guardados en la sesión anterior se pierden, a no ser que la estructura de datos se haya definido como remanente (contadores CTU, CTD y CTUD).

Si se aceptan los ajustes predeterminados del diálogo de opciones de llamada una vez insertada la instrucción de contador en el editor de programas, automáticamente se asignará un DB de instancia **que no puede definirse como remanente**. Para que los datos de contador puedan ser remanentes, hay que usar un DB global o un DB multiinstancia.

Asignar un DB global para guardar los datos de contador como datos remanentes

Esta opción funciona independientemente de dónde se encuentre el contador (OB, FC o FB).

1. Crear un DB global:
   - Haga doble clic en "Agregar nuevo bloque" en el árbol del proyecto
   - Haga clic en el icono del bloque de datos (DB)
   - Elija DB global como tipo
   - Si desea poder definir elementos individuales en este DB como remanentes, asegúrese de que está activada la casilla que limita el acceso a solo simbólico.
   - Haga clic en "Aceptar".
2. Agregar estructura(s) de contador al DB:
   - En el nuevo DB global, agregue una nueva variable estática utilizando uno de los tipos de datos de contador. Asegúrese de elegir el tipo que desea utilizar para los valores predeterminado y de contaje.
   - En la columna "Remanencia" compruebe la casilla correspondiente para que la estructura sea remanente.
   - Repita este procedimiento para crear estructuras para todos los contadores que desee guardar en ese DB. Es posible insertar cada estructura de contador en un DB global único, o insertar múltiples estructuras de contador en el mismo DB global. También es posible asignar al DB global otras variables estáticas aparte de los contadores. La colocación de múltiples estructuras de contador en el mismo DB global permite reducir el número general de bloques.
   - Cambie el nombre de las estructuras de temporizador si lo desea.
3. Abra el bloque de programa para editar la posición en la que desea colocar el contador remanente (OB, FC o FB).
4. Sitúe la instrucción de contador en la posición deseada.
5. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón Cancelar. Ahora debería ver una nueva instrucción de contador con "???" justo arriba y abajo del nombre de la instrucción.
6. En la parte superior de la nueva instrucción de contador escriba el nombre (no utilice la función de ayuda para examinar) del DB global y de la estructura de contador que ha creado antes (ejemplo: "Bloque_datos_3.Estático_1"). Con esa acción se rellenan los tipos de valores predeterminado y de contaje correspondientes (ejemplo: UInt para una estructura IEC_UCounter).

Tipo de datos de contador	Tipo correspondiente para los valores predeterminado y de contaje
IEC_Counter	INT

IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

Asignar un DB multiinstancia para guardar los datos de contador como datos remanentes

Esta opción solo funciona si el contador se coloca en un FB

Esta opción depende de si las propiedades del FB especifican "Acceso optimizado al bloque" (solo permite el direccionamiento simbólico). Para comprobar cómo está configurado el atributo de acceso para un FB existente, haga clic con el botón derecho sobre el FB en el árbol del proyecto, seleccione Propiedades y, a continuación, Atributos.

Si el FB especifica "Acceso optimizado al bloque" (solo permite el direccionamiento simbólico):

1. Abra el FB para editarlo.
2. Sitúe la instrucción de contador en la posición deseada dentro del FB.
3. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón de multiinstancia. La opción de multiinstancia solo está disponible si la instrucción se está colocando en un FB.
4. En el diálogo de opciones de llamada, cambie el nombre del contador si lo desea.
5. Haga clic en "Aceptar". La instrucción de contador aparece en el editor con el tipo INT para los valores predeterminado y de contaje, y la estructura IEC_COUNTER aparece en la interfaz de FB, bajo Estático.
6. Si lo desea, cambie el tipo de en la instrucción de contador de INT a uno de los tipos restantes. La estructura de contador cambiará de la forma correspondiente.
7. En caso necesario, abra el editor de interfaz de FB (puede ser necesario hacer clic en la flecha pequeña para expandir la vista).
8. En Estático, localice la estructura de contador que se acaba de crear.
9. En la columna Remanencia correspondiente a dicha estructura, cambie la selección a "Remanencia". Cada vez que este FB se llame posteriormente desde otro bloque de programa se creará un DB de instancia con esta definición de interfaz con la estructura de contador marcada como remanente.

Si el FB no especifica "Acceso optimizado al bloque", el tipo de acceso al bloque es estándar, que es compatible con configuraciones clásicas de S7-300/400 y permite el direccionamiento simbólico y directo. Para asignar una multiinstancia en un FB de acceso estándar al bloque, proceda del siguiente modo:

1. Abra el FB para editarlo.
2. Sitúe la instrucción de contador en la posición deseada dentro del FB.
3. Cuando aparezca el diálogo de opciones de llamada, haga clic en el botón de multiinstancia. La opción de multiinstancia solo está disponible si la instrucción se está colocando en un FB.
4. En el diálogo de opciones de llamada, cambie el nombre del contador si lo desea.

5. Haga clic en "Aceptar". La instrucción de contador aparece en el editor con el tipo INT para los valores predeterminado y de contaje, y la estructura IEC_COUNTER aparece en la interfaz de FB, bajo Estático.
6. Si lo desea, cambie el tipo de en la instrucción de contador de INT a uno de los tipos restantes. La estructura de contador cambiará de la forma correspondiente.
7. Abra el bloque que usará este FB.
8. Sitúe el FB en la posición deseada. De este modo se crea un bloque de datos de instancia para este FB.
9. Abra el bloque de datos de instancia creado al colocar el FB en el editor.
10. En Estático, localice la estructura de contador en cuestión. En la columna Remanencia correspondiente a dicha estructura, cambie la casilla de verificación de modo que la estructura sea remanente.

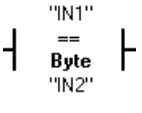
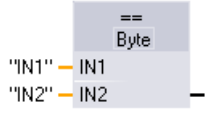
Tipo indicado en la instrucción de contador (para valores predeterminado y de contaje)	Tipo de estructura correspondiente indicado en la interfaz de FB
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

8.4 Comparación

8.4.1 Instrucciones de comparación

Tabla 8-28 Instrucciones de comparación

KOP	FUP	SCL	Descripción

		`out := in1 = in2;`
		or
		`IF in1 = in2`
		`  THEN out := 1;`
		`  ELSE out := 0;`
		`  END_IF;`
			Compara varios elementos del mismo tipo de datos. Si la comparación de contactos KOP es TRUE (verdadera), se activa el contacto. Si la comparación de cuadros FUP es TRUE (verdadera), la salida del cuadro es TRUE.

[1] En KOP y FUP: haga clic en el nombre de la instrucción (p. ej. "==") para cambiar el tipo de comparación en la lista desplegable. Haga clic en "???" y seleccione un tipo de datos en la lista desplegable.

Tabla 8-29 Tipos de datos para los parámetros

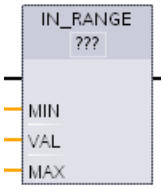
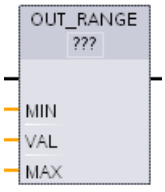
Parámetro	Tipo de datos	Descripción
IN1, IN2	Byte, Word, DWord, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, WString, Char, Char, Time, Date, TOD, DTL, Constante	Valores que deben compararse

Tabla 8-30 Descripciones de comparaciones

Tipo de relación	La comparación se cumple si ...
=	IN1 es igual a IN2
<>	IN1 es diferente de IN2
>=	IN1 es mayor o igual a IN2
<=	IN1 es menor o igual a IN2
>	IN1 es mayor que IN2
<	IN1 es menor que IN2

8.4.2 IN_Range (valor dentro del rango) y OUT_Range (valor fuera del rango)

Tabla 8-31 Instrucciones de valor dentro del rango y valor fuera del rango

KOP / FUP	SCL	Descripción
	`out := IN_RANGE(min, val, max);`	Comprueba si un valor de entrada está dentro o fuera de un rango de valores específico. Si la comparación es TRUE (verdadera), la salida del cuadro es TRUE.
	`out := OUT_RANGE(min, val, max);`	

[1] En KOP y FUP: haga clic en "???" y seleccione el tipo de datos en la lista desplegable.

Tabla 8-32 Tipos de datos para los parámetros

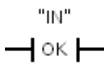
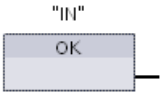
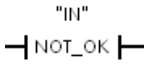
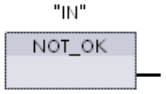
Parámetro	Tipo de datos[1]	Descripción
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Constante	Entradas de comparación

[1] Los parámetros de entrada MIN, VAL y MAX deben tener un mismo tipo de datos.

- La comparación IN_RANGE se cumple si: MIN <= VAL <= MAX
- La comparación OUT_RANGE se cumple si: VAL < MIN o VAL > MAX

8.4.3 OK (comprobar validez) y NOT_OK (comprobar invalidez)

Tabla 8-33 Instrucciones OK (comprobar validez) y NOT_OK (comprobar invalidez)

KOP	FUP	SCL	Descripción
		No disponible	Comprueba si una referencia de datos de entrada es un número real válido según la especificación IEEE 754.
		No disponible	

[1] En KOP y FUP: Si el contacto KOP es TRUE, se activa este contacto y conduce corriente. Si el cuadro FUP es TRUE (verdadero), la salida del cuadro es TRUE.

Tabla 8-34 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Datos de entrada

Tabla 8-35 Operación

Instrucción	La comprobación del número Real es TRUE (verdadera) si:
OK	El valor de entrada es un número real válido [1]
NOT_OK	El valor de entrada no es un número real válido [1]

[1] Un valor Real o LReal no es válido si es +/- INF (infinito), NaN (no es un número) o si es un valor desnormalizado. Un valor desnormalizado es un número muy próximo a cero. La CPU sustituye un valor desnormalizado por cero en los cálculos.

8.4.4 Instrucciones de comparación Variant y Array

8.4.4.1 Instrucciones de comparación de igualdad y desigualdad

La CPU S7-1200 ofrece instrucciones para consultar el tipo de datos de una variable a la que apunta un operando Variant para saber si el tipo de datos del otro operando es igual o no.

Asimismo, la CPU S7-1200 ofrece instrucciones para consultar el tipo de datos de un elemento Array para saber si el tipo de datos del otro operando es igual o no.

En estas instrucciones se compara <Operand1> con <Operand2>. <Operand1> debe tener el tipo de datos Variant. <Operand2> puede ser un tipo de datos elemental de un tipo de datos PLC. En KOP y FUP, <Operand1> es el operando anterior a la instrucción. En KOP, <Operand2> es el operando posterior a la instrucción.

Para todas las instrucciones, el resultado de las operaciones lógicas (RLO) es 1 (verdadero) si pasa la prueba de igualdad o desigualdad y es 0 (falso) si no la pasa.

Las instrucciones de comparación de tipo igualdad y desigualdad son las siguientes:

- EQ_Type (comparar si el tipo de datos es IGUAL al tipo de datos de una variable)
- NE_Type (comparar si el tipo de datos es DIFERENTE del tipo de datos de una variable)
- EQ_ElemType (comparar si el tipo de datos de un elemento del ARRAY es IGUAL al tipo de datos de una variable)
- NE_ElemType (comparar si el tipo de datos de un elemento del ARRAY es DIFERENTE al tipo de datos de una variable)

Tabla 8-36 Instrucciones EQ y NE

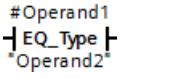
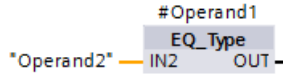
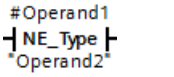
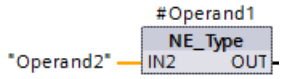
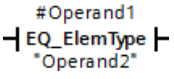
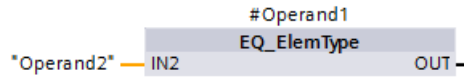
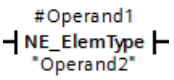
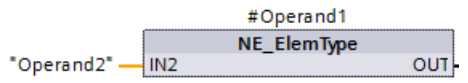
KOP	FUP	SCL	Descripción
		No disponible	Prueba si la variable apuntada por Variant en el Operand1 es del mismo tipo de datos que la variable del Operand2.
		No disponible	Prueba si la variable apuntada por Variant en el Operand1 es de un tipo de datos diferente que la variable del Operand2.
		No disponible	Prueba si el elemento Array apuntado por Variant en el Operand1 es del mismo tipo de datos que la variable del Operand2.
		No disponible	Prueba si el elemento Array apuntado por Variant en el Operand1 es de un tipo de datos diferente que la variable del Operand2.

Tabla 8-37 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Operand1	Variant	Primer operando
Operand2	Cadenas de bits, enteros, números en coma flotante, temporizadores, fecha y hora, cadenas de caracteres, ARRAY, tipos de datos PLC	Segundo operando

8.4.4.2 Instrucciones de comparación con cero

Las instrucciones IS_NULL y NOT_NULL se utilizan para determinar si la entrada apunta a un objeto o no.

Para las dos instrucciones, <Operand> debe tener el tipo de datos Variant.

Tabla 8-38 Instrucciones IS_NULL (consultar si es un puntero IGUAL A CERO) NOT_NULL (consultar si es un puntero DIFERENTE DE CERO)

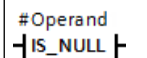
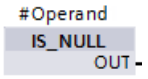
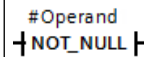
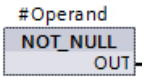
KOP	FUP	SCL	Descripción
		No disponible	Comprueba si la variable apuntada por Variant en Operand es cero y, por tanto, no es un objeto.
		No disponible	Comprueba si la variable apuntada por Variant en Operand no es cero y, por tanto, apunta a un objeto.

Tabla 8-39 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Operand	Variant	Operando para evaluar cero o no cero.

8.4.4.3 IS_ARRAY (Consultar si es un ARRAY)

Puede emplear la instrucción "Consultar si es un ARRAY" para consultar si Variant apunta a una variable del tipo de datos Array.

<Operand> debe tener el tipo de datos Variant.

Las instrucciones devuelven 1 (verdadero) si el operando es un Array.

Tabla 8-40 IS_ARRAY (Consultar si es un ARRAY)

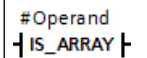
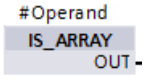
KOP	FUP	SCL	Descripción
		IS_ARRAY(_variant_in_)	Comprueba si la variable apuntada por Variant en Operand es un Array.

Tabla 8-41 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Operand	Variant	Operando para evaluar si es un Array.

8.5 Funciones matemáticas

8.5.1 CALCULATE (calcular)

Tabla 8-42 Instrucción CALCULATE

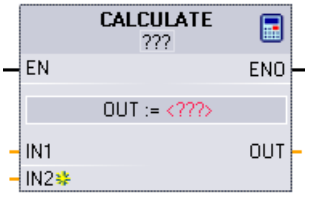
KOP / FUP	SCL	Descripción
	Utilice las expresiones matemáticas SCL estándar para crear la ecuación.	La instrucción CALCULATE permite crear una función matemática que funciona con entradas (IN1, IN2, ... INn) y genera el resultado en OUT, según la ecuación definida. • En primer lugar, seleccione un tipo de datos. Todas las entradas y la salida deben tener un mismo tipo de datos. • Para agregar otra entrada, haga clic en el icono de la última entrada.

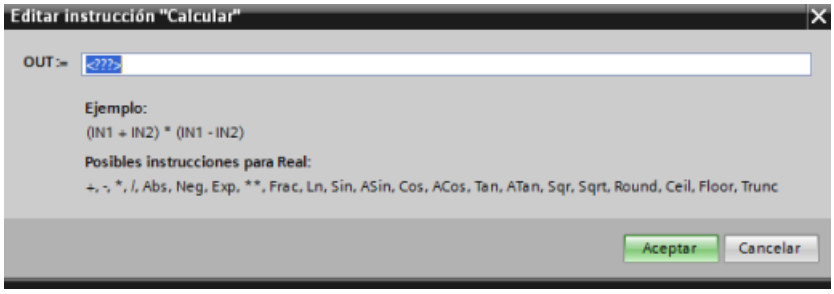
Tabla 8-43 Tipos de datos de los parámetros

Parámetro	Tipo de datos[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] Los parámetros IN y OUT deben tener un mismo tipo de datos (con conversiones implícitas de los parámetros de entrada). Ejemplo: un valor SINT para una entrada se convertiría a un valor INT o REAL si OUT fuera un INT o REAL.

Haga clic en el icono de la calculadora para abrir el cuadro de diálogo y definir la función matemática. La ecuación se introduce como entradas (p. ej. IN1 y IN2) y operaciones. Cuando se hace clic en "Aceptar" para guardar la función, el cuadro de diálogo crea automáticamente las entradas de la instrucción CALCULATE.

El cuadro de diálogo muestra un ejemplo y una lista de posibles instrucciones que pueden incluirse en función del tipo de datos del parámetro OUT:



Nota

También es necesario crear una entrada para las constantes que pudiera haber en la función. En este caso, el valor constante se introduciría en la entrada asociada de la instrucción CALCULATE.

Si se introducen constantes como entradas, es posible copiar la instrucción CALCULATE a otras ubicaciones del programa de usuario sin tener que cambiar la función. Entonces, pueden modificarse los valores o las variables de las entradas de la instrucción sin modificar la función.

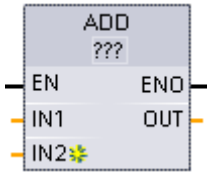
Cuando se ejecuta CALCULATE y todas las operaciones individuales del cálculo se realizan de forma correcta, entonces ENO = 1. Si no es así, ENO = 0.

Para un ejemplo de la instrucción CALCULATE, véase "AUTOHOTSPOT".

8.5.2 Instrucciones "Sumar", "Restar", "Multiplicar" y "Dividir"

Tabla 8-44 Instrucciones "Sumar", "Restar", "Multiplicar" y "Dividir"

KOP / FUP	SCL	Descripción

	`out := in1 + in2;`
	`out := in1 - in2;`
	`out := in1 * in2;`
	`out := in1 / in2;`
		• ADD: Sumar (IN1 + IN2 = OUT)
		• SUB: Restar (IN1 - IN2 = OUT)
		• MUL: Multiplicar (IN1 * IN2 = OUT)
		• DIV: Dividir (IN1 / IN2 = OUT)
		Una operación de división de enteros trunca la parte fraccionaria del cociente y produce un valor de salida entero.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-45 Tipos de datos para los parámetros (KOP y FUP)

Parámetro	Tipo de datos[1]	Descripción
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, constante	Entradas de la operación matemática
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Salida de la operación matemática

[1] Los parámetros IN1, IN2 y OUT deben ser del mismo tipo de datos.



Para agregar una entrada ADD o MUL, haga clic en el icono "Crear" o haga clic con el botón derecho del ratón en el conector de entrada del parámetro IN existente y seleccione el comando "Insertar entrada".

Para quitar una entrada, haga clic con el botón derecho del ratón en el conector de entrada de uno de los parámetros IN existentes (si hay más entradas además de las dos originales) y seleccione el comando "Borrar".

Si está habilitada (EN = 1), la instrucción matemática realiza la operación indicada en los valores de entrada (IN1 e IN2) y almacena el resultado en la dirección de memoria que indica el

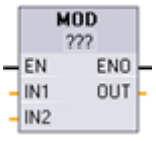
parámetro de salida (OUT). Una vez finalizada correctamente la operación, la instrucción pone ENO a 1.

Tabla 8-46 Estado de ENO

ENO	Descripción
1	No hay error
0	El resultado de la operación matemática quedaría fuera del rango numérico válido del tipo de datos seleccionado. Se devuelve la parte menos significativa del resultado que quepa en el tamaño de destino.
0	División por 0 (IN2 = 0): El resultado es indefinido y se devuelve cero.
0	Real/LReal: Si uno de los valores de entrada es NaN (no es un número), se devuelve NaN.
0	ADD Real/LReal: Si ambos valores IN son INF con signos diferentes, la operación no está permitida y se devuelve NaN.
0	SUB Real/LReal: Si ambos valores IN son INF con signos iguales, la operación no está permitida y se devuelve NaN.
0	MUL Real/LReal: Si un valor IN es cero y el otro es INF, la operación no está permitida y se devuelve NaN.
0	DIV Real/LReal: Si ambos valores IN son cero o INF, la operación no está permitida y se devuelve NaN.

8.5.3 MOD (obtener resto de división)

Tabla 8-47 Instrucción MOD (obtener resto de división)

KOP / FUP	SCL	Descripción
	`out := in1 MOD in2;`	La instrucción MOD se puede utilizar para obtener el resto de una operación de división de enteros. El valor de la entrada IN1 se divide por el valor de la entrada IN2 y el producto se deposita en la salida OUT.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-48 Tipos de datos para parámetros

Parámetro	Tipo de datos[1]	Descripción
IN1 e IN2	SInt, Int, DInt, USInt, UInt, UDInt, constante	Entradas modulo
OUT	SInt, Int, DInt, USInt, UInt, UDInt	Salida modulo

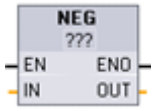
[1] Los parámetros IN1 y IN2 y OUT deben ser del mismo tipo de datos.

Tabla 8-49 Valores ENO

ENO	Descripción
1	No hay error
0	Valor IN2 = 0, a OUT se le asigna el valor cero

8.5.4 NEG (generar complemento a dos)

Tabla 8-50 Instrucción NEG (generar complemento a dos)

KOP / FUP	SCL	Descripción
	`-(in);`	La instrucción NEG invierte el signo aritmético del valor del parámetro IN y deposita el resultado en el parámetro OUT.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-51 Tipos de datos para parámetros

Parámetro	Tipo de datos[1]	Descripción
IN	SInt, Int, DInt, Real, LReal, constante	Entrada de la operación matemática
OUT	SInt, Int, DInt, Real, LReal	Salida de la operación matemática

[1] Los parámetros IN y OUT deben ser del mismo tipo de datos.

Tabla 8-52 Estado de ENO

ENO	Descripción
1	No hay error
0	El resultado está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: NEG (-128) arroja el resultado +128 que excede el límite máximo del tipo de datos.

8.5.5 INC (incrementar) y DEC (decrementar)

Tabla 8-53 Instrucciones INC y DEC

KOP / FUP	SCL	Descripción
	`in_out := in_out + 1;`	Incrementa un valor de número entero con o sin signo: Valor IN_OUT +1 = valor IN_OUT
	`in_out := in_out - 1;`	Decrementa un valor de número entero con o sin signo: Valor IN_OUT - 1 = valor IN_OUT

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-54 Tipos de datos para parámetros

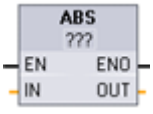
Parámetro	Tipo de datos	Descripción
IN/OUT	SInt, Int, DInt, USInt, UInt, UDInt	Entrada/salida de la operación matemática

Tabla 8-55 Estado de ENO

ENO	Descripción
1	Sin error
0	El resultado está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: INC (+127) da como resultado +128, es decir, excede el máximo de tipos de datos.

8.5.6 ABS (calcular valor absoluto)

Tabla 8-56 Instrucción ABS (calcular valor absoluto)

KOP / FUP	SCL	Descripción
	`out := ABS(in);`	Calcula el valor absoluto de un entero con signo o número real indicado en el parámetro IN y deposita el resultado en el parámetro OUT.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-57 Tipos de datos para parámetros

Parámetro	Tipo de datos[1]	Descripción
IN	SInt, Int, DInt, Real, LReal	Entrada de la operación matemática
OUT	SInt, Int, DInt, Real, LReal	Salida de la operación matemática

[1] Los parámetros IN y OUT deben ser del mismo tipo de datos.

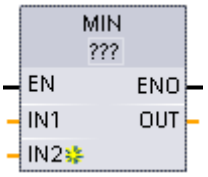
Tabla 8-58 Estado de ENO

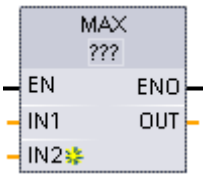
ENO	Descripción
1	No hay error
0	El resultado de la operación aritmética está fuera del rango numérico válido del tipo de datos seleccionado. Ejemplo de SInt: ABS (-128) da como resultado +128, es decir, excede el máximo de tipos de datos.

8.5.7 MIN (determinar mínimo) y MAX (determinar máximo)

Tabla 8-59 Instrucciones MIN (determinar mínimo) y MAX (determinar máximo)

KOP / FUP	SCL	Descripción

	`out:= MIN(`
	`in1:=_variant_in_,`
	`in2:=_variant_in_`
	`[,...in32]);`
		La instrucción MIN compara el valor de dos parámetros IN1 y IN2 y asigna el valor mínimo (menor) al parámetro OUT.

	`out:= MAX(`
	`in1:=_variant_in_,`
	`in2:=_variant_in_`
	`[,...in32]);`
		La instrucción MAX compara el valor de dos parámetros IN1 y IN2 y asigna el valor máximo (mayor) al parámetro OUT.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-60 Tipos de datos para los parámetros

Parámetro	Tipo de datos[1]	Descripción
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD, constante	Entradas de la operación matemática (hasta 32 entadas)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Salida de la operación matemática

[1] Los parámetros IN1, IN2 y OUT deben tener un mismo tipo de datos.



Para agregar una entrada, haga clic en el icono "Crear" o haga clic con el botón derecho del ratón en el conector de entrada del parámetro IN existente y seleccione el comando "Insertar entrada".

Para quitar una entrada, haga clic con el botón derecho del ratón en el conector de entrada de uno de los parámetros IN existentes (si hay más entradas además de las dos originales) y seleccione el comando "Borrar".

Tabla 8-61 Estado de ENO

ENO	Descripción
1	No hay error
0	Solo para el tipo de datos Real: • Al menos una entrada no es un número real (NaN). • La salida OUT resultante es +/- INF (infinito).

8.5.8 LIMIT (ajustar valor límite)

Tabla 8-62 Instrucción LIMIT (ajustar valor límite)

KOP / FUP	SCL	Descripción

	`LIMIT(MN:=_variant_in_,`
	`IN:=_variant_in_,`
	`MX:=_variant_in_,`
	`OUT:=_variant_out_);`
		La instrucción Limit comprueba si el valor del parámetro IN se encuentra dentro del rango de valores especificado por los parámetros MIN y MAX and if not, clamps the value at MIN or MAX.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-63 Tipos de datos para los parámetros

Parámetro	Tipo de datos[1]	Descripción
MN, IN y MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD· constante	Entradas de la operación matemática
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Salida de la operación matemática

[1] Los parámetros MN, IN, MX y OUT deben ser del mismo tipo de datos.

Si el valor del parámetro IN está dentro del rango indicado, el valor de IN se deposita en el parámetro OUT. Si el valor del parámetro IN está fuera del rango indicado, el valor OUT es entonces el valor del parámetro MIN (si el valor IN es menor que el valor MIN) o del parámetro MAX (si el valor IN es mayor que el valor MAX).

Tabla 8-64 Estado de ENO

ENO	Descripción
1	No hay error
0	Real: Si uno o varios de los valores de MIN, IN y MAX es NaN (no es un número), se devuelve NaN.
0	Si MIN es mayor que MAX, el valor IN se asigna a OUT.

Ejemplos de SCL:

- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Resultado: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Resultado: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Resultado: MyVal = 10

8.5.9 Instrucciones para exponentes, logaritmos y trigonometría

Las instrucciones en coma flotante sirven para programar operaciones matemáticas utilizando los tipos de datos Real o LReal:

- SQR: Calcular cuadrado ($IN^2 = OUT$)
- SQRT: Calcular raíz cuadrada ($\sqrt{IN} = OUT$)
- LN: Calcular logaritmo natural (LN(IN) = OUT)

- EXP: Calcular valor exponencial (e ^{IN}=OUT), donde la base e = 2,71828182845904523536
- EXPT: elevar a potencia (IN1 ^{IN2} = OUT)
  Los parámetros IN1 y OUT de EXPT tienen siempre el mismo tipo de datos, que debe ser Real o LReal. Para el parámetro del exponente IN2 se puede elegir entre varios tipos de datos.
- FRAC: Determinar decimales (parte fraccionaria del número en coma flotante IN = OUT)
- SIN: Calcular valor de seno (seno(IN radianes) = OUT)
- ASIN: Calcular arcoseno (arcoseno(IN) = OUT radianes), donde seno(OUT radianes) = IN
- COS: Calcular coseno (coseno(IN radianes) = OUT)
- ACOS: Calcular arcocoseno (arcocoseno(IN) = OUT radianes), donde coseno(OUT radianes) = IN
- TAN: Calcular tangente (tangente(IN radianes) = OUT)
- ATAN: Calcular arcotangente (arcotangente(IN) = OUT radianes), donde tangente(OUT radianes) = IN

Tabla 8-65 Ejemplos de instrucciones matemáticas en coma flotante

KOP / FUP	SCL	Descripción

	`out := SQR(in);`
	o
	`out := in * in;`
		Cuadrado: IN ^{2} = OUT
		Ejemplo: Si IN = 9, OUT = 81.

	`out := in1 ** in2;`	Exponencial general: IN1 ^{IN2} = OUT Ejemplo: Si IN1 = 3 y IN2 = 2, OUT = 9.

[1] En KOP y FUP: Haga clic en "???" (junto al nombre de la instrucción) y seleccione un tipo de datos en el menú desplegable.

[2] En SCL: También es posible utilizar los operadores matemáticos básicos de SCL para crear expresiones matemáticas.

Tabla 8-66 Tipos de datos para parámetros

Parámetro	Tipo de datos	Descripción
IN, IN1	Real, LReal, constante	Entradas
IN2	SInt, Int, DInt, USInt, UInt,UDInt, Real, LReal, constante	Entrada exponencial EXPT
OUT	Real, LReal	Salidas

Tabla 8-67 Estado de ENO

ENO	Instrucción	Condición	Resultado (OUT)
1	Todas	No hay error	Resultado válido
0	SQR	El resultado excede el rango Real/LReal válido	+INF
		IN es +/- NaN (no es un número)	+NaN
	SQRT	IN es negativo	-NaN
		IN es +/- INF (infinito) o +/- NaN	+/- INF o +/- NaN
	LN	IN es 0,0, negativo, -INF o -NaN	-NaN
		IN es +INF o +NaN	+INF o +NaN
	EXP	El resultado excede el rango Real/LReal válido	+INF
		IN es +/- NaN	+/- NaN
	SIN, COS, TAN	IN es +/- INF o +/- NaN	+/- INF o +/- NaN
	ASIN, ACOS	IN está fuera del rango válido de -1,0 a +1,0	+NaN
		IN es +/- NaN	+/- NaN
	ATAN	IN es +/- NaN	+/- NaN
	FRAC	IN es +/- INF o +/- NaN	+NaN
	EXPT	IN1 es +INF e IN2 no es -INF	+INF
		IN1 es negativo o -INF	+NaN si IN2 es Real/LReal, -INF en caso contrario
		IN1 o IN2 es +/- NaN	+NaN
		IN1 es 0,0 e IN2 es Real/LReal (solo)	+NaN

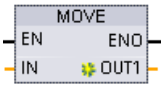
8.6 Transferencia

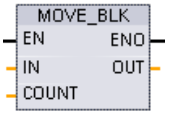
8.6.1 MOVE (copiar valor), MOVE_BLK (copiar área), UMOVE_BLK (copiar área sin interrupciones) y MOVE_BLK_VARIANT (copiar área)

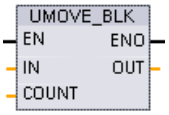
Las instrucciones de desplazamiento permiten copiar elementos de datos en otra dirección de memoria y convertir un tipo de datos en otro. El proceso de desplazamiento no modifica los datos de origen.

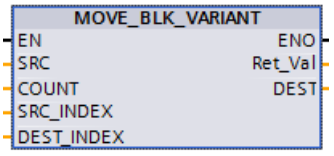
- La instrucción MOVE copia un elemento de datos individual de la dirección de origen que indica el parámetro IN en la dirección de destino que indica el parámetro OUT.
- Las instrucciones MOVE_BLK y UMOVE_BLK tienen un parámetro COUNT adicional. COUNT especifica cuántos elementos de datos se copian. El número de bytes por elemento copiado depende del tipo de datos asignado a los nombres de variables de los parámetros IN y OUT en la tabla de variables PLC.

Tabla 8-68 Instrucciones MOVE, MOVE_BLK, UMOVE_BLK y MOVE_BLK_VARIANT

KOP / FUP	SCL	Descripción
	`out1 := in;`	Copia un elemento de datos almacenado en una dirección indicada en una o varias direcciones diferentes. [1]

	`MOVE_BLK(`
	`  in:=_variant_in,`
	`  count:=_uint_in,`
	`  out=>_variant_out);`
		Desplazamiento con interrupciones que copia un bloque de elementos de datos en otra dirección.

	`UMOVE_BLK(`
	`  in:=_variant_in,`
	`  count:=_uint_in,`
	`  out=>_variant_out);`
		Desplazamiento sin interrupciones que copia un bloque de elementos de datos en otra dirección.

	`MOVE_BLK(`
	`  SRC:=_variant_in,`
	`  COUNT:=_udint_in,`
	`  SRC_INDEX:=_dint_in,`
	`  DEST_INDEX:=_dint_in,`
	`  DEST=>_variant_out);`
		Mueve el contenido de una área de memoria de origen a un área de memoria de destino.
		Puede copiar un Array completo o elementos de un Array en otro Array del mismo tipo de datos. El tamaño (número de elementos) del Array de origen y destino puede variar. Existe la posibilidad de copiar varios elementos o uno solo dentro de un Array. Los tipos de datos Variant pueden utilizarse para apuntar tanto a los Arrays de origen como de destino.

[1] Instrucción MOVE: para agregar otra salida en KOP o FUP, haga clic en el icono "Crear" situado junto al parámetro de salida. Para SCL, utilice múltiples instrucciones de asignación. También se puede usar una de las construcciones de bucle.

Tabla 8-69 Tipos de datos para la instrucción MOVE

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC tipos de datos, tipos de datos PLC	Dirección de origen
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC tipos de datos, tipos de datos PLC	Dirección de destino



Para agregar salidas MOVE, haga clic en el icono "Crear" o haga clic con el botón derecho en el conector de salida de uno de los parámetros OUT existentes y seleccione el comando "Insertar salida".

Para eliminar una salida, haga clic con el botón derecho del ratón en el conector de salida de uno de los parámetros OUT existentes (si hay más salidas además de las dos originales) y seleccione el comando "Borrar".

Tabla 8-70 Tipos de datos para las instrucciones MOVE_BLK y UMOVE_BLK

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal Byte, Word, DWord, Time, Date, TOD, WChar	Dirección de origen inicial
COUNT	UInt	Número de elementos de datos que deben copiarse
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Dirección de destino inicial

Tabla 8-71 Tipos de datos para la instrucción MOVE_BLK_VARIANT

Parámetro	Tipo de datos	Descripción
SRC	Variant (que apunta a un Array o a un elemento Array individual)	Bloque de origen desde el que se copia
COUNT	UDInt	Número de elementos de datos que deben copiarse
SRC_INDEX	DInt	Índice basado en cero en el Array SRC
DEST_INDEX	DInt	Índice basado en cero en el Array DEST
RET_VAL	Int	Información de error
DEST	Variant (que apunta a un Array o a un elemento Array individual)	Área de destino en la que se copia el contenido del bloque de origen

Nota

Reglas para las operaciones de copia de datos

- Para copiar el tipo de datos Bool, utilice las instrucciones SET_BF, RESET_BF, R, S o bobina de salida (KOP) (Página 212)
- Para copiar un solo tipo de datos elemental, utilice MOVE
- Para copiar un Array de un tipo de datos elemental, utilice MOVE_BLK o UMOVE_BLK
- Para copiar una estructura, utilice MOVE
- Para copiar una cadena, utilice S_MOVE (Página 335)
- Para copiar un solo carácter en una cadena, utilice MOVE
- Las instrucciones MOVE_BLK y UMOVE_BLK no pueden utilizarse para copiar Arrays o estructuras en las áreas de memoria I, Q o M.

Las instrucciones MOVE_BLK y UMOVE_BLK se diferencian en la forma de procesar las alarmas:

- Los eventos de alarma **se ponen en cola de espera y se procesan** durante la ejecución de MOVE_BLK. Utilice la instrucción MOVE_BLK si los datos contenidos en la dirección de destino del desplazamiento no se utilizan en un OB de alarma. Si se utilizan, los datos de destino no tienen que ser coherentes. Si se interrumpe una instrucción MOVE_BLK, el último elemento de datos desplazado estará completo y será coherente en la dirección de destino. La instrucción MOVE_BLK se reanuda una vez finalizada la ejecución del OB de alarma.
- Los eventos de alarma **se ponen en cola de espera pero no se procesarán** hasta que no finalice la ejecución de UMOVE_BLK. Utilice la instrucción UMOVE_BLK si la operación de desplazamiento debe finalizarse y los datos de destino deben ser coherentes antes de la ejecución de un OB de alarma. Encontrará más información en el apartado Coherencia de datos (Página 187).

ENO siempre es verdadero tras ejecutarse la instrucción MOVE.

Tabla 8-72 Estado de ENO

ENO	Condición	Resultado
1	No hay error	Todos los elementos de COUNT se han copiado correctamente.
0	El rango de origen (IN) o destino (OUT) excede el área de memoria disponible.	Se copian los elementos que quepan. No se copian elementos parciales.

Tabla 8-73 Códigos de condición para la instrucción MOVE_BLK_VARIANT

RET_VAL (W#16#...)	Descripción
0000	No hay error
80B4	Los tipos de datos no concuerdan.
8151	No es posible acceder al parámetro SRC.
8152	El operando del parámetro SRC es un tipo no válido.
8153	Error de generación del código en el parámetro SRC
8154	El operando del parámetro SRC tiene el tipo de datos Bool.
8281	El parámetro COUNT tiene un valor no válido.
8382	El valor del parámetro SRC_INDEX está fuera de los límites de Variant.
8383	El valor del parámetro SRC_INDEX está fuera del límite superior del Array.
8482	El valor del parámetro DEST_INDEX está fuera de los límites de Variant.
8483	El valor del parámetro DEST_INDEX está fuera del límite superior del Array.
8534	El parámetro DEST está protegido contra escritura.
8551	No es posible acceder al parámetro DEST.
8552	El operando del parámetro DEST es un tipo no válido.
8553	Error de generación del código en el parámetro DEST
8554	El operando del parámetro DEST tiene el tipo de datos Bool.
* Es posible visualizar códigos de error en el editor de programas como valores enteros o hexadecimales.	

8.6.2 Deserializar

La instrucción "Deserializar" se utiliza para volver a convertir la representación secuencial de un tipo de datos PLC (UDT) en un tipo de datos PLC y rellenar todo su contenido. Si la comparación es TRUE la salida del cuadro es TRUE.

El área de memoria que tiene la representación secuencial de un tipo de datos PLC debe tener el tipo de datos Array of Byte y debe declararse que el bloque de datos tiene un acceso estándar (no optimizado). Asegúrese de que hay suficiente espacio de memoria antes de ejecutar la conversión.

La instrucción permite volver a convertir varias representaciones secuenciales de tipos de datos PLC convertidos en sus tipos de datos originales.

Nota

Si solo desea volver a convertir una representación secuencial individual de un tipo de datos PLC (UDT) también puede emplear la instrucción "TRCV: recibir datos a través de la conexión".

Tabla 8-74 Instrucción DESERIALIZE

KOP / FUP	SCL	Descripción

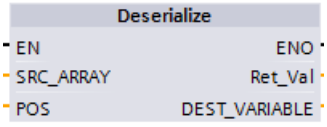
	`ret_val := Deserialize(`
	`SRC_ARRAY:=_variant_in_,`
	`DEST_VARIABLE=>_variant_out_`
	`,`
	`POS:=_dint_inout_);`
		Vuelve a convertir la representación secuencial de un tipo de datos PLC (UDT) en un tipo de datos PLC y rellena todo su contenido

Tabla 8-75 Parámetros de la instrucción DESERIALIZE

Parámetro	Tipo	Tipo de datos	Descripción
SRC_ARRAY	IN	Variant	Bloque de datos global que contiene el flujo de datos
DEST_VARIABLE	INOUT	Variant	Variable en la que se guarda el tipo de datos PLC convertido (UDT)
POS	INOUT	DInt	Número de bytes que usa el tipo de datos PLC convertido
RET_VAL	OUT	Int	Información de error

Tabla 8-76 Parámetro RET_VAL

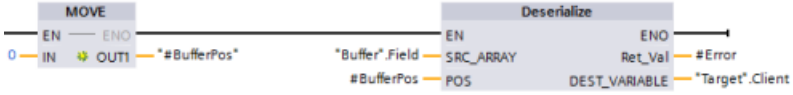
RET_VAL* (W#16#...)	Descripción
0000	No hay error
80B0	Las áreas de memoria para los parámetros SRC_ARRAY y DEST_VARIABLE se solapan.
8136	El bloque de datos del parámetro DEST_VARIABLE no es un bloque con acceso estándar.
8150	El tipo de datos Variant en el parámetro SRC_ARRAY no contiene valores.

RET_VAL* (W#16#...)	Descripción
8151	Error de generación del código en el parámetro SRC_ARRAY.
8153	No hay suficiente memoria libre en el parámetro SRC_ARRAY.
8250	El tipo de datos Variant en el parámetro DEST_VARIABLE no contiene valores.
8251	Error de generación del código en el parámetro DEST_VARIABLE.
8254	Tipo de datos no válido en el parámetro DEST_VARIABLE.
8382	El valor del parámetro POS está fuera de los límites del Array.
* Es posible visualizar los códigos de error como valores enteros o hexadecimales en el editor de programas.	

Ejemplo: instrucción Deserializar

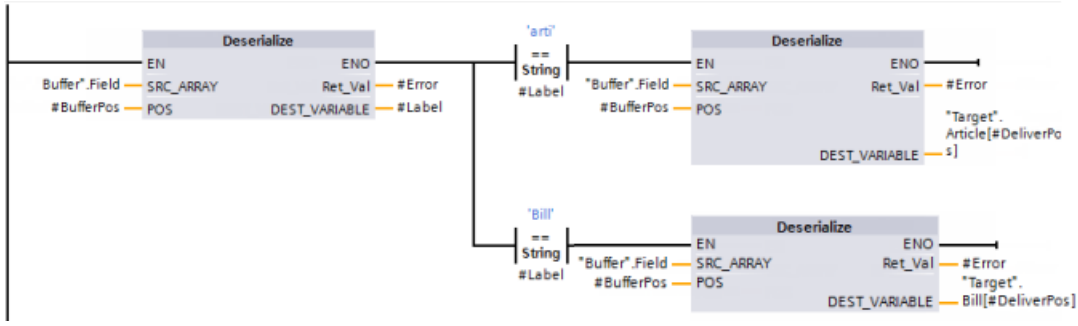
El ejemplo siguiente muestra cómo trabaja la instrucción:

Segmento 1:



La instrucción "MOVE" desplaza el valor "0" a la variable del bloque de datos "#BufferPos". Seguidamente, la instrucción Deserialize deserializa la representación secuencial de los datos del cliente del bloque de datos "Buffer" y la escribe en el bloque de datos "Target". La instrucción Deserialize calcula el número de bytes que usan los datos convertidos y los guarda en la variable del bloque de datos "#BufferPos".

Segmento 2:



La instrucción "Deserialize" deserializa la representación secuencial del flujo de datos apuntado por "Buffer" y escribe los caracteres en el operando "#Label". La lógica compara los caracteres empleando las instrucciones de comparación "arti" y "Bill". Si la comparación de "arti" = TRUE, los datos son datos de artículo que deben deserializarse y escribirse en la estructura de datos de "Article" del bloque de datos "Target". Si la comparación de "Bill" = TRUE, los datos son datos de facturación que deben deserializarse y escribirse en la estructura de datos de "Bill" del bloque de datos "Target".

Interfaz del bloque de función (o la función):

		Nombre	Tipo de datos
1		Input	
2		DeliverPos	Int
3		Output	
4		InOut	
5		Static	
6		Temp	
7		BufferPos	DInt
8		Error	Int
9		Label	String[4]

Tipos de datos PLC personalizados:

La estructura de los dos tipos de datos PLC (UDTs) para este ejemplo es la siguiente:

Article

		Nombre	Tipo de datos
1		Number	DInt
2		Declaration	String
3		Colli	Int

Client

		Nombre	Tipo de datos
1		Title	Int
2		Firstname	String[10]
3		Surname	String[10]

Bloques de datos:

Los dos bloques de datos para este ejemplo son los siguientes:

Target

		Nombre	Tipo de datos
1		Static	
2		Client	"Client"
3		Article	Array[0..10] of "Article"
4		Bill	Array[0..10] of Int

Buffer

		Nombre	Tipo de datos
1		Static	
2		Field	Array[0..294] of Byte

8.6.3 Serialize

La instrucción "Serialize" se utiliza para convertir diferentes tipos de datos PLC (UDT) en una representación secuencial sin perder la estructura.

La instrucción puede emplearse para guardar temporalmente varios elementos de datos estructurados del programa en un búfer, por ejemplo en un bloque de datos global, y enviarlos a otra CPU. El área de memoria en la que se guardan los tipos de datos PLC convertidos debe tener el tipo de datos ARRAY of BYTE y declararse como acceso estándar. Asegúrese de que hay suficiente espacio de memoria antes de ejecutar la conversión.

El parámetro POS contiene información sobre el número de bytes que usa el tipo de datos PLC convertido.

Nota

Si solo desea enviar un tipo de datos PLC simple (UDT), puede emplear la instrucción "TSEND: enviar datos a través de la conexión".

Tabla 8-77 Instrucción SERIALIZE

KOP / FUP	SCL	Descripción

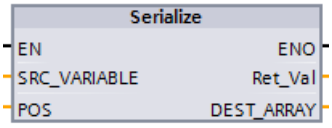
	`ret_val := Serialize(`
	`  SRC_VARIABLE=>_variant_in_,`
	`  DEST_ARRAY:=_variant_out_,`
	`  POS:=_dint_inout_);`
		Convierte un tipo de datos PLC (UDT) en una representación secuencial.

Tabla 8-78 Parámetros de la instrucción SERIALIZE

Parámetro	Tipo	Tipo de datos	Descripción
SRC_VARIABLE	IN	Variant	El tipo de datos PLC (UDT) que debe convertirse en una representación serie
DEST_ARRAY	INOUT	Variant	Bloque de datos en el que debe guardarse el flujo de datos generado
POS	INOUT	DInt	Número de bytes que usa el tipo de datos PLC convertido. El parámetro POS calculado está basado en cero.
RET_VAL	OUT	Int	Información de error

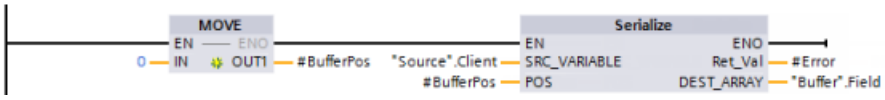
Tabla 8-79 Parámetro RET_VAL

RET_VAL* (W#16#...)	Descripción
0000	No hay error
80B0	Las áreas de memoria para los parámetros SRC_VARIABLE y DEST_ARRAY se solapan.
8150	El tipo de datos Variant en el parámetro SRC_VARIABLE no contiene valores.
8152	Error de generación del código en el parámetro SRC_VARIABLE.
8236	El bloque de datos del parámetro DEST_ARRAY no es un bloque con acceso estándar.
8250	El tipo de datos Variant en el parámetro DEST_ARRAY no contiene valores.
8252	Error de generación del código en el parámetro DEST_ARRAY.
8253	No hay suficiente memoria libre en el parámetro DEST_ARRAY.
8254	Tipo de datos no válido en el parámetro DEST_VARIABLE.
8382	El valor del parámetro POS está fuera de los límites del Array.
* Es posible visualizar los códigos de error como valores enteros o hexadecimales en el editor de programas.	

Ejemplo: Instrucción Serialize

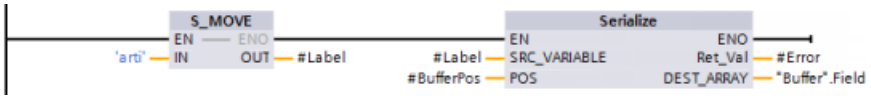
El ejemplo siguiente muestra cómo trabaja la instrucción:

Segmento 1:



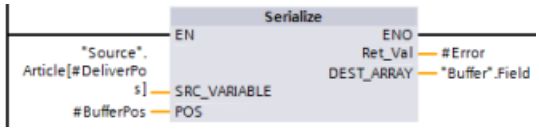
La instrucción "MOVE" desplaza el valor "0" al parámetro "#BufferPos". La instrucción "Serialize" serializa los datos de cliente del bloque de datos "Source" y los escribe en representación secuencial en el bloque de datos "Buffer". La instrucción guarda el número de bytes usados por la representación secuencial en el parámetro "#BufferPos".

Segmento 2:



Seguidamente, la lógica inserta algún texto separador para facilitar la deserialización posterior de la representación secuencial. La instrucción "S_MOVE" desplaza la cadena de texto "arti" al parámetro "#Label". La instrucción "Serialize" escribe estos caracteres en el bloque de datos "Buffer" a continuación de los datos de cliente de origen. La instrucción agrega el número de bytes de la cadena de texto "arti" al número previamente guardado en el parámetro "#BufferPos".

Segmento 3:



La instrucción "Serialize" serializa los datos de un artículo concreto, que se calcula en runtime, del bloque de datos "Source" y los escribe en representación secuencial en el bloque de datos "Buffer" después de los caracteres "arti".

Interfaz de bloque:

		Nombre	Tipo de datos
1		Input	
2		DeliverPos	Int
3		Output	
4		InOut	
5		Static	
6		Temp	
7		BufferPos	DInt
8		Error	Int
9		Label	String[4]

Tipos de datos PLC personalizados:

La estructura de los dos tipos de datos PLC (UDTs) para este ejemplo es la siguiente:

Article

	Nombre	Tipo de datos
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client

	Nombre	Tipo de datos
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Bloques de datos:

Los dos bloques de datos para este ejemplo son los siguientes:

Source

		Nombre	Tipo de datos
1		Static	
2		Client	"Client"
3		Article	Array[0..10] of "Article"

Buffer

		Nombre	Tipo de datos
1		Static	
2		Field	Array[0..294] of Byte

8.6.4 FILL_BLK (rellenar área) y UFILL_BLK (rellenar área sin interrupciones)

Tabla 8-80 Instrucciones FILL_BLK y UFILL_BLK

KOP / FUP	SCL	Descripción

	`FILL_BLK(`
	`  in:=_variant_in,`
	`  count:=int,`
	`  out=>_variant_out);`
		Instrucción "Rellenar área sin interrupciones": rellena un área de direcciones con copias de un elemento de datos específico

	`UFILL_BLK(`
	`  in:=_variant_in,`
	`  count:=int,`
	`  out=>_variant_out);`
		Instrucción "Rellenar área sin interrupciones": Rellena un área de direcciones con copias de un elemento de datos específico

Tabla 8-81 Tipos de datos para parámetros

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Dirección de origen de los datos
COUNT	UDint, USInt, UInt	Número de elementos de datos que deben copiarse
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Dirección de destino de los datos

Nota

Reglas para las operaciones de rellenar área

- Para rellenar el tipo de datos BOOL utilice SET_BF, RESET_BF, R, S, o bobina de salida (KOP)
- Para rellenar con un solo tipo de datos simple, utilice MOVE
- Para rellenar una matriz con un tipo de datos simple, utilice FILL_BLK o UFILL_BLK
- Para rellenar un solo carácter en una cadena, utilice MOVE
- Las instrucciones FILL_BLK y UFILL_BLK no pueden utilizarse para rellenar matrices en las áreas de memoria I, Q o M.

Las instrucciones FILL_BLK y UFILL_BLK copian el elemento de datos de origen IN en el destino, cuya dirección inicial se indica en el parámetro OUT. El proceso de copia se repite y un área de direcciones adyacentes se rellena hasta que el número de copias sea igual al parámetro COUNT.

Las instrucciones FILL_BLK y UFILL_BLK se diferencian en la forma de procesar las alarmas:

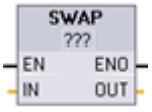
- Los eventos de alarma **se ponen en cola de espera y se procesan** durante la ejecución de FILL_BLK. Utilice la instrucción FILL_BLK si los datos contenidos en la dirección de destino del desplazamiento no se utilizan en un OB de alarma. Si se utilizan, los datos de destino no tienen que ser coherentes.
- Los eventos de alarma **se ponen en cola de espera pero no se procesarán** hasta que no finalice la ejecución de UFILL_BLK. Utilice la instrucción UFILL_BLK si la operación de desplazamiento debe finalizarse y los datos de destino deben ser coherentes antes de la ejecución de un OB de alarma.

Tabla 8-82 Estado de ENO

ENO	Condición	Resultado
1	No hay error	El elemento IN se ha copiado correctamente a todos los destinos de COUNT.
0	El rango de destino (OUT) excede el área de memoria disponible	Se copian los elementos que quepan. No se copian elementos parciales.

8.6.5 SWAP (cambiar disposición)

Tabla 8-83 Instrucción SWAP

KOP / FUP	SCL	Descripción
	`out := SWAP(in);`	Invierte el orden de los bytes para elementos de datos de dos bytes y cuatro bytes. El orden de los bits no se modifica dentro de los distintos bytes. ENO es siempre TRUE (verdadero) tras ejecutarse la instrucción SWAP.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-84 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Word, DWord	Bytes de datos ordenados en IN
OUT	Word, DWord	Bytes de datos en orden inverso en OUT

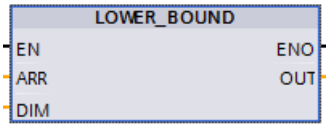
Ejemplo 1	Parámetro IN = MB0 (antes de la ejecución)		Parámetro OUT = MB4 (tras la ejecución)	
Dirección	MW0	MB1	MW4	MB5
W#16#1234	12	34	34	12
WORD	MSB	LSB	MSB	LSB

Ejemplo 2	Parámetro IN = MB0 (antes de la ejecución)				Parámetro OUT = MB4 (tras la ejecución)			
Dirección	MD0	MB1	MB2	MB3	MD4	MB5	MB6	MB7
DW#16# 12345678 DWORD	12 MSB	34	56	78 LSB	78 MSB	56	34	12 LSB

8.6.6 LOWER_BOUND: (leer límite inferior del ARRAY)

Tabla 8-85 Instrucción LOWER_BOUND

KOP / FUP	SCL	Descripción

	`out := LOWER_BOUND(`
	`  ARR:=_variant_in_,`
	`  DIM:=_udint_in_);`
		En la interfaz del bloque es posible declarar variables con ARRAY[*]. Para estas variables locales pueden leerse los límites del ARRAY. La dimensión necesaria debe especificarse en el parámetro DIM.
		La instrucción LOWER_BOUND (leer límite inferior del ARRAY) permite leer el límite inferior variable del ARRAY.

Parámetros

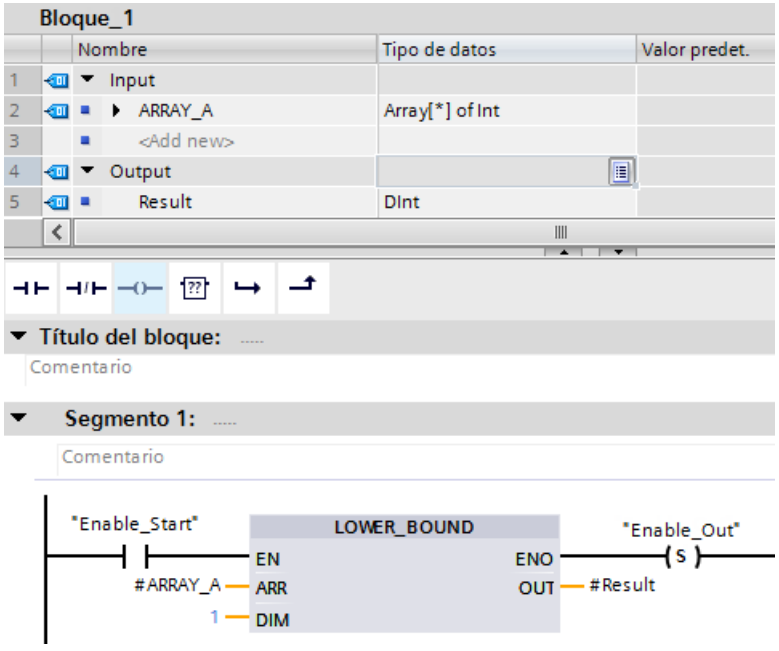
La tabla siguiente muestra los parámetros de la instrucción "LOWER_BOUND: leer límite inferior del ARRAY":

Parámetros	Declaración	Tipo de datos	Área de memoria	Descripción
EN	Input	BOOL	I, Q, M, D, L	Entrada de habilitación
ENO	Output	BOOL	I, Q, M, D, L	La salida de habilitación ENO tiene el estado lógico "0" si se cumple una de las siguientes condiciones: • La entrada de habilitación EN tiene el estado lógico "0". • La dimensión especificada en la entrada DIM no existe.
ARR	Input	ARRAY [*]	FB: sección InOut FC: secciones Input e InOut	ARRAY del que debe leerse el límite inferior variable.
DIM	Input	UDINT	I, Q, M, D, L o constante	Dimensión del ARRAY del que debe leerse el límite inferior variable.
OUT	Output	DINT	I, Q, M, D, L	Resultado

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)":

Ejemplo

En la interfaz del bloque de función (FC), el parámetro de entrada ARRAY_A es un array unidimensional con dimensiones variables.



Si el operando "Enable_Start" devuelve el estado lógico "1", la CPU ejecuta la instrucción LOWER_BOUND . Lee el límite inferior variable del ARRAY #ARRAY_A a partir del array unidimensional. Si la instrucción se ejecuta sin errores, activa el operando "Enable_Out" y pone el operando "Result" al límite inferior del array.

8.6.7 UPPER_BOUND: (leer límite superior de ARRAY)

Tabla 8-86 Instrucción LOWER_BOUND

KOP / FUP	SCL	Descripción

UPPER_BOUND
EN ENO
ARR OUT
DIM
	`out := UPPER_BOUND(`
	`  ARR:=_variant_in_,`
	`  DIM:=_udint_in_);`
		En la interfaz del bloque es posible declarar variables con ARRAY[*]. Para estas variables locales pueden leerse los límites del ARRAY. La dimensión necesaria debe especificarse en el parámetro DIM.
		La instrucción UPPER_BOUND (leer límite superior de ARRAY) permite leer el límite superior variable del ARRAY.

Parámetros

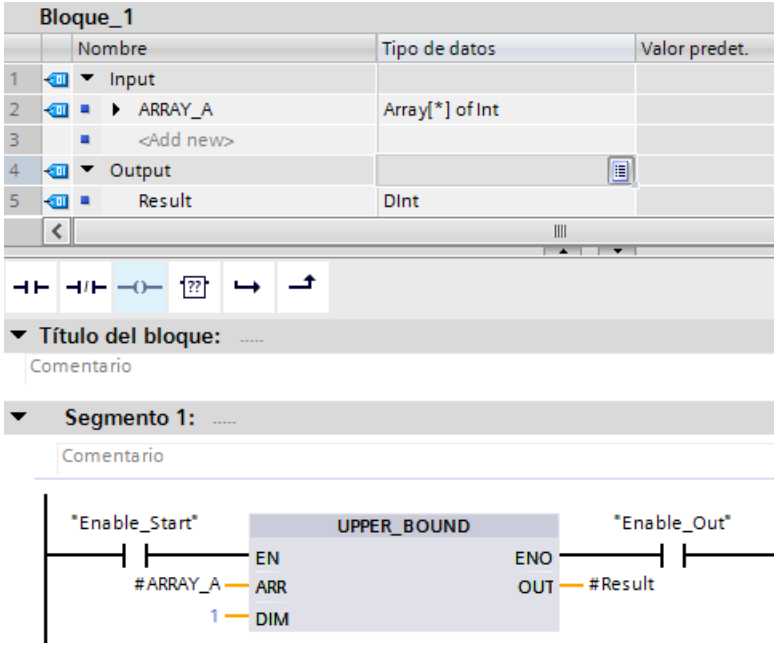
La tabla siguiente muestra los parámetros de la instrucción "UPPER_BOUND: leer límite superior de ARRAY":

Parámetros	Declaración	Tipo de datos	Área de memoria	Descripción
EN	Input	BOOL	I, Q, M, D, L	Entrada de habilitación
ENO	Output	BOOL	I, Q, M, D, L	Salida de habilitación
ARR	Input	ARRAY [*]	FB: sección InOut FC: secciones Input e InOut	ARRAY del que debe leerse el límite superior variable.
DIM	Input	UDINT	I, Q, M, D, L o constante	Dimensión del ARRAY del que debe leerse el límite superior variable.
OUT	Output	DINT	I, Q, M, D, L	Resultado

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)":

Ejemplo

En la interfaz del bloque de función (FC), el parámetro de entrada ARRAY_A es un array unidimensional con dimensiones variables.



Si el operando "Enable_Start" devuelve el estado lógico "1", la CPU ejecuta la instrucción . Lee el límite superior variable del ARRAY #ARRAY_A a partir del array unidimensional. Si la instrucción se ejecuta sin errores, activa los operandos "Enable_Out" y "Result".

8.6.8 Instrucciones de leer / escribir memoria

8.6.8.1 PEEK y POKE (solo SCL)

SCL ofrece instrucciones PEEK y POKE que permiten leer de bloques de datos, E/S o memoria, o escribir en ellos. Hay que proporcionar parámetros de offsets de byte, o de bit, específicos para la operación.

Nota

Para usar las instrucciones PEEK y POKE con bloques de datos deben usarse bloques de datos estándar (no optimizados). Nótese que las instrucciones PEEK y POKE solo transfieren datos. Desconocen los tipos de datos que hay en las direcciones.

```
PEEK(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_);
```

Lee el byte referenciado por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo de referencia al bloque de datos:

```
%MB100 := PEEK(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

Ejemplo de referencia a la entrada IB3:

```
%MB100 := PEEK(area:=16#81,
dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_);
```

Lee la palabra referenciada por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MW200 := PEEK_WORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_);
```

Lee la doble palabra referenciada por byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MD300 := PEEK_DWORD(area:=16#84,
dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
          dbNumber:=_in_,
          byteOffset:=_in_,
          bitOffset:=_in_);
```

Lee un booleano referenciado por el bitOffset y el byteOffset del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
dbNumber:=1, byteOffset:=#ii,
bitOffset:=#j);
```

```
POKE(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     value:=_in_);
```

Escribe el valor (Byte, Word o DWord) en el byteOffset referenciado del bloque de datos, E/S o área de memoria referenciados.

Ejemplo de referencia al bloque de datos:

```
POKE(area:=16#84, dbNumber:=2,
byteOffset:=3, value:="Tag_1");
```

Ejemplo de referencia a la salida QB3:

```
POKE(area:=16#82, dbNumber:=0,
byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
     dbNumber:=_in_,
     byteOffset:=_in_,
     bitOffset:=_in_,
     value:=_in_);
```

Escribe el valor booleano en el bitOffset y byteOffset referenciados del bloque de datos, E/S o área de memoria referenciados.

Ejemplo:

```
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_);
```

Escribe el número de "recuento" de bytes empezando por el offset de bytes referenciado del bloque de datos, E/S o área de memoria de origen referenciados en el byteOffset referenciado del bloque de datos, E/S o área de memoria de destino referenciados.

Ejemplo:

```
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);
```

Para las instrucciones PEEK y POKE, se aplican los valores siguientes de los parámetros "area", "area_src" y "area_dest". Para otras áreas distintas de los bloques de datos, el parámetro dbNumber debe ser 0.

16#81	I
16#82	Q
16#83	M
16#84	DB

8.6.8.2 Instrucciones Leer y escribir en formato Big y Little Endian (SCL)

La CPU S7-1200 ofrece instrucciones SCL para leer y escribir datos en formato Little-Endian y en formato Big-Endian. El formato Little-Endian significa que el byte con el bit menos significativo está en la dirección de memoria más baja. El formato Big-Endian significa que el byte con el bit más significativo está en la dirección de memoria más baja.

Las cuatro instrucciones SCL para leer y escribir datos en formato Little-Endian y Big-Endian son las siguientes:

- READ_LITTLE (Leer datos en formato Little-Endian)
- WRITE_LITTLE (Escribir datos en formato Little-Endian)
- READ_BIG (Leer datos en formato Big-Endian)
- WRITE_BIG (Escribir datos en formato Big-Endian)

Tabla 8-87 Instrucciones Leer y escribir en formato Big y Little Endian

KOP / FUP	SCL	Descripción

No disponible	`READ_LITTLE(`
	`  src_array:=_variant_in_,`
	`  dest_Variable =>_out_,`
	`  pos:=_dint_inout)`
		Lee datos de un área de memoria y los escribe en una variable individual en formato de byte Little-Endian.

No disponible	`WRITE_LITTLE(`
	`  src_variable:=_in_,`
	`  dest_array =>_variant_inout_,`
	`  pos:=_dint_inout)`
		Escribe datos de una variable individual en un área de memoria en formato de byte Little-Endian.

No disponible	`READ_BIG(`
	`  src_array:=_variant_in_,`
	`  dest_Variable =>_out_,`
	`  pos:=_dint_inout)`
		Lee datos de un área de memoria y los escribe en una variable individual en formato de byte Big-Endian.

No disponible	`WRITE_BIG(`
	`  src_variable:=_in_,`
	`  dest_array =>_variant_inout_,`
	`  pos:=_dint_inout)`
		Escribe datos de una variable individual en un área de memoria en formato de byte Big-Endian.

Tabla 8-88 Parámetros de las instrucciones READ_LITTLE and READ_BIG

Parámetro	Tipo de datos	Descripción
src_array	Array of Byte	Área de memoria desde la que se leen datos
dest_Variable	Cadenas de bits, enteros, números en coma flotante, temporizadores, fecha y hora, cadenas de caracteres	Variable de destino en la que se escriben los datos
pos	DINT	Posición basada en cero desde la que se comienzan a leer datos de la entrada src_array.

Tabla 8-89 Parámetros de las instrucciones WRITE_LITTLE and WRITE_BIG

Parámetro	Tipo de datos	Descripción
src_variable	Cadenas de bits, enteros, números en coma flotante, TOD, DATA, Char, WChar	Datos de origen de la variable
dest_array	Array of Byte	Área de memoria en la que se escriben datos
pos	DINT	Posición basada en cero en la que se comienzan a escribir datos en la salida dest_array.

Tabla 8-90 Parámetro RET_VAL

RET_VAL* (W#16#...)	Descripción
0000	Sin errores
80B4	SRC_ARRAY o DEST_ARRAY no es un Array of Byte
8382	El valor del parámetro POS está fuera de los límites del Array.
8383	El valor del parámetro POS está dentro de los límites del Array pero el tamaño del área de memoria excede el límite superior del Array.
*Es posible visualizar los códigos de error como valores enteros o hexadecimales en el editor de programas.	

8.6.9 Instrucciones Variant

8.6.9.1 VariantGet (Leer valor de una variable VARIANT)

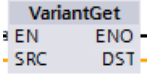
Existe la posibilidad de usar la instrucción "Leer valor de una variable Variant" para leer el valor de una variable a la que apunta el puntero Variant en el parámetro SRC y escribirlo en la variable del parámetro DST.

El parámetro SRC tiene el tipo de datos Variant. Cualquier tipo de datos excepto para Variant puede especificarse en el parámetro DST.

El tipo de datos de la variable en el parámetro DST debe coincidir con el tipo de datos al que apunta Variant.

Tabla 8-91 Instrucción VariantGet

KOP / FUP	SCL	Descripción

	`VariantGet(`
	`  SRC:=_variant_in_,`
	`  DST=>_variant_out_);`
		Lee la variable a la que apunta el parámetro SRC y la escribe en la variable del parámetro DST.

Nota

Para copiar estructuras y Arrays, puede utilizarse la instrucción "MOVE_BLK_VARIANT: Copiar área".

Tabla 8-92 Parámetros de la instrucción VariantGet

Parámetro	Tipo de datos	Descripción
SRC	Variant	Puntero hacia los datos de origen
DST	Cadenas de bits, enteros, números en coma flotante, temporizadores, fecha y hora, cadenas de caracteres, elementos ARRAY, tipos de datos PLC	Destino en el que se escriben los datos

Tabla 8-93 Estado de ENO

ENO	Condición	Resultado
1	No hay error	La instrucción ha copiado los datos de variable a los que apuntaba SRC en la variable DST.
0	La entrada de habilitación EN tiene el estado lógico "0" o el tipo de datos no concuerda.	La instrucción no ha copiado datos.

8.6.9.2 VariantPut (escribir valor en una variable VARIANT)

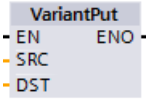
La instrucción "Escribir valor en una variable VARIANT" se utiliza para escribir el valor de la variable en el parámetro SRC en la variable del parámetro DST al que apunta VARIANT.

El parámetro DST tiene el tipo de datos VARIANT. Cualquier tipo de datos excepto para VARIANT puede especificarse en el parámetro SRC.

El tipo de datos de la variable en el parámetro SRC debe coincidir con el tipo de datos a la que apunta VARIANT.

Tabla 8-94 Instrucción VariantPut

KOP / FUP	SCL	Descripción

	`VariantPut(`
	`  SRC:=_variant_in_,`
	`  DST=>_variant_in_);`
		Escribe la variable referenciada por el parámetro SRC en la variante a la que apunta el parámetro DST

Nota

Para copiar estructuras y Arrays, puede utilizarse la instrucción "MOVE_BLK_VARIANT: Copiar área".

Tabla 8-95 Parámetros de la instrucción VariantPut

Parámetro	Tipo de datos	Descripción
SRC	Bit strings, integers, floating-point numbers, timers, date and time, character strings, ARRAY elements, PLC data types	Puntero hacia los datos de origen
DST	Variant	Destino en el que se escriben los datos

Tabla 8-96 Estado de ENO

ENO	Condición	Resultado
1	No hay error	La instrucción ha copiado los datos de la variable SRC en la variable DST.
0	La entrada de habilitación EN tiene el estado lógico "0" o el tipo de datos no concuerda.	La instrucción no ha copiado datos.

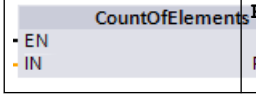
8.6.9.3 CountOfElements (consultar número de elementos de ARRAY)

La instrucción "Consultar número de elementos de ARRAY" se utiliza para consultar cuántos elementos Array hay en una variable a la que apunta Variant.

Si se trata de un ARRAY unidimensional, la instrucción devuelve la diferencia entre los límites superior e inferior +1. Si se trata de un Array multidimensional, la instrucción devuelve el producto de todas las dimensiones.

Tabla 8-97 Instrucción CountOfElements

KOP / FUP	SCL	Descripción

	`Result := CountOfElements(`
	`  ENO=>variant_in_);`
	RET_VAL
		Cuenta el número de elementos Array a los que apunta el parámetro IN.

Nota

Si Variant apunta a un Array of Bool, la instrucción cuenta los elementos de relleno hasta el límite de bytes más cercano. Por ejemplo, la instrucción devuelve 8 como recuento de un Array[0..1] of Bool.

Tabla 8-98 Parámetros de la instrucción CountOfElements

Parámetro	Tipo de datos	Descripción
IN	Variant	Variable con elementos Array que deben contarse
RET_VAL	UDint	Resultado de la instrucción

Tabla 8-99 Estado de ENO

ENO	Condición	Resultado
1	No hay error	La instrucción devuelve el número de elementos Array.
0	La entrada de habilitación EN tiene el estado lógico "0" o Variant no apunta a un Array.	La instrucción devuelve 0.

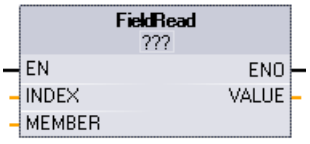
8.6.10 Instrucciones Legacy

8.6.10.1 Instrucciones FieldRead (Leer campo) y FieldWrite (Escribir campo)

Nota

STEP 7 V10.5 **no soportaba** una referencia variable como índice de matriz o matrices multidimensionales. Las instrucciones FieldRead y FieldWrite se utilizaban para proporcionar operaciones de índice de matriz variables para una matriz unidimensional. STEP 7 V11 y superior **sí que admite** una variable como índice de matriz y matrices multidimensionales. FieldRead y FieldWrite se han incluido en STEP 7 V11 y superior para disponer de una mayor compatibilidad retroactiva con programas que utilizaban estas instrucciones.

Tabla 8-100 Instrucciones FieldRead y FieldWrite

KOP / FUP	SCL	Descripción
	`value := member[index];`	FieldRead lee el elemento de matriz con el valor de índice INDEX de la matriz cuyo primer elemento está especificado por el parámetro MEMBER. El valor del elemento de matriz se transfiere a la posición especificada en el parámetro VALUE.
	`member[index] := value;`	WriteField transfiere el valor de la posición especificada por el parámetro VALUE a la matriz cuyo primer elemento se especifica en el parámetro MEMBER. El valor se transfiere al elemento de la matriz cuyo índice de matriz se especifica en el parámetro INDEX.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-101 Tipos de datos para parámetros

Parámetro y tipo		Tipo de datos	Descripción
Índice alfabético	Entrada	DInt	El número de índice del elemento de matriz que debe leerse o en el que debe escribirse
Miembro [1]	Entrada	Números binarios, enteros, números en coma flotante, temporizadores, DATE, TOD, CHAR y WCHAR como componentes de una variable ARRAY	Posición del primer elemento en una matriz unidimensional definida en un bloque de datos global o una interfaz de bloque. Ejemplo: Si el índice de matriz se especifica como [-2..4], el índice del primer elemento es -2 y no 0.
Valor [1]	Out	Números binarios, enteros, números en coma flotante, temporizadores, DATE, TOD, CHAR, WCHAR	Posición en la que se copia el elemento de matriz especificado (FieldRead) Posición del valor que se copia en el elemento de matriz especificado (FieldWrite)

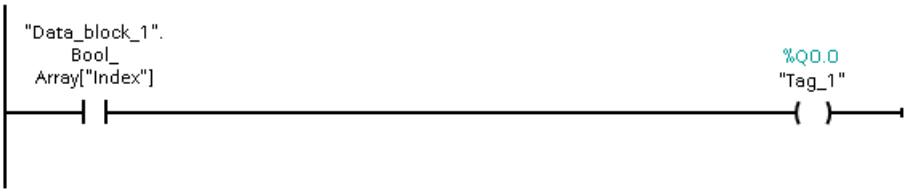
[1] El tipo de datos del elemento de matriz especificado por los parámetros MEMBER y VALUE debe tener el mismo tipo de datos.

La salida de habilitación ENO es 0 si se cumple una de las siguientes condiciones:

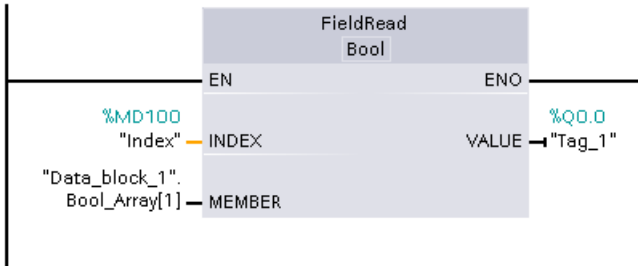
- La entrada EN tiene el estado lógico "0"
- El elemento de matriz especificado en el parámetro INDEX no está definido en la matriz referenciada en el parámetro MEMBER
- Durante el procesamiento se produce un error, p. ej. un rebosamiento

Ejemplo: Acceso a datos mediante indexación de matriz

Para acceder a los elementos de una matriz con una variable, basta con utilizar la variable como índice de matriz en la lógica del programa. Por ejemplo, el segmento siguiente activa una salida basada en el valor booleano de una matriz de booleanos en "Data_block_1" referenciada por la variable PLC "Índice".



La lógica del índice de matriz variable es equivalente al método antiguo utilizando la instrucción FieldRead:



Las instrucciones FieldWrite y FieldRead se pueden sustituir por la lógica de índice de matriz variable.

SCL no tiene instrucciones FieldRead ni FieldWrite, pero admite el direccionamiento indirecto de una matriz con una variable:

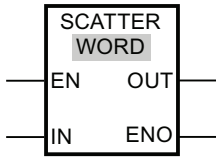
```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

8.6.11 SCATTER

SCATTER: Dispersión de una secuencia de bits en bits individuales

La instrucción Dispersión de una secuencia de bits en bits individuales descompone una variable del tipo de datos BYTE, WORD o DWORD en bits individuales y los guarda en un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC exclusivamente con elementos booleanos.

Tabla 8-102 SCATTER

KOP / FUP	SCL	Descripción
	`SCATTER (IN := #SourceWord, OUT => #DestinationArray );`	La instrucción SCATTER: Dispersión de una secuencia de bits en bits individuales descompone una variable del tipo de datos BYTE, WORD o DWORD en bits individuales y los guarda en un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC exclusivamente con elementos booleanos.

Nota

ARRAY of BOOL multidimensional

Con la instrucción "Dispersión de una secuencia de bits en bits individuales" no está permitido utilizar un ARRAY of BOOL multidimensional.

Nota

Longitud del ARRAY, STRUCT o tipo de datos PLC

El ARRAY, el STRUCT anónimo o el tipo de datos PLC debe tener exactamente el número de elementos especificado en la secuencia de bits. Para el tipo de datos BYTE, por ejemplo, esto significa que ARRAY, STRUCT o el tipo de datos PLC debe tener exactamente 8 elementos (WORD = 16 y DWORD = 32).

Nota

Nota sobre la disponibilidad de la instrucción

La instrucción puede utilizarse con una CPU de la serie S7-1200 a partir de la versión de firmware > 4.2 y, con una CPU de la serie S7-1500, a partir de la versión de firmware 2.1.

De esta forma es posible descomponer una palabra de estado, además de leer y cambiar el estado de los diferentes bits mediante el índice, por ejemplo. El uso de GATHER permite recomponer los bits en una secuencia.

La salida de habilitación ENO devuelve el estado lógico "0" si se cumple una de las siguientes condiciones:

- La entrada de habilitación EN tiene el estado lógico "0".
- El ARRAY, STRUCT o tipo de datos PLC no proporciona suficientes elementos BOOL.

Tipos de datos de la instrucción SCATTER

La tabla siguiente muestra los parámetros de la instrucción:

Parámetro	Declaración	Tipo de datos		Área de memoria	Descripción
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L o constante	Entrada de habilitación
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Salida de habilitación
IN	Input	BYTE, WORD, DWORD	BYTE, WORD, DWORD	I, Q, M, D, L	Secuencia de bits que se ha descompuesto. Los valores no deben estar en el área de periferia o en el DB de un objeto tecnológico.
OUT	Output	ARRAY[*] of BOOL, STRUCT o tipo de datos PLC *: 8, 16, 32 o 64 elementos	ARRAY[*] of BOOL, STRUCT o tipo de datos PLC *: 8, 16, 32 o 64 elementos	I, Q, M, D, L	ARRAY, STRUCT o tipo de datos PLC en el que se guardan los diferentes bits

Encontrará información adicional sobre los tipos de datos válidos en "Consulte también".

Ejemplo con un ARRAY

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceWord		WORD
EnableOut	Salida	BOOL
DestinationArray		ARRAY[0..15] of BOOL

El ejemplo siguiente muestra cómo trabaja la instrucción:

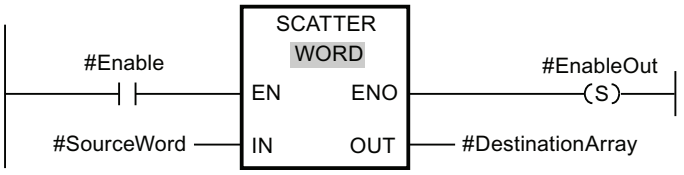


Figura 8-1 Ejemplo 2 de SCATTER

La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceWord	WORD (16 bits)
OUT	DestinationUDT	El operando "DestinationUDT" tiene el tipo de datos PLC (UDT). Consta de 16 elementos, por lo que es igual de grande que el WORD que debe descomponerse.

Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. El operando #SourceWord del tipo de datos WORD se descompone en sus diferentes bits (16) y se asigna a los elementos individuales del operando #DestinationArray. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Encontrará información adicional y el código del programa del ejemplo citado anteriormente aquí: Librería de ejemplo para instrucciones.

Ejemplo con un tipo de datos PLC (UDT)

Cree el siguiente tipo de datos 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

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceWord		WORD
EnableOut	Salida	BOOL
DestinationUDT		"myBits"

El ejemplo siguiente muestra cómo trabaja la instrucción:

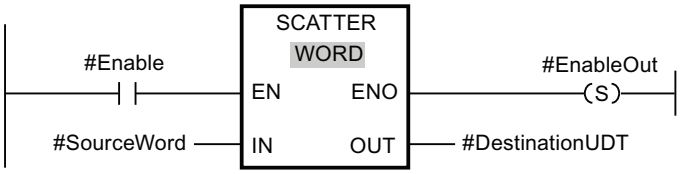


Figura 8-2 SCATTER_Bsp_example

La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceWord	WORD (16 bits)
OUT	DestinationUDT	El operando "DestinationUDT" tiene el tipo de datos PLC (UDT). Consta de 16 elementos, por lo que es igual de grande que el WORD que debe descomponerse.

Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. El operando #SourceWord del tipo de datos WORD se descompone en sus diferentes bits (16) y se asigna a los elementos individuales del operando #DestinationArray. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Consulte también

Nuevas funciones (Página 35)

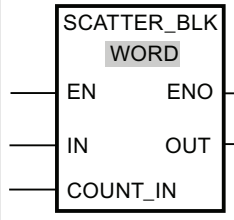
8.6.12 SCATTER_BLK

SCATTER_BLK: Dispersión de elementos de un ARRAY of secuencia de bits en bits individuales

La instrucción "Dispersión de elementos de un ARRAY of secuencia de bits en bits individuales" descompone uno o varios elementos de un ARRAY of BYTE, WORD o DWORD en bits individuales y los guarda en un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC exclusivamente con elementos booleanos. En el parámetro COUNT_IN se especifica el número de elementos del ARRAY de origen que van a descomponerse. El ARRAY de origen en el parámetro IN puede tener más elementos de los especificados en el parámetro COUNT_IN. El ARRAY of BOOL, el STRUCT anónimo o el tipo de datos PLC debe tener suficientes elementos para guardar los bits de las secuencias descompuestas. No obstante, el área de memoria de destino puede ser más grande.

Tabla 8-103 SCATTER_BLK

KOP / FUP	SCL	Descripción

	`SCATTER_BLK(IN:=_byte_in_,`
	`COUNT_IN:=_uint_in_,`
	`OUT=>_bool_out_);_IN:=_uint_in_,`
		La instrucción "Dispersión de elementos de un ARRAY of secuencia de bits en bits individuales" descompone uno o varios elementos de un ARRAY of BYTE, WORD o DWORD en bits individuales y los guarda en un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC exclusivamente con elementos booleanos. En el parámetro COUNT_IN se especifica el número de elementos del ARRAY de origen que van a descomponerse. El ARRAY de origen en el parámetro IN puede tener más elementos de los especificados en el parámetro COUNT_IN. El ARRAY of BOOL, el STRUCT anónimo o el tipo de datos PLC debe tener suficientes elementos para guardar los bits de las secuencias descompuestas. No obstante, el área de memoria de destino puede ser más grande.

Nota

NO se escriben datos cuando ENO es False

En la CPU S7-1200 y solo para la instrucción SCATTER_BLK: no se escriben datos en la salida si ENO es FALSE.

Nota

ARRAY of BOOL multidimensional

Si el ARRAY es un ARRAY of BOOL multidimensional, los bits de relleno de las dimensiones contenidas también se cuentan aunque no se hayan declarado explícitamente.

Ejemplo 1: un ARRAY[1..10,0..4,1..2] of BOOL se trata como un ARRAY[1..10,0..4,1..8] of BOOL o como un ARRAY[0..399] of BOOL.

Ejemplo 2: en el parámetro IN se interconecta un ARRAY[0..5] of WORD (sourceArrayWord[2]). El parámetro COUNT_IN tiene el valor "3". En el parámetro OUT se interconecta un ARRAY[0..1,0..5,0..7] of BOOL (destinationArrayBool[0,0,0]). Tanto el array del parámetro IN como el del parámetro OUT tienen un tamaño de 96 bits. El ARRAY of WORD se descompone en 48 bits individuales.

Nota

Si el límite inferior del ARRAY de destino no es "0", tenga en cuenta lo siguiente:

Por motivos de rendimiento, el índice debe empezar siempre en un límite de BYTE, WORD o DWORD. Esto significa que el índice debe calcularse empezando por el límite inferior del ARRAY. La fórmula siguiente se usa como base de cálculo:

Índices válidos = límite inferior de ARRAY + n(número de secuencias de bits) x número de bits de las secuencias de bits deseadas

Para un ARRAY[-2..45] of BOOL y la secuencia de bits WORD, el cálculo es el siguiente:

- Índice válido (-2) = -2 + 0 × 16
- Índice válido (14) = -2 + 1 × 16
- Índice válido (30) = -2 + 2 × 16

Encontrará un ejemplo más adelante.

Nota

Disponibilidad de la instrucción

La instrucción puede utilizarse con una CPU de la serie S7-1200 a partir de la versión de firmware > 4.2 y, con una CPU de la serie S7-1500, a partir de la versión de firmware 2.1.

De esta forma es posible descomponer palabras de estado, además de leer y cambiar el estado de los diferentes bits mediante el índice, por ejemplo. El uso de GATHER permite recomponer los bits en una secuencia.

La salida de habilitación ENO devuelve el estado lógico "0" si se cumple una de las siguientes condiciones:

- La entrada de habilitación EN tiene el estado lógico "0".
- El ARRAY de origen tiene menos elementos de los especificados en el parámetro COUNT_IN.
- El índice del ARRAY de destino no empieza en un límite de BYTE, WORD o DWORD. En este caso no se escriben resultados en el ARRAY of BOOL.
- El ARRAY[*] of BOOL, STRUCT o tipo de datos PLC no proporciona el número necesario de elementos.
  - CPU S7-1500: en este caso se descomponen y escriben todas las secuencias de bits posibles en el ARRAY of BOOL, el STRUCT anónimo o el tipo de datos PLC. Las secuencias de bits restantes ya no se tienen en cuenta.
  - CPU S7-1200: no hay ninguna operación de copia.

Tipos de datos de la instrucción SCATTER_BLK

La tabla siguiente muestra los parámetros de la instrucción:

Parámetro	Declaración	Tipo de datos		Área de memoria	Descripción
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L o constante	Entrada de habilitación
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Salida de habilitación
IN	Input	Elemento de un ARRAY[*] of <secuencia de bits>	Elemento de un ARRAY[*] of <secuencia de bits>	I, Q, M, D, L	ARRAY of <secuencia de bits> que se ha descompuesto. Los valores no deben estar en el área de periferia o en el DB de un objeto tecnológico.
COUNT_IN	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	I, Q, M, D, L	Contador del número de elementos del ARRAY de origen que van a descomponerse. El valor no debe estar en el área de periferia ni en la base de datos de un objeto tecnológico.
OUT	Output	Elemento de un ARRAY[*] of BOOL, STRUCT o tipo de datos PLC	Elemento de un ARRAY[*] of BOOL, STRUCT o tipo de datos PLC	I, Q, M, D, L	ARRAY, STRUCT o tipo de datos PLC en el que se guardan los diferentes bits

La secuencia de bits necesaria se selecciona en la lista desplegable "???" del cuadro de la instrucción.

Encontrará información adicional sobre los tipos de datos válidos en "Consulte también".

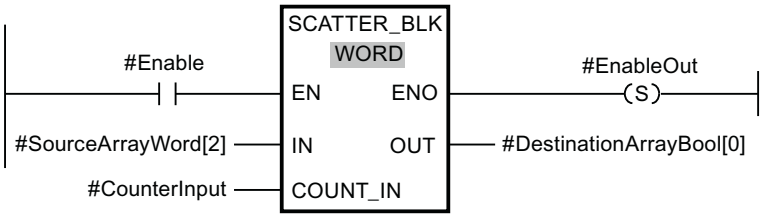
Ejemplo de un ARRAY de destino con el límite inferior "0"

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceArrayWord		ARRAY[0..5] of WORD
CounterInput		UDINT

Variable	Sección	Tipo de datos
EnableOut	Salida	BOOL
DestinationArrayBool		ARRAY[0..95] of BOOL

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceArrayWord[2]	ARRAY[0..5] of WORD (es posible descomponer 96 bits)
COUNT_IN	CounterInput = 3	UDINT3 (deben descompnerse 3 WORDs o 48 bits. Esto significa que al menos 48 bits deben estar disponibles en el ARRAY de destino.)
OUT	DestinationArrayBool[0]	El operando "DestinoArrayBool" es del tipo de datos ARRAY[0..95] of BOOL. Esto significa que proporciona 96 elementos BOOL.

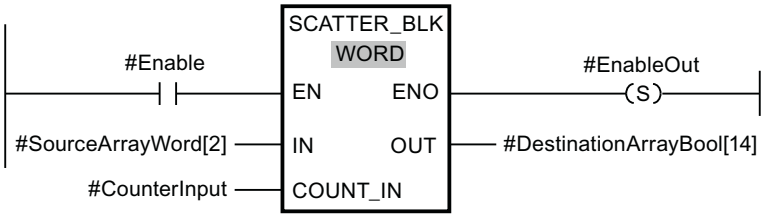
Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. El 3.°, 4.° y 5.° WORD del operando #SourceArrayWord se descompone en sus diferentes bits (48) y se asigna a partir del primer elemento a los distintos elementos del operando #DestinationArrayBool. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Ejemplo de un ARRAY de destino con el límite inferior "-2"

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceArrayWord		ARRAY[0..5] of WORD
CounterInput		UDINT
EnableOut	Salida	BOOL
DestinationArrayBool		ARRAY[-2..93] of BOOL

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceArrayWord[2]	ARRAY[0..5] of WORD (es posible descomponer 96 bits)
COUNT_IN	CounterInput = 3	UDINT3 (deben descompnerse 3 WORDs o 48 bits. Esto significa que al menos 48 bits deben estar disponibles en el ARRAY de destino.)
OUT	DestinationArrayBool[14]	El operando "DestinoArrayBool" es del tipo de datos ARRAY[-2..93] of BOOL. Esto significa que proporciona 96 elementos BOOL.

Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. El 3.°, 4.° y 5.° WORD del operando #SourceArrayWord se descompone en sus diferentes bits (48) y se asigna a partir del 16.° elemento a los distintos elementos del operando #DestinationArrayBool. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN. Los 32 bits restantes no se escriben.

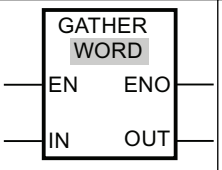
Encontrará información adicional y el código del programa del ejemplo citado anteriormente aquí: Librería de ejemplo para instrucciones.

8.6.13 GATHER

GATHER

La instrucción "Recopilación de bits individuales en una secuencia de bits" recompone los bits de un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC en una secuencia de bits exclusivamente con elementos booleanos. La secuencia de bits se guarda en una variable del tipo de datos BYTE, WORD, DWORD o LWORD.

Tabla 8-104 GATHER

KOP / FUP	SCL	Descripción
	`GATHER(IN := #SourceArray, OUT => #DestinationArray);`	La instrucción GATHER: La instrucción "Recopilación de bits individuales en una secuencia de bits" recompone los bits de un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC en una secuencia de bits exclusivamente con elementos booleanos. La secuencia de bits se guarda en una variable del tipo de datos BYTE, WORD o DWORD.

Nota

ARRAY of BOOL multidimensional

Con la instrucción "Recopilación de bits individuales en una secuencia de bits" no está permitido utilizar un ARRAY of BOOL multidimensional.

Nota

Longitud del ARRAY, STRUCT o tipo de datos PLC

El ARRAY, el STRUCT o el tipo de datos PLC debe tener exactamente el número de elementos especificado en la secuencia de bits.

Para el tipo de datos BYTE, por ejemplo, esto significa que ARRAY, STRUCT anónimo o el tipo de datos PLC debe tener exactamente 8 elementos (WORD = 16 y DWORD = 32).

Nota

Disponibilidad de la instrucción

La instrucción puede utilizarse con una CPU de la serie S7-1200 a partir de la versión de firmware > 4.2 y, con una CPU de la serie S7-1500, a partir de la versión de firmware 2.1.

La salida de habilitación ENO devuelve el estado lógico "0" si se cumple una de las siguientes condiciones:

- La entrada de habilitación EN tiene el estado lógico "0".
- El ARRAY, el STRUCT anónimo o el tipo de datos PLC (UDT) tiene menos o más elementos BOOL de los especificados en la secuencia de bits. En este caso, los elementos BOOL no se transfieren.
- Están disponibles menos bits de los necesarios.

Tipos de datos de la instrucción GATHER

La tabla siguiente muestra los parámetros de la instrucción:

Parámetro	Declaración	Tipo de datos		Área de memoria	Descripción
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L o constante	Entrada de habilitación
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Salida de habilitación
IN	Input	ARRAY[*] of BOOL, STRUCT o tipo de datos PLC *: 8, 16, 32 o 64 elementos	ARRAY[*] of BOOL, STRUCT o tipo de datos PLC *: 8, 16, 32 o 64 elementos	I, Q, M, D, L	ARRAY, STRUCT o tipo de datos PLC, cuyos bits están reagrupados en una secuencia de bits. Los valores no deben estar en el área de periferia o en el DB de un objeto tecnológico.
OUT	Output	BYTE, WORD, DWORD	BYTE, WORD, DWORD	I, Q, M, D, L	Secuencia de bits reagrupada, guardada en una variable

La secuencia de bits necesaria se selecciona en la lista desplegable "???" del cuadro de la instrucción.

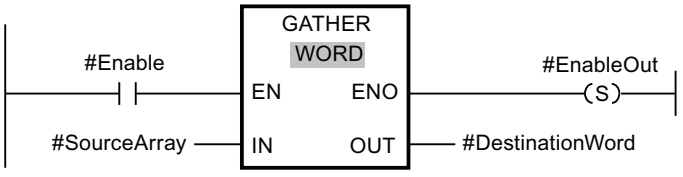
Encontrará información adicional sobre los tipos de datos válidos en "Consulte también".

Ejemplo con un ARRAY

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceArray		ARRAY[0..15] of BOOL
EnableOut	Salida	BOOL
DestinationWord		WORD

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceArray	El operando "SourceArray" es del tipo de datos ARRAY[0..15] of BOOL. Consta de 16 elementos, por lo que es igual de grande que el WORD en el que deben reagruparse los bits.
OUT	DestinationWord	WORD (16 bits)

Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. Los bits del operando #SourceArray se reagrupan en un WORD. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Encontrará información adicional y el código del programa del ejemplo citado anteriormente aquí: Librería de ejemplo para instrucciones.

Ejemplo con un tipo de datos PLC (UDT)

Cree el siguiente tipo de datos PLC "myBits":

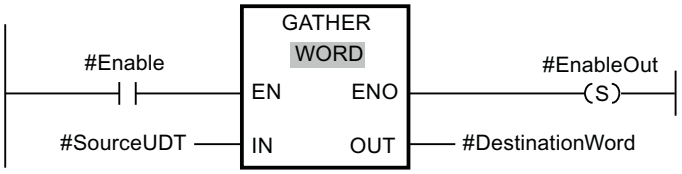
myBits

	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

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceUDT		"myBits"
EnableOut	Salida	BOOL
DestinationWord		WORD

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceUDT	El operando "SourceUDT" tiene el tipo de datos PLC (UDT). Consta de 16 elementos, por lo que es igual de grande que el WORD en el que deben reagruparse los bits.
OUT	DestinationWord	WORD (16 bits)

Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. Los bits del operando #SourceUDT se reagrupan en un WORD. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

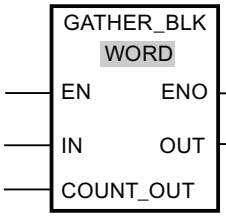
8.6.14 GATHER_BLK

Descripción

La instrucción "Recopilación de bits individuales en varios elementos de un ARRAY of secuencia de bits" recompone los bits de un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC en uno o varios elementos de un ARRAY of <secuencia de bits> exclusivamente con elementos booleanos. En el parámetro COUNT_OUT se especifica el número de elementos del ARRAY de destino que van a escribirse. Con este paso también se especifica implícitamente cuántos elementos del ARRAY of BOOL, el STRUCT anónimo o el tipo de datos PLC se necesitan. El ARRAY de destino en el parámetro OUT puede tener más elementos de los especificados en el parámetro COUNT_OUT. El ARRAY of <secuencia de bits> debe tener suficientes elementos para guardar los bits que se reagruparán. No obstante, el ARRAY de destino puede ser más grande.

Tabla 8-105 GATHER

KOP / FUP	SCL	Descripción

	`GATHER_BLK(IN := #SourceArrayBool[0],`
	`COUNT_OUT := #CounterOutput,`
	`OUT => #DestinationArrayWord[2]);`
		La instrucción "Recopilación de bits individuales en varios elementos de un ARRAY of secuencia de bits" recompone los bits de un ARRAY of BOOL, un STRUCT anónimo o un tipo de datos PLC en uno o varios elementos de un ARRAY of <secuencia de bits> exclusivamente con elementos booleanos. En el parámetro COUNT_OUT se especifica el número de elementos del ARRAY de destino que van a escribirse. Con este paso también se especifica implícitamente cuántos elementos del ARRAY of BOOL, el STRUCT anónimo o el tipo de datos PLC se necesitan. El ARRAY de destino en el parámetro OUT puede tener más elementos de los especificados en el parámetro COUNT_OUT. El ARRAY of <secuencia de bits> debe tener suficientes elementos para guardar los bits que se reagruparán. No obstante, el ARRAY de destino puede ser más grande.

Nota

NO se escriben datos cuando ENO es False

En la CPU S7-1200 y solo para la instrucción GATHER_BLK: no se escriben datos en la salida si ENO es FALSE.

Nota

ARRAY of BOOL multidimensional

Si el ARRAY es un ARRAY of BOOL multidimensional, los bits de relleno de las dimensiones contenidas también se cuentan aunque no se hayan declarado explícitamente.

Ejemplo 1: un ARRAY[1..10,0..4,1..2] of BOOL se trata como un ARRAY[1..10,0..4,1..8] of BOOL o como un ARRAY[0..399] of BOOL.

Ejemplo 2: en el parámetro OUT se interconecta un ARRAY[0..5] of WORD (sourceArrayWord[2]). El parámetro COUNT_IN tiene el valor "3". En el parámetro IN se interconecta un ARRAY[0..1,0..5,0..7] of BOOL (destinationArrayBool[0,0,0]). Tanto el array del parámetro IN como el del parámetro OUT tienen un tamaño de 96 bits. Se recomponen 48 bits individuales del ARRAY of BOOL.

Nota

Si el límite inferior del ARRAY de origen no es "0", tenga en cuenta lo siguiente:

Por motivos de rendimiento, el índice debe empezar siempre en un límite de BYTE, WORD o DWORD. Esto significa que el índice debe calcularse empezando por el límite inferior del ARRAY. La fórmula siguiente se usa como base de cálculo:

Índices válidos = límite inferior de ARRAY + n(número de secuencias de bits) x número de bits de las secuencias de bits deseadas

Para un ARRAY[-2..45] of BOOL y la secuencia de bits WORD, el cálculo es el siguiente:

- Índice válido (-2) = -2 + 0 × 16
- Índice válido (14) = -2 + 1 × 16
- Índice válido (30) = -2 + 2 × 16

Encontrará un ejemplo más adelante.

Nota

Disponibilidad de la instrucción

La instrucción puede utilizarse con una CPU de la serie S7-1200 a partir de la versión de firmware > 4.2 y, con una CPU de la serie S7-1500, a partir de la versión de firmware 2.1.

La salida de habilitación ENO devuelve el estado lógico "0" si se cumple una de las siguientes condiciones:

- La entrada de habilitación EN tiene el estado lógico "0".
- El índice del ARRAY de origen no empieza en un límite de BYTE, WORD o DWORD. En este caso no se escriben resultados en el ARRAY of <secuencia de bits>.
- El ARRAY[*] of <secuencia de bits> no proporciona el número necesario de elementos.
  - CPU S7-1500: en este caso se recomponen y escriben todas las secuencias de bits posibles en el ARRAY of <secuencia de bits>. Los bits restantes dejan de tenerse en cuenta.
  - CPU S7-1200: no hay ninguna operación de copia.

Tipos de datos de la instrucción GATHER_BLK

La tabla siguiente muestra los parámetros de la instrucción:

Parámetro	Declaración	Tipo de datos		Área de memoria	Descripción
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	I, Q, M, D, L o constante	Entrada de habilitación
ENO	Output	BOOL	BOOL	I, Q, M, D, L	Salida de habilitación
IN	Input	Elemento de un ARRAY[*] of BOOL, STRUCT o tipo de datos PLC	Elemento de un ARRAY[*] of BOOL, STRUCT o tipo de datos PLC	I, Q, M, D, L	ARRAY of BOOL, STRUCT o tipo de datos PLC cuyos bits se recomponen (ARRAY de origen) Los valores no deben estar en el área de periferia o en el DB de un objeto tecnológico.
COUNT_OUT	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	I, Q, M, D, L	Contador que cuenta el número de elementos del ARRAY de destino que van a escribirse. El valor no debe estar en el área de periferia ni en la base de datos de un objeto tecnológico.
OUT	Output	Elemento de un ARRAY[*] of <secuencia de bits>	Elemento de un ARRAY[*] of <secuencia de bits>	I, Q, M, D, L	ARRAY of <secuencia de bits> en el que se guardan los bits (ARRAY de destino)

La secuencia de bits necesaria se selecciona en la lista desplegable "???" del cuadro de la instrucción.

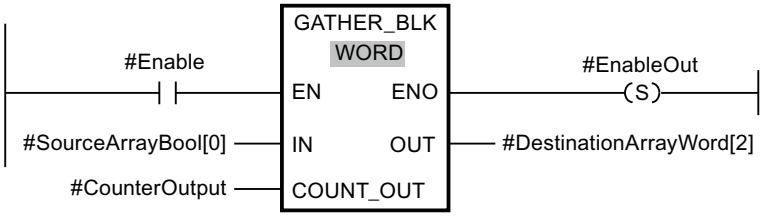
Encontrará información adicional sobre los tipos de datos válidos en "Consulte también".

Ejemplo de un ARRAY de origen con el límite inferior "0"

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceArrayBool		ARRAY[0..95] of BOOL
CounterOutput		UDINT
EnableOut	Salida	BOOL
DestinationArrayWord		ARRAY[0..5] of WORD

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceArrayBool[0]	El operando "SourceArrayBool" es del tipo de datos ARRAY[0..95] of BOOL. Esto significa que proporciona 96 elementos BOOL que pueden recomponerse en palabras.
COUNT_OUT	CounterOutput = 3	UDINT3 (deben escribirse 3 palabras. Esto significa que debe haber 48 bits disponibles en el ARRAY de origen.)
OUT	DestinationArrayWord[2]	El operando "DestinationArrayWord" es del tipo de datos ARRAY[0..5] of WORD. Esto significa que hay disponibles 6 elementos WORD.

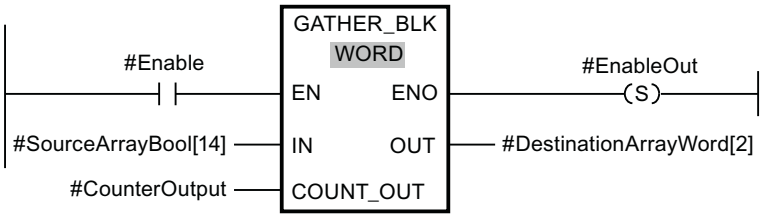
Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. A partir del primer elemento del operando #SourceArrayBool se recomponen 48 bits en el operando #DestinationArrayWord. El punto inicial en el ARRAY de destino es el tercer elemento. Esto significa que los 16 primeros bits se escriben en la tercera palabra, los segundos 16 bits en la cuarta palabra y los terceros 16 bits en la quinta palabra del ARRAY de destino. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Ejemplo de un ARRAY de origen con el límite inferior "-2"

Cree las variables siguientes en la interfaz de bloque:

Variable	Sección	Tipo de datos
Enable	Entrada	BOOL
SourceArrayBool		ARRAY[-2..93] of BOOL
CounterOutput		UDINT
EnableOut	Salida	BOOL
DestinationArrayWord		ARRAY[0..5] of WORD

El ejemplo siguiente muestra cómo trabaja la instrucción:



La tabla siguiente muestra cómo trabaja la instrucción utilizando valores de operando concretos:

Parámetro	Operando	Tipo de datos
IN	SourceArrayBool[14]	El operando "SourceArrayBool" es del tipo de datos ARRAY[-2..93] of BOOL. Puesto que el punto inicial es el 16.º elemento, solo están disponibles 80 elementos BOOL que pueden recomponerse en palabras.
COUNT_OUT	CounterOutput = 3	UDINT3 (deben escribirse 3 palabras. Esto significa que debe haber 48 bits disponibles en el ARRAY de origen.)
OUT	DestinationArrayWord[2]	El operando "DestinationArrayWord" es del tipo de datos ARRAY[0..5] of WORD. Esto significa que hay disponibles 6 elementos WORD.

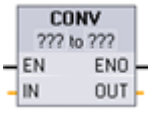
Si el operando #Enable devuelve el estado lógico "1" en la entrada de habilitación EN, la instrucción se ejecuta. Empezando por el 16.º elemento del operando #SourceArrayBool se recomponen 48 bits en el operando #DestinationArrayWord. El punto inicial en el ARRAY de destino es el tercer elemento. Esto significa que los 16 primeros bits del ARRAY de origen se ignoran. Los segundos 16 bits se escriben en la tercera palabra, los terceros 16 bits en la cuarta palabra y los cuartos 16 bits en la quinta palabra del ARRAY de destino. Los 64 bits restantes del ARRAY de origen no se tienen en cuenta. Si se produce un error mientras se ejecuta la instrucción, el operando #EnableOut devuelve el estado lógico "0" en la salida de habilitación EN.

Encontrará información adicional y el código del programa del ejemplo citado anteriormente aquí: Librería de ejemplo para instrucciones.

8.7 Conversión

8.7.1 CONV (convertir valor)

Tabla 8-106 Instrucción Convertir valor (CONV)

KOP / FUP	SCL	Descripción
	`out := <data type in>_TO_<data type out>(in);`	Convierte un elemento de datos de un tipo de datos a otro tipo de datos.

[1] En KOP y FUP: haga clic en "???" y seleccione los tipos de datos del menú desplegable.

[2] En SCL: Defina la instrucción de conversión identificando el tipo de datos del parámetro de entrada (in) y el parámetro de salida (out). Por ejemplo, DWORD_TO_REAL convierte un valor DWord en un valor Real.

Tabla 8-107 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Cadena de bits[1], SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Valor de entrada
OUT	Cadena de bits[1], SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Valor de entrada convertido a un nuevo tipo de datos

[1] La instrucción no le permite seleccionar cadenas de bits (Byte, Word, DWord). Para introducir como parámetro de la instrucción un operando del tipo de datos Byte, Word o DWord, seleccione un entero sin signo con la misma longitud en bits. Por ejemplo, USInt para un Byte, UInt para un Word o UDInt para un DWord.

Tras haber seleccionado el tipo de datos que se desea convertir, las conversiones posibles aparecen en la lista desplegable (convertir a). Las conversiones de y a BCD16 están limitadas al tipo de datos Int. Las conversiones de y a BCD32 están limitadas al tipo de datos DInt.

Tabla 8-108 Estado de ENO

ENO	Descripción	Resultado de OUT
1	No hay error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN
0	El resultado excede el rango válido del tipo de datos de OUT	OUT está puesto al valor de IN

8.7.2 Instrucciones de conversión de SCL

Instrucciones de conversión de SCL

Tabla 8-109 Conversión de un Bool, Byte, Word o DWord

Tipo de datos	Instrucción	Resultado
Bool	**BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT**	El valor se transfiere al bit menos significativo del tipo de datos de destino.
Byte	**BYTE_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	BYTE_TO_WORD, BYTE_TO_DWORD	El valor se transfiere al byte menos significativo del tipo de datos de destino.
	BYTE_TO_SINT, BYTE_TO_USINT	El valor se transfiere al tipo de datos de destino.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	El valor se transfiere al byte menos significativo del tipo de datos de destino.
Word	**WORD_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	WORD_TO_BYTE	El byte menos significativo del valor fuente se transfiere al tipo de datos de destino.
	WORD_TO_DWORD	El valor se transfiere a la palabra menos significativa del tipo de datos de destino.
	WORD_TO_SINT, WORD_TO_USINT	El byte menos significativo del valor fuente se transfiere al tipo de datos de destino.
	WORD_TO_INT, WORD_TO_UINT	El valor se transfiere al tipo de datos de destino.
	WORD_TO_DINT, WORD_TO_UDINT	El valor se transfiere a la palabra menos significativa del tipo de datos de destino.
DWord	**DWORD_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT	El byte menos significativo del valor fuente se transfiere al tipo de datos de destino.
	DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	La palabra menos significativa del valor fuente se transfiere al tipo de datos de destino.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	El valor se transfiere al tipo de datos de destino.

Tabla 8-110 Conversión de un entero corto (SInt o USInt)

Tipo de datos	Instrucción	Resultado
SInt	**SINT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	SINT_TO_BYTE	El valor se transfiere al tipo de datos de destino.
	SINT_TO_WORD, SINT_TO_DWORD	El valor se transfiere al byte menos significativo del tipo de datos de destino.
	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	El valor está convertido.
USInt	**USINT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	USINT_TO_BYTE	El valor se transfiere al tipo de datos de destino.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	El valor se transfiere al byte menos significativo del tipo de datos de destino.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	El valor está convertido.

Tabla 8-111 Conversión de un entero (Int o UInt)

Tipo de datos	Instrucción	Resultado
Int	**INT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	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	El valor está convertido.
	INT_TO_WORD	El valor se transfiere al tipo de datos de destino.
	INT_TO_DINT	El valor se transfiere al byte menos significativo del tipo de datos de destino.
UInt	**UINT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	El valor está convertido.
	UINT_TO_WORD, UINT_TO_DATE	El valor se transfiere al tipo de datos de destino.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	El valor se transfiere al byte menos significativo del tipo de datos de destino.

Tabla 8-112 Conversión de un entero doble (Dint o UDInt)

Tipo de datos	Instrucción	Resultado
DInt	**DINT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	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	El valor está convertido.
	DINT_TO_DWORD, DINT_TO_TIME	El valor se transfiere al tipo de datos de destino.
UDInt	**UDINT_TO_BOOL**	El bit menos significativo se transfiere al tipo de datos de destino.
	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	El valor está convertido.
	UDINT_TO_DWORD, UDINT_TO_TOD	El valor se transfiere al tipo de datos de destino.

Tabla 8-113 Conversión de un número real (Real o LReal)

Tipo de datos	Instrucción	Resultado
Real	**REAL_TO_DWORD, REAL_TO_LREAL**	El valor se transfiere al tipo de datos de destino.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	El valor está convertido.
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**	El valor está convertido.

Tabla 8-114 Conversión de Time, DTL, TOD o Date

Tipo de datos	Instrucción	Resultado
Time	**TIME_TO_DINT**	El valor se transfiere al tipo de datos de destino.
DTL	**DTL_TO_DATE, DTL_TO_TOD**	El valor está convertido.
TOD	**TOD_TO_UDINT**	El valor está convertido.
Date	**DATE_TO_UINT**	El valor está convertido.

Tabla 8-115 Conversión de un Char o String

Tipo de datos	Instrucción	Resultado
Char	**CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT**	El valor está convertido.
	CHAR_TO_STRING	El valor se transfiere al primer carácter de la cadena.

Tipo de datos	Instrucción	Resultado
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`	El valor está convertido.
	`STRING_TO_CHAR`	El primer carácter de la cadena se copia en Char.

8.7.3 ROUND (redondear número) y TRUNC (truncar a entero)

Tabla 8-116 Instrucciones ROUND y TRUNC

KOP / FUP	SCL	Descripción
	`out := ROUND (in);`	Convierte un número real en un entero. Para KOP/FUP, haga clic en "???" en el cuadro de instrucciones para seleccionar el tipo de datos para la salida, por ejemplo "DInt". Para SCL, el tipo de datos predeterminado para la salida de la instrucción ROUND es DINT. Para redondear a otro tipo de datos de salida, introduzca el nombre de la instrucción con el nombre explícito del tipo de datos, por ejemplo ROUND_REAL o ROUND_LREAL. La fracción del número real se redondea al número entero más cercano (IEEE - redondear al número más cercano). Si el número se encuentra exactamente entre dos enteros (p. ej. 10,5), el número se redondeará al entero par. Ejemplo: • ROUND (10.5) = 10 • ROUND (11.5) = 12
	`out := TRUNC(in);`	TRUNC convierte un número real en un entero. La parte fraccionaria del número real se trunca a cero (IEEE - redondear hacia cero).

[1] En KOP y FUP: Haga clic en "???" (junto al nombre de la instrucción) y seleccione un tipo de datos en el menú desplegable.

Tabla 8-117 Tipos de datos para los parámetros

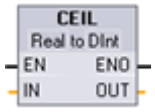
Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Número en coma flotante en la entrada
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Salida redondeada o truncada

Tabla 8-118 Estado de ENO

ENO	Descripción	Resultado de OUT
1	No hay error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN

8.7.4 CEIL y FLOOR (redondear un número en coma flotante al siguiente entero superior o inferior)

Tabla 8-119 Instrucciones CEIL y FLOOR

KOP / FUP	SCL	Descripción
	`out := CEIL(in);`	Convierte un número real (Real o LReal) en el siguiente entero mayor o igual a ese número real (IEEE - redondear hacia el infinito positivo).
	`out := FLOOR(in);`	Convierte un número real (Real o LReal) en el siguiente entero menor o igual a ese número real (IEEE - redondear hacia el infinito negativo).

[1] En KOP y FUP: Haga clic en "???" (junto al nombre de la instrucción) y seleccione un tipo de datos en el menú desplegable.

Tabla 8-120 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Real, LReal	Número en coma flotante en la entrada
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Salida convertida

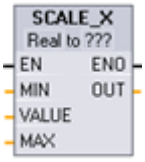
Tabla 8-121 Estado de ENO

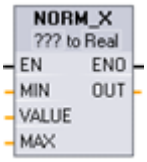
ENO	Descripción	Resultado de OUT
1	No hay error	Resultado válido
0	IN es +/- INF o +/- NaN	+/- INF o +/- NaN

8.7.5 SCALE_X (escalar) y NORM_X (normalizar)

Tabla 8-122 Instrucciones SCALE_X y NORM_X

KOP / FUP	SCL	Descripción

	`out :=SCALE_X(min:=_in_,`
	`value:=_in_,`
	`max:=_in_);`
		Escala el parámetro VALUE real normalizado (donde 0,0 <= VALUE <= 1,0) al tipo de datos y rango de valores especificados por los parámetros MIN y MAX:
		OUT = VALUE (MAX - MIN) + MIN

	`out :=NORM_X(min:=_in_,`
	`value:=_in_,`
	`max:=_in_);`
		Normaliza el parámetro VALUE dentro del rango de valores especificado por los parámetros MIN y MAX:
		OUT = (VALUE - MIN) / (MAX - MIN),
		donde (0,0 <= OUT <= 1,0)

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-123 Tipos de datos para los parámetros

Parámetro	Tipo de datos[1]	Descripción
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Entrada que indica el valor mínimo del rango
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valor de entrada que se debe escalar o normalizar
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Entrada que indica el valor máximo del rango
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Valor de salida escalado o normalizado

[1] Para SCALE_X: Los parámetros MIN, MAX y OUT deben tener el mismo tipo de datos.
Para NORM_X: Los parámetros MIN, VALUE y MAX deben tener el mismo tipo de datos.

Nota

SCALE_X parámetro VALUE debe restringirse a (0,0 <= VALUE <= 1,0)

Si el parámetro VALUE es menos que 0,0 o mayor que 1,0:

- La operación de escala lineal puede producir valores OUT menores que el parámetro MIN o mayores que el valor del parámetro MAX de valores OUT comprendidos dentro del rango de valores del tipo de datos OUT. La ejecución de SCALE_X pone ENO = TRUE para estos casos.
- Es posible generar números escalados no comprendidos en el rango del tipo de datos de OUT. En estos casos, el parámetro OUT se ajusta a un valor intermedio igual a la parte menos significativa del número real escalado antes de la conversión final al tipo de datos de OUT. En ese caso, la ejecución de SCALE_X pone ENO = FALSE.

NORM_X parámetro VALUE debe restringirse a (MIN <= VALUE <= MAX)

Si el parámetro VALUE es menor que MIN o mayor que MAX, la operación de escala lineal puede producir valores OUT normalizados menores que 0,0 o mayores que 1,0. La ejecución de NORM_X pone ENO = TRUE en este caso.

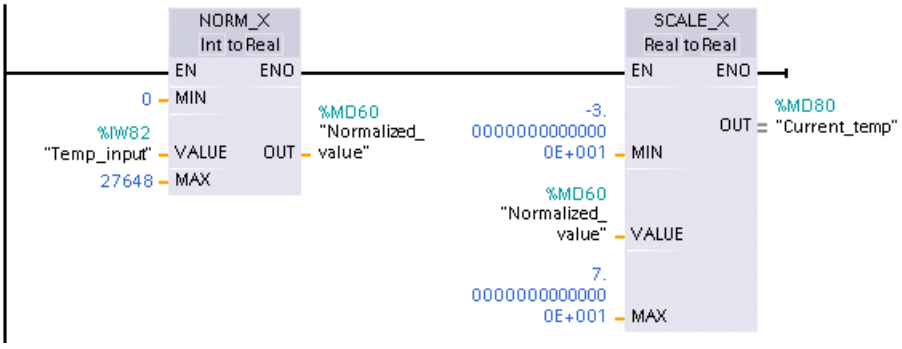
Tabla 8-124 Estado de ENO

ENO	Condición	Resultado de OUT
1	No hay error	Resultado válido
0	El resultado excede el rango válido del tipo de datos de OUT	Resultado intermedio: La parte menos significativa de un número real antes de la conversión final al tipo de datos de OUT.
0	Parámetros MAX <= MIN	SCALE_X: la parte menos significativa del número real VALUE que debe rellenar el tamaño de OUT. NORM_X: VALUE en el tipo de datos VALUE ampliado para rellenar una palabra doble.
0	Parámetro VALUE = +/- INF o +/- NaN	VALUE se escribe en OUT

Ejemplo (KOP): Normalización y escalado de un valor de entrada analógica

Una entrada analógica de un módulo de señales analógicas o Signal Board que usa entrada de intensidad se encuentra en el rango de valores válidos entre 0 y 27648. Suponiendo que una entrada analógica representa una temperatura en la que el valor 0 de la entrada analógica representa -30,0 grados C y 27648 representa 70,0 grados C.

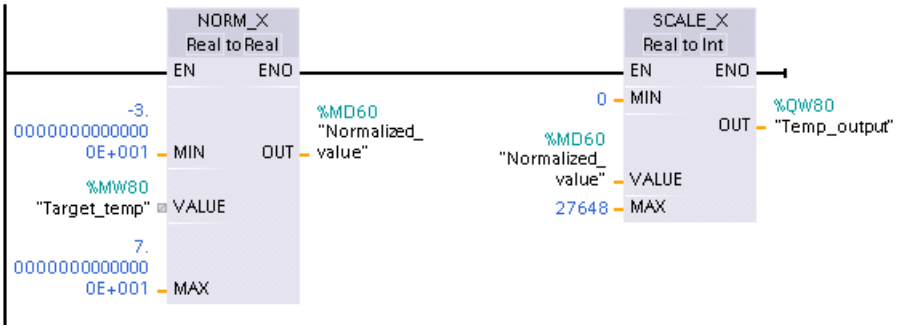
Para transformar el valor analógico en las correspondientes unidades de ingeniería, normalice la entrada a un valor entre 0,0 y 1,0 y a continuación escálelo entre -30,0 y 70,0. El valor resultante es la temperatura representada por la entrada analógica en grados Celsius:



Recuerde que si la entrada analógica provenía de un módulo de señales analógicas o una Signal Board con tensión, el valor MIN para la instrucción NORM_X sería -27648 en lugar de 0.

Ejemplo (KOP): Normalización y escalado de un valor de salida analógico

Una salida analógica de un módulo de señales analógicas o Signal Board que utiliza una salida de intensidad debe estar en el rango de valores válidos entre 0 y 27648. Supongamos que una salida analógica representa un ajuste de temperatura en el que el valor 0 de la entrada analógica representa -30,0 grados Celsius y 27648 representa 70,0 grados Celsius. Para convertir un valor de temperatura guardado que se encuentra entre -30,0 y 70,0 en un valor para la salida analógica dentro del rango entre 0 y 27648, es necesario normalizar el valor correspondiente a unidades de ingeniería en un valor entre 0,0 y 1,0, y a continuación escalarlo al rango de la salida analógica, de 0 a 27648:



Recuerde que si la salida analógica estaba destinada a un módulo de señales analógicas o una Signal Board con tensión, el valor MIN para la instrucción SCALE_X sería -27648 en lugar de 0.

En los Datos técnicos encontrará información adicional sobre representaciones de entradas analógica (Página 1352) y representaciones de salidas analógicas (Página 1353) en intensidad y tensión.

8.7.6 Instrucciones de conversión Variant

8.7.6.1 VARIANT_TO_DB_ANY (convertir VARIANT en DB_ANY)

La instrucción "VARIANT to DB_ANY" se utiliza para leer el operando del parámetro IN y convertirlo en el tipo de datos DB_ANY. El parámetro IN es del tipo de datos Variant y representa un bloque de datos de instancia o un bloque de datos ARRAY. Cuando se crea el programa, no es necesario saber qué bloque de datos se corresponde con el parámetro IN. La instrucción lee el número del bloque de datos en tiempo de ejecución y lo escribe en el operando del parámetro RET_VAL .

Tabla 8-125 Instrucción VARIANT_TO_DB_ANY

KOP / FUP	SCL	Descripción
No disponible	`RET_VAL := VARIANT_TO_DB_ANY( in := _variant_in_, err => _int_out_);`	Lee el operando del parámetro Variant IN y lo guarda en el resultado de la función, que es del tipo DB_ANY

Tabla 8-126 Parámetros de la instrucción VARIANT_TO_DB_ANY

Parámetro	Tipo de datos	Descripción
IN	Variant	Variant que representa un bloque de datos de instancia o un bloque de datos Array
RET_VAL	DB_ANY	Tipo de datos de salida DB_ANY que contiene el número de bloque de datos convertido
ERR	Int	Información de error

Tabla 8-127 Estado de ENO

ENO	Condición	Resultado
1	No hay error	La instrucción convierte la entrada Variant y la guarda en la salida de función DB_ANY
0	La entrada de habilitación EN tiene el estado lógico "0" o el parámetro IN no es válido.	La instrucción no hace nada.

Tabla 8-128 Códigos de salida de error para la instrucción VARIANT_TO_DB_ANY

Err (W#16#...)	Descripción
0000	No hay error
252C	El tipo de datos Variant en el parámetro IN tiene el valor 0. La CPU cambia al estado operativo STOP.
8131	El bloque de datos no existe o es demasiado corto (primer acceso).

Err (W#16#...)	Descripción
8132	El bloque de datos es demasiado corto y no es un bloque de datos Array (segundo acceso).
8134	El bloque de datos está protegido contra escritura
8150	El tipo de datos Variant en el parámetro IN presenta el valor "0". Para recibir este mensaje de error, la propiedad del bloque "Tratamiento local de errores en el bloque" debe estar activada. De lo contrario, la CPU cambia al estado STOP y envía el código de error 16#252C.
8154	El bloque de datos tiene el tipo de datos incorrecto.
* Es posible visualizar códigos de error en el editor de programas como valores enteros o hexadecimales.	

8.7.6.2 DB_ANY_TO_VARIANT (convertir DB_ANY en VARIANT)

La instrucción "DB_ANY to VARIANT" se utiliza para leer el número de un bloque de datos que cumple los requisitos listados a continuación. El operando del parámetro IN tiene el tipo de datos DB_ANY, lo que significa que no es necesario saber durante la creación del programa qué bloque de datos se leerá. La instrucción lee el número del bloque de datos en tiempo de ejecución y lo escribe en el resultado de la función RET_VAL por medio de un puntero VARIANT.

Tabla 8-129 Instrucción DB_ANY_TO_VARIANT

KOP / FUP	SCL	Descripción

No disponible	`RET_VAL := DB_ANY_TO_VARIANT(`
	`  in := _db_any_in_,`
	`  err => _int_out_);`
		Lee el número de bloque de datos del parámetro Variant IN y lo guarda en el resultado de la función, que es del tipo Variant

Tabla 8-130 Parámetros de la instrucción DB_ANY_TO_VARIANT

Parámetro	Tipo de datos	Descripción
IN	DB_ANY	Variante que contiene el número del bloque de datos
RET_VAL	Variant	Tipo de datos de salida DB_ANY que contiene el número de bloque de datos convertido
ERR	Int	Información de error

Tabla 8-131 Estado de ENO

ENO	Condición	Resultado
1	No hay error	La instrucción convierte el número de bloque de datos en Variant y lo guarda en la salida de función DB_ANY
0	La entrada de habilitación EN tiene el estado lógico "0" o el parámetro IN no es válido.	La instrucción no hace nada.

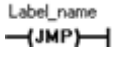
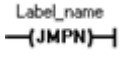
Tabla 8-132 Códigos de salida de error para la instrucción DB_ANY_TO_VARIANT

Err (W#16#...)	Descripción
0000	No hay error
8130	El número del bloque de datos es 0.
8131	El bloque de datos no existe o es demasiado corto.
8132	El bloque de datos es demasiado corto y no es un bloque de datos Array.
8134	El bloque de datos está protegido contra escritura.
8154	El bloque de datos tiene el tipo de datos incorrecto.
8155	Código de tipo desconocido
* Es posible visualizar códigos de error en el editor de programas como valores enteros o hexadecimales.	

8.8 Control del programa

8.8.1 Instrucciones JMP (Saltar si RLO = 1), JMPN (Saltar si RLO = 0) y LABEL (Etiqueta)

Tabla 8-133 Instrucciones JMP, JMPN y LABEL

KOP	FUP	SCL	Descripción
		Véase la instrucción GOTO (Página 316).	Saltar si RLO (resultado de la operación lógica) = 1: Si fluye corriente a una bobina JMP (KOP) o si se cumple el cuadro JMP (FUP), la ejecución del programa continúa con la primera instrucción que le sigue a la etiqueta indicada.
			Saltar si RLO = 0: Si no fluye corriente a una bobina JMPN (KOP) o si la entrada del cuadro JMPN es FALSE (FUP), la ejecución del programa continuará con la primera instrucción que le sigue a la etiqueta indicada.
			Etiqueta de destino de una instrucción de salto JMP o JMPN.

[1] Los nombres de las etiquetas se crean tecleando directamente en la instrucción LABEL. Utilice el icono de ayuda para seleccionar los nombres de etiqueta disponibles para el campo de nombre de etiqueta JMP y JMPN. El nombre de la etiqueta también puede teclearse directamente en la instrucción JMP o JMPN.

Tabla 8-134 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Label_name	Identificador de la etiqueta	Identificador de las instrucciones de salto y etiqueta correspondiente al destino de salto

- Cada etiqueta debe ser única dentro de un bloque lógico.
- Se puede saltar dentro de un bloque lógico, pero no se puede saltar de un bloque lógico a otro.
- Se puede saltar hacia delante o hacia atrás.
- Se puede saltar a la misma etiqueta desde más de un punto del mismo bloque lógico.

8.8.2 JMP_LIST (definir lista de saltos)

Tabla 8-135 Instrucción JMP_LIST

KOP / FUP	SCL	Descripción

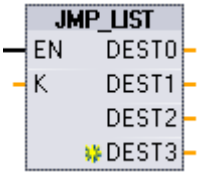
	`CASE k OF`
	`  0: GOTO dest0;`
	`  1: GOTO dest1;`
	`  2: GOTO dest2;`
	`  [n: GOTO destn;]`
	`END_CASE;`
		La instrucción JMP_LIST actúa como un distribuidor de saltos de programa para controlar la ejecución de secciones de un programa. Dependiendo del valor de la entrada K, se produce un salto a la correspondiente etiqueta del programa. La ejecución del programa continúa con las instrucciones que siguen a la etiqueta de destino. Si el valor de la entrada K excede el número de etiquetas - 1, no se produce ningún salto y el procesamiento continúa con el siguiente segmento del programa.

Tabla 8-136 Tipos de datos para parámetros

Parámetro	Tipo de datos	Descripción
K	UInt	Valor de control del distribuidor de saltos
DEST0, DEST1, .., DESTn.	Etiquetas de programa	Etiquetas de destino correspondientes a valores específicos del parámetro K: Si el valor de K es 0, se produce un salto a la etiqueta del programa asignada a la salida DEST0. Si el valor de K es 1, se produce un salto a la etiqueta del programa asignada a la salida DEST1, etc. Si el valor de la entrada K excede el (número de etiquetas - 1), no se produce ningún salto y el procesamiento continúa con el siguiente segmento del programa.

En KOP y FUP: Cuando se coloca el cuadro JMP_LIST en el programa, hay dos salidas de etiqueta. Los destinos de saltos se pueden agregar o borrar.



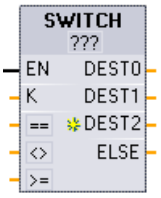
Haga clic en el icono "Crear" dentro del cuadro (en la parte izquierda del último parámetro DEST) para agregar nuevas salidas de etiquetas.



- Haga clic con el botón derecho sobre un conector de salida y seleccione el comando "Insertar salida".
- Haga clic con el botón derecho sobre un conector de salida y seleccione el comando "Borrar".

8.8.3 SWITCH (distribuidor de saltos)

Tabla 8-137 Instrucción SWITCH

KOP / FUP	SCL	Descripción
	`No disponible`	La instrucción SWITCH actúa como un distribuidor de salto de programa para controlar la ejecución de secciones de un programa. Dependiendo del resultado de la comparación entre el valor de la entrada K y los valores asignados a las entradas de comparación específicas, se produce un salto a la etiqueta del programa que corresponde al primer test de comparación que sea TRUE. Si ninguna de las comparaciones es TRUE, se produce un salto asignado a la etiqueta asignada a ELSE. La ejecución del programa continúa con las instrucciones que siguen a la etiqueta de destino.

[1] En KOP y FUP: Haga clic debajo del nombre del cuadro y seleccione un tipo de datos en el menú desplegable.

[2] En SCL: utilice una serie de comparaciones IF-THEN.

Tabla 8-138 Tipos de datos para parámetros

Parámetro	Tipo de datos[1]	Descripción
K	UInt	Entrada de valor de comparación común
==, <>, <, <=, >. >=	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Entradas de valor de comparación separadas para tipos de comparación específicos
DEST0, DEST1, .., DESTn, ELSE	Etiquetas de programa	Etiquetas de destino correspondientes a comparaciones determinadas: Primero se procesa la entrada de comparación bajo la entrada K y siguiente a ella, lo que provoca un salto a la etiqueta asignada a DEST0 si la comparación entre el valor K y esta entrada es TRUE. El siguiente test de comparación utiliza la siguiente entrada situada debajo y provoca un salto a la etiqueta asignada a DEST1 si la comparación es TRUE; las comparaciones restantes se procesan de forma similar y si ninguna de ellas es TRUE, se produce un salto asignado a la etiqueta asignada a la salida ELSE.

[1] La entrada K y las entradas de comparación (==, <>, <, <=, >, >=) deben tener el mismo tipo de datos.

Agregar y borrar entradas y especificar tipos de comparación

Si el cuadro SWITCH de KOP o FUP se coloca en el programa en primer lugar, hay dos entradas de comparación. Se pueden asignar tipos de comparación y agregar entradas/destinos de salto de la forma indicada a continuación.



Haga clic sobre un operador de comparación dentro del cuadro y seleccione un nuevo operador de la lista desplegable.



Haga clic en el icono "Crear" dentro del cuadro (a la izquierda del último parámetro DEST) para agregar nuevos destinos de comparación.



- Haga clic con el botón derecho sobre un conector de entada y seleccione el comando "Insertar entrada".
- Haga clic con el botón derecho en un conector de entrada y seleccione el comando "Borrar".

Tabla 8-139 Selección del tipo de datos del cuadro SWITCH y operaciones de comparación permitidas

Tipo de datos	Comparación	Sintaxis de operadores
Byte, Word, DWord	Igual a	==
	Diferente	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Igual a	==
	Diferente	<>
	Mayor o igual	>=
	Menor o igual	<=
	Mayor	>
	Menor	<

Reglas de posicionamiento del cuadro SWITCH

- No se permite una conexión de instrucción KOP/FUP en frente de la entrada de comparación.
- No hay salida ENO, por lo que se permite una sola instrucción SWITCH en un segmento y debe ser la última operación del segmento.

8.8.4 RET (retroceder)

La instrucción RET opcional sirve para finalizar la ejecución del bloque actual. Solo si fluye corriente a la bobina RET (LAD) o si se cumple la entrada del cuadro RET (FUP), la ejecución del programa del bloque actual finalizará en ese punto y las instrucciones posteriores a la instrucción RET no se ejecutarán. Si el bloque actual es un OB, se ignora el parámetro "Return_Value". Si el bloque actual es una FC o un FB, el valor del parámetro "Return_Value" se devuelve a la rutina que efectúa la llamada como valor ENO del cuadro llamado.

No es necesario introducir manualmente la instrucción RET como última instrucción de un bloque. Esto se efectúa automáticamente. Un solo bloque puede comprender varias instrucciones RET.

Para SCL, véase la instrucción RETURN (Página 316).

Tabla 8-140 Instrucción de control de ejecución "Retroceder" (RET)

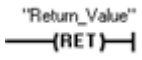
KOP	FUP	SCL	Descripción
		`RETURN;`	Finaliza la ejecución del bloque actual

Tabla 8-141 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
Return_Value	Bool	El parámetro "Return_value" de la instrucción RET se asigna a la salida ENO del cuadro de llamada de bloque en el bloque que efectúa la llamada.

Procedimiento para utilizar la instrucción RET en un bloque lógico FC (ejemplo):

1. Cree un proyecto nuevo y agregue una FC:
2. Edite la FC:
   - Agregue instrucciones del árbol de instrucciones.
   - Agregue una instrucción RET, incluyendo uno de los siguientes valores para el parámetro "Return_Value":
     TRUE, FALSE o una posición de memoria que indique el valor de retorno requerido.
   - Agregue más instrucciones.
3. Llame la FC desde el bloque MAIN [OB1].

La entrada EN del cuadro FC del bloque lógico MAIN se debe cumplir para comenzar la ejecución de la FC.

El valor que indica la instrucción RET en la FC se encontrará en la salida ENO del cuadro FC del bloque lógico tras ejecutarse la FC para la que fluye corriente a la instrucción RET.

8.8.5 ENDIS_PW (limitar y habilitar legitimación de la contraseña)

Tabla 8-142 Instrucción ENDIS_PW

KOP / FUP	SCL	Descripción

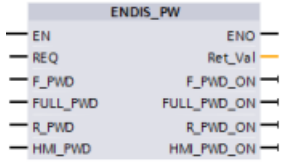
	`ENDIS_PW(`
	`  req:=_bool_in_,`
	`  f_pwd:=_bool_in_,`
	`  full_pwd:=_bool_in_,`
	`  r_pwd:=_bool_in_,`
	`  hmi_pwd:=_bool_in_,`
	`  f_pwd_on=>_bool_out_,`
	`  full_pwd_on=>_bool_out_,`
	`  r_pwd_on=>_bool_out_,`
	`  hmi_pwd_on=>_bool_out_);`
		La instrucción ENDIS_PW puede permitir o no permitir las conexiones de cliente a una CPU S7-1200, aunque el cliente disponga de la contraseña correcta.
		Esta instrucción no inhabilita contraseñas del servidor web.

Tabla 8-143 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Realiza la función si REQ=1
F_PWD	IN	Bool	Contraseña de seguridad positiva: permitir (=1) o no permitir (=0)
FULL_PWD	IN	Bool	Contraseña de acceso completo: permitir (=1) o no permitir (=0) la contraseña de acceso completo
R_PWD	IN	Bool	Contraseña de acceso de lectura: permitir (=1) o no permitir (=0)
HMI_PWD	IN	Bool	Contraseña HMI: permitir (=1) o no permitir (=0)
F_PWD_ON	OUT	Bool	Estados de la contraseña de seguridad positiva: permitida (=1) o no permitida (=0)
FULL_PWD_ON	OUT	Bool	Estado de la contraseña de acceso completo: permitida (=1) o no permitida (=0)
R_PWD_ON	OUT	Bool	Estado de la contraseña de solo lectura: permitida (=1) o no permitida (=0)
HMI_PWD_ON	OUT	Bool	Estado de la contraseña HMI: permitida (=1) o no permitida (=0)
Ret_Val	OUT	Word	Resultado de función

Al llamar ENDIS_PW con REQ=1 no se permiten los tipos de contraseña en los que el parámetro de entrada de la contraseña correspondiente es FALSE. Todo tipo de contraseña puede permitirse o no permitirse de forma independiente. Por ejemplo, si se permite la contraseña de seguridad y el resto no se permite, puede restringir el acceso a la CPU a un pequeño grupo de empleados.

ENDIS_PW se ejecuta de forma síncrona en un ciclo y los parámetros de salida de la contraseña siempre indican el estado actual de la contraseña independientemente del parámetro de entrada REQ. Todas las contraseñas que configure como permisibles deben poder cambiarse a no permitidas/permitidas. De lo contrario, se devolverá un error y se permitirán todas las contraseñas que se permitían antes de ejecutar ENDIS_PW. Esto significa que en una CPU

estándar (que no tenga configurada la contraseña de seguridad) F_PWD debe estar siempre puesto a 1 para que el valor de retorno sea 0. En este caso, F_PWD_ON siempre es 1.

Nota

- La ejecución de ENDIS_PW puede bloquear el acceso de los dispositivos HMI, si no se permite la contraseña para HMI.
- Las sesiones de cliente que se autorizaron antes de que se ejecutara ENDIS_PW pueden finalizarse al ejecutar ENDIS_PW en función del nivel de legitimación existente. Por ejemplo, una conexión legitimada con protección por contraseña de lectura se cancelará con ENDIS_PW (REQ=1, R_PWD=0). Otras conexiones con nivel de protección más bajo también se cancelan en este escenario. En consecuencia, las conexiones legitimadas con acceso completo se mantienen.

Después de un arranque, el acceso a la CPU está restringido por contraseñas previamente definidas en la configuración habitual de protección de la CPU. La posibilidad de desactivar una contraseña válida debe restablecerse ejecutando de nuevo ENDIS_PW. Sin embargo, si ENDIS_PW se ejecuta inmediatamente y se desactivan contraseñas necesarias, es posible que se bloquee el acceso al TIA Portal. Existe la posibilidad de utilizar una instrucción de temporizador para retardar la ejecución de ENDIS_PW y dejar tiempo para introducir las contraseñas antes de que se desactiven.

Nota

Restaurar una CPU que bloquea la comunicación con el TIA Portal

Consulte el tema "Recuperación de una contraseña perdida (Página 135)" para saber cómo borrar la memoria de carga interna de un PLC con una SIMATIC Memory Card.

Si el estado operativo cambia a STOP a causa de errores, ejecución de STP o STEP 7, no se abolirá la protección. La protección es válida hasta que se reinicia la CPU. Consulte la tabla siguiente para más detalles.

Acción	Estado operativo	Control por contraseña ENDIS_PW
Después de resetear la memoria desde STEP 7	STOP	Activo: las contraseñas desactivadas siguen sin permiso.
Después del arranque o de cambiar una SIMATIC Memory Card	STOP	Off: no se desactivan contraseñas.
Tras ejecutar ENDIS_PW en un OB de ciclo o de arranque	STARTUP, RUN	Activo: las contraseñas se desactivan según los parámetros de ENDIS_PW
Después de cambiar el estado operativo de RUN o STARTUP a STOP mediante la instrucción STP, error o STEP 7.	STOP	Activo: las contraseñas desactivadas siguen sin permiso

⚠ ADVERTENCIA

Acceso no autorizado a una CPU protegida

Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU (Página 163) protegidos por contraseña e ID de usuario de servidor web (Página 848) con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.
- Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida.

Tabla 8-144 Códigos de condición

RET_VAL (W#16#...)	Descripción
0000	Sin errores
8090	La instrucción no se soporta.
80D0	No se ha configurado la contraseña de seguridad positiva.
80D1	No se ha configurado la contraseña de acceso de lectura/escritura.
80D2	No se ha configurado la contraseña de acceso de lectura.
80D3	No se ha configurado la contraseña de acceso HMI.

8.8.6 RE_TRIGR (reiniciar tiempo de vigilancia del ciclo)

Tabla 8-145 Instrucción RE_TRIGR

KOP / FUP	SCL	Descripción
	`RE_TRIGR();`	La instrucción RE_TRIGR (Reiniciar la vigilancia del tiempo de ciclo) sirve para prolongar el tiempo máximo permitido antes de que el temporizador de vigilancia del ciclo genere un error.

La instrucción RE_TRIGR se utiliza para reiniciar el temporizador de vigilancia del ciclo durante un único ciclo. De esta manera, el tiempo de ciclo máximo se prolonga un periodo de tiempo de ciclo máximo desde la última ejecución de la función RE_TRIGR.

Nota

Antes de la versión 2.2 del firmware de la CPU S7-1200, RE_TRIGR estaba limitado a la ejecución desde un OB de ciclo de programa y podía utilizarse para prolongar de forma indefinida el tiempo de ciclo del PLC. ENO = FALSE y el temporizador de vigilancia no se inicializa si RE_TRIGR se ejecuta desde un OB de arranque, de alarma o de error.

Para la versión 2.2 y posteriores del firmware, RE_TRIGR puede ejecutarse desde cualquier OB (incluidos OB de arranque, de interrupción y de error). Sin embargo, el ciclo del PLC solo puede prolongarse como máximo 10 veces el tiempo de ciclo máximo configurado.

Ajustar el tiempo de ciclo máximo del PLC

Configure el valor para el tiempo máximo de ciclo en la Configuración de dispositivos para "Tiempo de ciclo".

Tabla 8-146 Valores de tiempo de ciclo

Vigilancia del tiempo de ciclo	Valor mínimo	Valor máximo	Valor predeterminado
Tiempo de ciclo máximo	1 ms	6000 ms	150 ms

Timeout de vigilancia

Si el tiempo de ciclo máximo finaliza antes de haberse completado el ciclo, se generará un error. Si el programa de usuario incluye el OB de alarma de error de tiempo (OB 80), la CPU lo ejecuta, que puede incluir lógica de programa para crear una relación especial.

Si el programa de usuario no incluye un OB de alarma de error de tiempo, la primera condición de tiempo excedido (timeout) se pasa por alto y la CPU permanece en estado operativo RUN. Si ocurre un segundo timeout de tiempo de ciclo máximo en ese mismo ciclo (valor del tiempo de ciclo máximo multiplicado por 2), se disparará un error que provocará el cambio al estado operativo STOP.

En el estado operativo STOP se detiene la ejecución del programa mientras continúan las comunicaciones y los diagnósticos de sistema de la CPU.

8.8.7 STP (finalizar programa)

Tabla 8-147 Instrucción STP

KOP / FUP	SCL	Descripción
	`STP();`	STP pone la CPU en estado STOP. Cuando la CPU está en STOP, se detienen la ejecución del programa de usuario y las actualizaciones físicas desde la memoria imagen de proceso.

Encontrará más información en: Configurar las salidas en una transición de RUN a STOP (Página 98).

Si EN = TRUE, la CPU pasa a estado operativo STOP, se detiene la ejecución del programa y el estado de ENO carece de importancia. De lo contrario, EN = ENO = 0.

8.8.8 Instrucciones GET_ERROR y GET_ERROR_ID (consultar error e ID de error localmente)

Las instrucciones GET_ERROR proporcionan información acerca de errores de ejecución de bloques de programa. Si se ha insertado una instrucción GET_ERROR o GET_ERROR_ID en el bloque lógico, los errores del programa podrán tratarse en el bloque de programa.

GET_ERROR

Tabla 8-148 Instrucción GET_ERROR

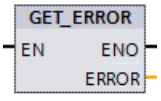
KOP / FUP	SCL	Descripción
	`GET_ERROR(_out_);`	Indica que ha ocurrido un error de ejecución de un bloque y rellena una estructura de datos de error predefinida con información detallada acerca del error.

Tabla 8-149 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
ERROR	ErrorStruct	Estructura de datos del error: Es posible cambiar el nombre de la estructura pero no sus elementos.

Tabla 8-150 Elementos de la estructura de datos ErrorStruct

Componentes de la estructura	Tipo de datos	Descripción
ERROR_ID	Word	ID de error
FLAGS	Byte	Indica si se ha producido un error durante una llamada de bloque. • 16#01: Error durante una llamada de bloque. • 16#00: Ningún error durante una llamada de bloque.

Componentes de la estructura		Tipo de datos	Descripción
REACTION		Byte	Respuesta predeterminada: • 0: Ignorar (error de escritura), • 1: Continuar con el valor de sustitución "0" (error de lectura), • 2: Omitir instrucción (error de sistema)
CODE_ADDRESS		CREF	Información sobre la dirección y el tipo de bloque
	BLOCK_TYPE	Byte	Tipo de bloque en el que ha ocurrido el error: • 1: OB • 2: FC • 3: FB
	CB_NUMBER	UInt	Número del bloque lógico
	OFFSET	UDInt	Referencia a la memoria interna
MODE		Byte	Modo de acceso: Dependiendo del tipo de acceso, puede obtenerse la siguiente información:

Modo	(A)	(B)	(C)	(D)	(E)
0					
1					Offset
2			Área		
3	Ubicación	Alcance		Número	
4			Área		Offset
5			Área	N.° DB	Offset
6	N.° Ptr. / Acc.		Área	N.° DB	Offset
7	N.° Ptr. / Acc.	N.° slot/ Alcance	Área	N.° DB	Offset

OPERAND_NUMBER		UInt	Número de operandos del comando de la máquina
POINTER_NUMBER_ LOCATION		UInt	(A) Puntero interno
SLOT_NUMBER_SCOPE		UInt	(B) Área de almacenamiento dentro de la memoria interna
DATA_ADDRESS		NREF	Información sobre la dirección de un operando
	AREA	Byte	(C) Área de memoria: • 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) Número de bloque de datos
	OFFSET	UDInt	(E) Dirección relativa del operando

GET_ERROR_ID

Tabla 8-151 Instrucción GetErrorID

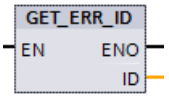
KOP / FUP	SCL	Descripción
	`GET_ERR_ID();`	Indica que ha ocurrido un error de ejecución de bloque de programa y notifica la ID (identificación) del error.

Tabla 8-152 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
ID	Word	Valores de identificador del error del elemento ERROR_ID de ErrorStruct

Tabla 8-153 Valores de Error_ID

ERROR_ID hexadecimal	ERROR_ID decimal	Error de ejecución de bloque de programa
0	0	No hay error
2520	9504	Cadena dañada
2522	9506	Error de lectura de operando fuera de rango
2523	9507	Error de escritura de operando fuera de rango
2524	9508	Error de lectura de área no válida
2525	9509	Error de escritura de área no válida
2528	9512	Error de lectura de alineación de datos (alineación de bit incorrecta)
2529	9513	Error de escritura de alineación de datos (alineación de bit incorrecta)
252C	9516	Error de puntero no inicializado
2530	9520	DB protegido contra escritura
2533	9523	Se ha utilizado un puntero no válido
2538	9528	Error de acceso: el DB no existe
2539	9529	Error de acceso: se ha utilizado un DB erróneo
253A	9530	El DB global no existe
253C	9532	Versión incorrecta o la FC no existe
253D	9533	La instrucción no existe
253E	9534	Versión incorrecta o el FB no existe
253F	9535	La instrucción no existe
2550	9552	Error de acceso: el DB no existe
2575	9589	Error de profundidad de anidamiento del programa
2576	9590	Error de asignación de datos locales
2942	10562	La entrada física no existe
2943	10563	La salida física no existe

Operación

De forma predeterminada, la CPU reacciona a un error de ejecución de bloque registrando un error en el búfer de diagnóstico. No obstante, si se insertan una o más instrucciones GET_ERROR o GET_ERROR_ID en un bloque lógico, este podrá tratar los errores en el bloque. En este caso, la CPU no registra un error en el búfer de diagnóstico. En su lugar, la información de error se deposita en la salida de la instrucción GET_ERROR o GET_ERROR_ID. Es posible leer información de error detallada con la instrucción GET_ERROR, o bien solo el identificador del error con la instrucción GET_ERROR_ID. Normalmente, el primer error es el más importante; los errores siguientes son solo consecuencias del primer error.

La primera ejecución de una instrucción GET_ERROR o GET_ERROR_ID en un bloque devuelve el primer error detectado durante la ejecución del bloque. Este error puede haberse producido en cualquier punto entre el inicio del bloque y la ejecución de GET_ERROR o GET_ERROR_ID. Las siguientes ejecuciones de GET_ERROR o GET_ERROR_ID devuelven el primer error desde la ejecución previa de GET_ERROR o GET_ERROR_ID. El historial de errores no se guarda y la ejecución de la instrucción en cuestión hará que el sistema PLC capture el siguiente error.

El tipo de datos ErrorStruct que utiliza la instrucción GET_ERROR puede agregarse en el editor del bloque de datos y editores de interfaz de bloque, de manera que la lógica del programa pueda acceder a estos valores. Seleccione ErrorStruct en la lista desplegable de tipos de datos para agregar esta estructura. Es posible crear varios ErrorStruct utilizando nombres unívocos. No es posible cambiar el nombre de los elementos de un ErrorStruct.

Condición de error indicada por ENO

Si EN = TRUE y se ejecuta GET_ERROR o GET_ERROR_ID, entonces:

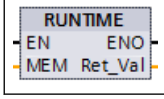
- ENO = TRUE indica que ha ocurrido un error de ejecución del bloque lógico y que hay un error de datos
- ENO = FALSE indica que no ha ocurrido ningún error de ejecución del bloque lógico

Es posible conectar la lógica del programa de reacción a errores a ENO que se activa cuando ocurre un error. Si existe un error, sus datos se almacenan en el parámetro de salida, donde el programa puede acceder a ellos.

GET_ERROR y GET_ERROR_ID pueden utilizarse para enviar información de error desde el bloque que se está ejecutando (bloque llamado) a un bloque invocante. Coloque la instrucción en el último segmento del bloque de programa llamado para notificar el estado de ejecución final del bloque llamado.

8.8.9 RUNTIME (medir tiempo de ejecución)

Tabla 8-154 Instrucción RUNTIME

KOP / FUP	SCL	Descripción
	`Ret_Val := RUNTIME(` `  _lread_inout_);`	Mide el tiempo de ejecución de todo el programa, bloques individuales o secuencias de comandos.

Si desea medir el tiempo de ejecución de todo el programa, llame la instrucción "Medir tiempo de ejecución" en el OB 1. La medición del tiempo de ejecución se inicia con la primera llamada

y la salida RET_VAL devuelve el tiempo de ejecución del programa después de la segunda llamada. El tiempo de ejecución medido incluye todos los procesos de la CPU que pueden ocurrir durante la ejecución del programa, por ejemplo interrupciones causadas eventos de alto nivel o comunicaciones. La instrucción "Medir tiempo de ejecución" lee un contador interno de la CPU y escribe el valor en el parámetro IN-OUT MEM. La instrucción calcula el tiempo de ejecución actual del programa de acuerdo con la frecuencia interna del contador y lo escribe en la salida RET_VAL

Si desea medir el tiempo de ejecución de bloques individuales o secuencias de comandos individuales, necesita tres segmentos separados. Llame la instrucción "Medir tiempo de ejecución" en un segmento individual dentro del programa. El punto de inicio de la medición del tiempo de ejecución se define en esta primera llamada de la instrucción. Seguidamente, se llama el bloque de programa necesario o la secuencia de comandos en el segmento siguiente. En otro segmento se llama la instrucción "Medir tiempo de ejecución" de nuevo y se asigna la misma memoria al parámetro IN-OUT MEM tal y como se ha hecho en la primera llamada de la instrucción. En el tercer segmento, la instrucción "Medir tiempo de ejecución" lee un contador interno de la CPU y calcula el tiempo de ejecución actual del bloque de programa o la secuencia de comandos de acuerdo con la frecuencia interna del contador; a continuación lo escribe en la salida RET_VAL.

La instrucción "Medir tiempo de ejecución" usa un contador interno de alta frecuencia para calcular el tiempo. Si el contador se desborda, la instrucción devuelve valores <= 0,0. Ignore estos valores de tiempo de ejecución.

Nota

La CPU no puede determinar con exactitud el tiempo de ejecución de una secuencia de comandos porque la secuencia de las instrucciones dentro de una secuencia de comandos cambia durante la compilación optimizada del programa.

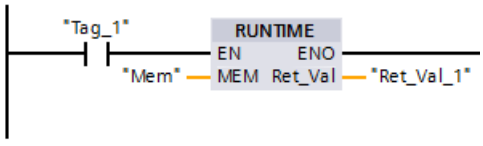
Tabla 8-155 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
MEM	LReal	Punto de inicio de la medición del tiempo de ejecución
RET_VAL	LReal	Tiempo de medición medido en segundos

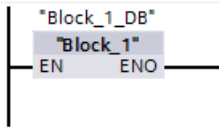
Ejemplo: Instrucción RUNTIME

El ejemplo siguiente muestra el uso de la instrucción RUNTIME para medir el tiempo de ejecución de un bloque de función:

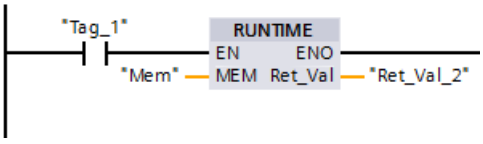
Segmento 1:



Segmento 2:



Segmento 3:



Cuando el operando "Tag_1" del segmento 1 tiene el estado lógico "1", se ejecuta la instrucción RUNTIME. El punto de inicio para la medición del tiempo de ejecución se define con la primera llamada de la instrucción y se respalda como referencia para la segunda llamada de la instrucción en el operando "Mem".

El bloque de función FB1 se ejecuta en el segmento 2.

Cuando el bloque de programa FB1 finaliza y el operando "Tag_1" tiene el estado lógico "1", se ejecuta la instrucción RUNTIME en el segmento 3. La segunda llamada de la instrucción calcula el tiempo de ejecución del bloque de programa y escribe el resultado en la salida RET_VAL_2.

8.8.10 Instrucciones de control del programa de SCL

8.8.10.1 Vista general de las instrucciones de control del programa de SCL

Structured Control Language (SCL) proporciona tres tipos de instrucción de control de programa para estructurar el programa de usuario:

- Instrucciones selectivas: Una instrucción selectiva permite dirigir la ejecución del programa a secuencias de instrucciones alternativas.
- Bucles: La ejecución de bucles puede controlarse utilizando instrucciones de iteración. Una instrucción de iteración especifica qué partes de un programa deberían iterarse en función de determinadas condiciones.
- Saltos del programa: Un salto del programa es un salto inmediato a una meta de salto específica y, con ello, a una instrucción diferente dentro del mismo bloque.

Estas instrucciones de control de programa utilizan la sintaxis del lenguaje de programación PASCAL.

Tabla 8-156 Tipos de instrucciones de control en SCL

Instrucción de control del programa		Descripción
Selectivo	Instrucción IF-THEN (Página 310)	Permite dirigir la ejecución del programa a una de las dos ramas alternativas, en función de que una condición sea TRUE o FALSE
	Instrucción CASE (Página 311)	Permite la ejecución selectiva en 1 de *n* ramas alternativas, en función del valor de una variable

Instrucción de control del programa		Descripción
Bucle	Instrucción FOR (Página 312)	Repite una secuencia de instrucciones mientras la variable de control permanezca dentro del rango de valores especificado
	Instrucción WHILE-DO (Página 313)	Repite una secuencia de instrucciones mientras se siga cumpliendo una condición de ejecución
	Instrucción REPEAT-UNTIL (Página 314)	Repite una secuencia de instrucciones hasta que se cumpla la condición de fin
Salto del programa	Instrucción CONTINUE (Página 315)	Detiene la ejecución de la iteración de bucle actual
	Instrucción EXIT (Página 315)	Abandona un bucle en cualquier momento independientemente de si se cumple o no la condición de fin
	Instrucción GOTO (Página 316)	Provoca un salto inmediato del programa a una etiqueta específica
	Instrucción RETURN (Página 316)	Hace que el programa abandone el bloque que se está ejecutando y regrese al bloque invocante

8.8.10.2 Instrucción IF-THEN

La instrucción IF-THEN es una instrucción condicional que controla el flujo del programa ejecutando una serie de instrucciones basándose en la evaluación de un valor Bool de una expresión lógica. También es posible utilizar paréntesis para anidar o estructurar la ejecución de instrucciones IF-THEN múltiples.

Tabla 8-157 Elementos de la instrucción IF-THEN

SCL	Descripción

`IF "condición" THEN`
`  instrucción_A;`
`  instrucción_B;`
`  instrucción_C;`
`  ;`
	Si "condición" es TRUE o 1, entonces ejecuta las siguientes instrucciones hasta que aparezca la instrucción END_IF.
	Si "condición" es FALSE o 0, salta a la instrucción END_IF (a no ser que el programa incluya instrucciones ELSIF o ELSE adicionales).

`[ELSIF "condición-n" THEN`
`  instrucción_N;`
`  ;]`
	La condición ELSEIF[1] opcional aporta condiciones adicionales que deben evaluarse. Ejemplo: Si "condición" de la instrucción IF-THEN es FALSE, entonces el programa evalúa "condición-n". Si "condición-n" es TRUE, ejecuta "instrucción_N".

`[ELSE`
`  instrucción_X;`
`  ;]`
	La instrucción ELSE opcional aporta instrucciones que deben ejecutarse si la "condición" de la instrucción IF-THEN es FALSE.

`END_IF;`	La instrucción END_IF finaliza la instrucción IF-THEN.

[1] Se pueden incluir varias instrucciones ELSIF dentro de la instrucción IF-THEN.

Tabla 8-158 Variables de la instrucción IF-THEN

Variables	Descripción
"condición"	Requerida. La expresión lógica puede ser TRUE (1) o FALSE (0).
"instrucción_A"	Opcional. Una o más instrucciones deben ejecutarse si "condición" es TRUE.
"condición-n"	Opcional. La expresión lógica que debe ser evaluada por la instrucción opcional ELSIF.

Variables	Descripción
"instrucción_N"	Opcional. Una o más instrucciones deben ejecutarse si "condición-n" de la instrucción ELSIF es TRUE.
"instrucción_X"	Opcional. Una o más instrucciones deben ejecutarse si "condición" de la instrucción IF-THEN es TRUE.

Las instrucciones IF se ejecutan de acuerdo con las siguientes normas:

- Se ejecuta la primera secuencia de las instrucciones cuya expresión lógica es TRUE. Las secuencias restantes de las instrucciones no se ejecutan.
- Si ninguna expresión booleana es TRUE, se ejecuta la secuencia de instrucciones correspondiente a ELSE (o ninguna secuencia si la rama ELSE no existe).
- Puede haber un número ilimitado de instrucciones ELSIF.

Nota

El uso de una o más ramas tiene la ventaja de que las expresiones lógicas que siguen a una expresión válida ya no se evalúan, al contrario que una secuencia de una instrucción IF. De ese modo es posible reducir el tiempo de ejecución de un programa.

8.8.10.3 Instrucción CASE

Tabla 8-159 Elementos de la instrucción CASE

SCL	Descripción

`CASE "Valor_test" OF`
`  "ListaValores": Instrucción[; Instrucción, ...]`
`  "ListaValores": Instrucción[; Instrucción, ...]`
`[ELSE`
`Instrucción Else[; Instrucción Else, ...]]`
`END_CASE;`
	La instrucción CASE ejecuta uno de varios grupos de instrucciones en función del valor de una expresión.

Tabla 8-160 Parámetros

Parámetro	Descripción
"Valor_Test"	Requerida. Cualquier expresión numérica del tipo de datos Int
"ListaValores"	Requerida. Un valor único o una lista de valores o rangos de valores separados por coma. (Utilice dos periodos para definir un rango de valores: 2..8) El siguiente ejemplo ilustra las diferentes variantes de la lista de valores: 1: Instrucción_A; 2, 4: Instrucción _B; 3, 5..7,9: Instrucción _C;
Instrucción	Requerida. Se ejecutan una o más instrucciones cuando "Valor_test" coincide con cualquier valor de la lista de valores
Instrucción Else	Opcional. Una o más instrucciones que se ejecutan si no hay ninguna concordancia con un valor de "ListaValores"

La instrucción CASE se ejecuta de acuerdo con las siguientes normas:

- La expresión Test_value debe devolver un valor del tipo Int.
- Si se procesa una instrucción CASE, el programa comprueba si el valor de la expresión de Test_value está contenido dentro de una lista de valores determinada. Si se encuentra una coincidencia, se ejecuta el componente de la instrucción asignado a la lista.
- Si no se encuentra ninguna coincidencia, se ejecuta la sección del programa posterior a ELSE o no se ejecuta ninguna instrucción si la rama ELSE no existe.

Ejemplo: instruccionesCASE anidadas

Las instrucciones CASE pueden anidarse. Cada instrucción CASE anidada debe tener asociada una instrucción END_CASE.

```
CASE "var1" OF
       1 : #var2 := 'A';
       2 : #var2 := 'B';
ELSE
       CASE "var3" OF

          65..90: #var2 := 'UpperCase';
          97..122: #var2 := 'LowerCase';
       ELSE

          #var2:= 'SpecialCharacter';
       END_CASE;
END_CASE;
```

8.8.10.4 Instrucción FOR

Tabla 8-161 Elementos de la instrucción FOR

SCL	Descripción

`FOR "variable_control" := "inicio" TO "fin"`
`[BY "incremento"] DO`
`  instrucción;`
`  ;`
`END_FOR;`
	Una instrucción FOR se utiliza para repetir una secuencia de instrucciones mientras la variable de control se encuentre dentro del rango de valores especificado. La definición de un bucle con FOR incluye la especificación de un valor inicial y otro final. Ambos valores deben ser del mismo tipo de datos que la variable de control.
	Los bucles FOR se pueden anidar. La instrucción END_FOR se refiere a la última instrucción FOR ejecutada.

Tabla 8-162 Parámetros

Parámetro	Descripción
"variable_control"	Requerida. Un entero (Int o DInt) que sirve como contador de bucles
"inicio"	Requerida. Expresión simple que especifica el valor inicial de las variables de control

Parámetro	Descripción
"fin"	Requerida. Expresión simple que determina el valor final de las variables de control
"Incremento"	Opcional. Cantidad con la que una "variable de control" incrementa después de cada bucle. El "incremento" debe tener el mismo tipo de datos que la "variable de control". Si el valor de "incremento" no está especificado, el valor de las variables de ejecución se incrementará en 1 después de cada bucle. No es posible cambiar el "incremento" mientras se ejecuta la instrucción FOR.

La instrucción FOR se ejecuta de la siguiente manera:

- Al comienzo del bucle, la variable de control se pone al valor inicial (asignación inicial) y cada vez que el bucle itera se incrementa con el incremento especificado (incremento positivo) o se decrementa (incremento negativo) hasta que se alcanza el valor final.
- Después de la ejecución de cada bucle se comprueba la condición (alcanzado valor final) para saber si se cumple o no. Si no se satisface la condición de fin, se volverá a ejecutar la secuencia de instrucciones, de lo contrario el bucle finalizará y continuará la ejecución con la instrucción inmediatamente después del bucle.

Reglas para la formulación de instrucciones FOR:

- La variable de control solo puede ser del tipo de datos Int o DInt.
- Se puede omitir la instrucción BY [incremento]. Si no se especifica ningún incremento, se asume automáticamente que es +1.

Para finalizar el bucle independientemente del estado de la expresión "condición" utilice la Instrucción EXIT (Página 315). La instrucción EXIT ejecuta la instrucción que sigue inmediatamente a la instrucción END_FOR.

Utilice la Instrucción CONTINUE (Página 315) para omitir las siguientes instrucciones de un bucle FOR y continuar con la comprobación de si se cumple o no la condición de fin.

8.8.10.5 Instrucción WHILE-DO

Tabla 8-163 Instrucción WHILE

SCL	Descripción

`WHILE "condición" DO`
`  Instrucción;`
`  Instrucción;`
`  ...;`
`END_WHILE;`
	La instrucción WHILE realiza una serie de instrucciones hasta que una condición determinada es TRUE.
	Los bucles WHILE se pueden anidar. La instrucción END_WHILE se refiere a la última instrucción WHILE ejecutada.

Tabla 8-164 Parámetros

Parámetro	Descripción
"condición"	Requerida. Una expresión lógica que evalúa si el estado es TRUE o FALSE. (Una condición "null" se interpreta como FALSE.)
Instrucción	Opcional. Una o más instrucciones que se ejecutan hasta que la comprobación de la condición sea TRUE.

Nota

La instrucción WHILE evalúa el estado de "condición" antes de ejecutar cualquier instrucción. Para ejecutar las instrucciones como mínimo una vez independientemente del estado de "condición" utilice la instrucción REPEAT (Página 314).

La instrucción WHILE se ejecuta de acuerdo con las siguientes normas:

- Antes de cada iteración del cuerpo del bucle se evalúa la condición de ejecución.
- El cuerpo del bucle que sigue a DO va iterando mientras la condición de ejecución tenga el valor TRUE.
- Cuando se da el valor FALSE, el bucle se salta y se ejecuta la instrucción que lo sigue.

Para finalizar el bucle independientemente del estado de la expresión "condición" utilice la Instrucción EXIT (Página 315). La instrucción EXIT ejecuta la instrucción que viene justo detrás de END_WHILE.

Utilice la instrucción CONTINUE para saltar las siguientes instrucciones de un bucle WHILE y continuar con la comprobación de si se cumple o no la condición de fin.

8.8.10.6 Instrucción REPEAT-UNTIL

Tabla 8-165 Instrucción REPEAT

SCL	Descripción
REPEAT **Instrucción;** **;** **UNTIL "condición"** **END_REPEAT;**	La instrucción REPEAT ejecuta una serie de instrucciones hasta que una condición determinada es TRUE. Los bucles REPEAT se pueden anidar. La instrucción END_REPEAT se refiere a la última instrucción REPEAT ejecutada.

Tabla 8-166 Parámetros

Parámetro	Descripción
Instrucción	Opcional. Una o más instrucciones que se ejecutan hasta que la condición sea TRUE.
"condition"	Requerida. Una o más expresiones del siguiente modo: Una expresión numérica o de cadena que evalúa si el estado es TRUE o FALSE. Una condición "null" se interpreta como FALSE.

Nota

Antes de evaluar el estado de "condición", la instrucción REPEAT ejecuta las instrucciones durante la primera iteración del bucle (incluso si "condición" es FALSE). Para comprobar el estado de "condición" antes de ejecutar las instrucciones utilice la instrucción WHILE (Página 313).

Para finalizar el bucle independientemente del estado de la expresión "condición" utilice la Instrucción EXIT (Página 315). La instrucción EXIT ejecuta la instrucción que viene justo detrás de END_REPEAT.

Utilice la Instrucción CONTINUE (Página 315) para omitir las siguientes instrucciones de un bucle REPEAT y continuar con la comprobación de si se cumple o no la condición de fin.

8.8.10.7 Instrucción CONTINUE

Tabla 8-167 Instrucción CONTINUE

SCL	Descripción

`CONTINUE`
`  Instrucción;`
`  ;`
	La instrucción CONTINUE salta las instrucciones siguientes de un bucle de programa (FOR, WHILE, REPEAT) y continúa el bucle comprobando si se da la condición de fin. Si no es así, el bucle continúa.

La instrucción CONTINUE se ejecuta de acuerdo con las siguientes normas:

- Esta instrucción finaliza inmediatamente la ejecución de un cuerpo de bucle.
- Dependiendo de si la condición de repetición del bucle se cumple o no, el bucle vuelve a ejecutarse o se interrumpe la iteración y se ejecuta la instrucción inmediatamente posterior.
- En una instrucción FOR la variable de control se incrementa con el incremento especificado inmediatamente después de una instrucción CONTINUE.

Utilice la instrucción CONTINUE solo dentro de un bucle. En los bucles anidados, CONTINUE solo hace referencia al bucle que lo incluye inmediatamente. Por regla general, CONTINUE se usa junto con una instrucción IF.

Si el bucle debe persistir independientemente de la comprobación de fin, utilice la instrucción EXIT.

Ejemplo: instrucción CONTINUE

El siguiente ejemplo muestra el uso de la instrucción CONTINUE para evitar un error de división entre 0 al calcular el porcentaje de un valor:

```
FOR i := 0 TO 10 DO
IF valor[i] = 0 THEN CONTINUE; END_IF;
    p := parte / valor[i] * 100;
    s := INT_TO_STRING(p);
    porcentaje := CONCAT(IN1:=s, IN2:="%");
END_FOR;
```

8.8.10.8 Instrucción EXIT

Tabla 8-168 Instrucción EXIT

SCL	Descripción
`EXIT;`	La instrucción EXIT se utiliza para finalizar un bucle (FOR, WHILE o REPEAT) en cualquier punto, independientemente de si se cumple o no la condición de fin.

La instrucción EXIT se ejecuta de acuerdo con las siguientes normas:

- Esta instrucción provoca la finalización inmediata de la instrucción de repetición situada directamente junto a la instrucción EXIT.
- La ejecución del programa continúa tras la interrupción del bucle (por ejemplo después de END_FOR).

Utilice la instrucción EXIT dentro de un bucle. En bucles anidados, la instrucción EXIT devuelve el procesamiento al siguiente nivel de anidamiento superior.

Ejemplo: instrucción EXIT

```
FOR i := 0 TO 10 DO
CASE valor[i, 0] OF
    1..10: valor [i, 1]:="A";
    11..40: valor [i, 1]:="B";
    41..100: valor [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

8.8.10.9 Instrucción GOTO

Tabla 8-169 Instrucción GOTO

SCL	Descripción

`GOTO JumpLabel;`
`instrucción;`
`... ;`
`JumpLabel: instrucción;`
	La instrucción GOTO omite instrucciones y salta a una etiqueta del mismo bloque.
	La etiqueta ("JumpLabel") y la instrucción GOTO deben estar en el mismo bloque. El nombre de una etiqueta solo puede asignarse una vez dentro de un bloque. Cada etiqueta puede ser el destino de varias instrucciones GOTO.

No es posible saltar a una sección de bucle (FOR, WHILE o REPEAT). Sí que es posible saltar desde dentro de un bucle.

Ejemplo: instrucción GOTO

En el ejemplo siguiente: en función del valor del operando "Tag_value", la ejecución del programa se reanuda en el punto definido por la etiqueta correspondiente. Si "Tag_value" = 2, la ejecución del programa se reanuda en la etiqueta "MyLabel2" y omite "MyLabel1".

```
CASE "Tag_value" OF
1 : GOTO MiEtiqueta1;
2 : GOTO MiEtiqueta2;
ELSE GOTO MiEtiqueta3;
END_CASE;
MiEtiqueta1: "Tag_1" := 1;
MiEtiqueta2: "Tag_2" := 1;
MiEtiqueta3: "Tag_4" := 1;
```

8.8.10.10 Instrucción RETURN

Tabla 8-170 Instrucción RETURN

SCL	Descripción
`RETURN;`	La instrucción RETURN abandona sin condiciones el bloque lógico que se está ejecutando. La ejecución del programa regresa al bloque invocante o al sistema operativo (si abandona un OB).

Ejemplo: instrucción RETURN:

```
IF "Error" <> 0 THEN
RETURN;
```

```
END_IF;
```

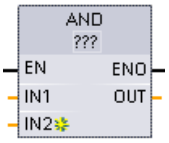
Nota

Una vez ejecutada la última instrucción, el bloque lógico regresa automáticamente al bloque invocante. No inserte una instrucción RETURN al final de un bloque lógico.

8.9 Operaciones lógicas con palabras

8.9.1 Instrucciones de operaciones lógicas AND, OR y XOR

Tabla 8-171 Instrucciones de operaciones lógicas AND, OR y XOR

KOP / FUP	SCL	Descripción
	`out := in1 AND in2;`	Y: Y lógica
	`out := in1 OR in2;`	O: O lógica
	`out := in1 XOR in2;`	XOR: O-EXCLUSIVA lógica

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.



Para agregar una entrada, haga clic en el icono "Crear" o haga clic con el botón derecho del ratón en el conector de entrada del parámetro IN existente y seleccione el comando "Insertar entrada".

Para quitar una entrada, haga clic con el botón derecho del ratón en el conector de entrada de uno de los parámetros IN existentes (si hay más entradas además de las dos originales) y seleccione el comando "Borrar".

Tabla 8-172 Tipos de datos para los parámetros

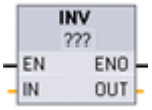
Parámetro	Tipo de datos	Descripción
IN1, IN2	Byte, Word, DWord	Entradas lógicas
OUT	Byte, Word, DWord	Salida lógica

[1] La selección del tipo de datos ajusta los parámetros IN1, IN2 y OUT a un mismo tipo de datos.

Los valores de bit correspondientes de IN1 y IN2 se combinan para producir un resultado lógico binario en el parámetro OUT. ENO es siempre TRUE (verdadero) tras ejecutarse estas instrucciones.

8.9.2 INV (complemento a 1)

Tabla 8-173 Instrucción INV

KOP / FUP	SCL	Descripción
	`No disponible`	Calcula el complemento binario a uno del parámetro IN. El complemento a uno se obtiene invirtiendo cada valor de bit del parámetro IN (cambiando cada 0 a 1 y cada 1 a 0). ENO es siempre TRUE (verdadero) tras ejecutarse esta instrucción.

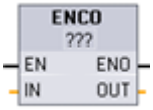
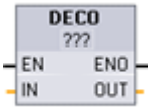
[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-174 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Elemento que debe invertirse
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Byte, Word, DWord	Salida invertida

8.9.3 Instrucciones DECO (Descodificar) y ENCO (Codificar)

Tabla 8-175 Instrucciones ENCO y DECO

KOP / FUP	SCL	Descripción
	`out := ENCO(_in_);`	Codifica un patrón de bits en un número binario La instrucción ENCO convierte el parámetro IN al número binario correspondiente al bit activado menos significativo del parámetro IN y deposita el resultado en el parámetro OUT. Si el parámetro IN es 0000 0001 o 0000 0000, el valor 0 se deposita en el parámetro OUT. Si el valor del parámetro IN es 0000 0000, ENO adopta el estado lógico FALSE.
	`out := DECO(_in_);`	Descodifica un número binario a un patrón de bits La instrucción DECO descodifica un número binario del parámetro IN poniendo a "1" el bit correspondiente en el parámetro OUT (todos los demás bits se ponen a 0). ENO es siempre TRUE (verdadero) tras ejecutarse la instrucción DECO. Nota: El tipo de datos predeterminado para la instrucción DECO es DWORD. En SCL, cambie el nombre de instrucción a DECO_BYTE o DECO_WORD descodificar un valor de byte o de palabra y asígnelo a una dirección o variable de byte o palabra.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-176 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Patrón de bits que debe codificarse DECO: Valor que debe descodificarse
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Valor codificado DECO: Patrón de bits descodificado

Tabla 8-177 Estado de ENO

ENO	Condición	Resultado (OUT)
1	No hay error	Número de bit válido
0	IN es cero	OUT se pone a cero

La selección del tipo de datos del parámetro OUT de DECO como Byte, Word o DWord delimita el rango útil del parámetro IN. Si el valor del parámetro IN supera el rango útil, se realiza una operación modulo para obtener los bits menos significativos, tal y como se indica a continuación.

Rango del parámetro IN de DECO:

- 3 bits (valores 0-7) IN se utilizan para activar 1 posición de bit en Byte OUT
- 4 bits (valores 0-15) IN se utilizan para activar 1 posición de bit en Word OUT
- 5 bits (valores 0-31) IN se utilizan para activar 1 posición de bit en DWord OUT

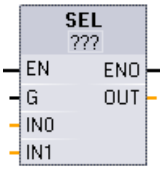
Tabla 8-178 Ejemplos

Valor IN de DECO			Valor OUT de DECO (descodificar un bit)
Byte OUT 8 bits	Mín. IN	0	00000001
	Máx. IN	7	10000000
Word OUT 16 bits	Mín. IN	0	0000000000000001
	Máx. IN	15	1000000000000000
DWord OUT 32 bits	Mín. IN	0	00000000000000000000000000000001
	Máx. IN	31	10000000000000000000000000000000

8.9.4 Instrucciones SEL (Seleccionar), MUX (Multiplexar) y DEMUX (Desmultiplexar)

Tabla 8-179 Instrucción SEL (seleccionar)

KOP / FUP	SCL	Descripción

	`out := SEL(`
	`  g:=_bool_in,`
	`  in0:-_variant_in,`
	`  in1:=_variant_in);`
		La instrucción SEL (Seleccionar) asigna uno de dos valores de entrada al parámetro OUT, dependiendo del valor del parámetro G.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.

Tabla 8-180 Tipos de datos para la instrucción SEL

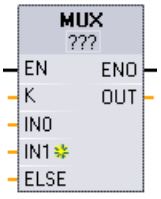
Parámetro	Tipo de datos[1]	Descripción
G	Bool	• 0 selecciona IN0 • 1 selecciona IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entradas
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Salida

[1] Las variables de entrada y la de salida deben tener el mismo tipo de datos.

Códigos de condición: ENO siempre es TRUE tras la ejecución de la instrucción SEL.

Tabla 8-181 Instrucción MUX (multiplexar)

KOP / FUP	SCL	Descripción

	`out := MUX(`
	`  k:=_unit_in,`
	`  in1:=variant_in,`
	`  in2:=variant_in,`
	`  [...in32:=variant_in ,]`
	`  inelse:=variant_in);`
		La instrucción MUX copia uno de varios valores de entrada en el parámetro OUT, dependiendo del valor del parámetro K. Si el valor del parámetro K supera (IN*n* - 1), el valor del parámetro ELSE se copia en el parámetro OUT.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.



Para agregar una entrada, haga clic en el icono "Crear" o haga clic con el botón derecho del ratón en el conector de entrada del parámetro IN existente y seleccione el comando "Insertar entrada".

Para quitar una entrada, haga clic con el botón derecho del ratón en el conector de entrada de uno de los parámetros IN existentes (si hay más entradas además de las dos originales) y seleccione el comando "Borrar".

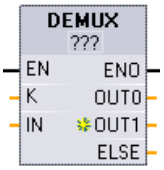
Tabla 8-182 Tipos de datos para la instrucción MUX

Parámetro	Tipo de datos	Descripción
K	UInt	• 0 selecciona IN1 • 1 selecciona IN2 • *n* selecciona IN*n*
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entradas
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entrada del valor sustitutivo (opcional)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Salida

[1] Las variables de entrada y la de salida deben tener el mismo tipo de datos.

Tabla 8-183 Instrucción DEMUX (desmultiplexar)

KOP / FUP	SCL	Descripción

	`DEMUX(`
	`k:=_unit_in,`
	`in:=variant_in,`
	`out1:=variant_in,`
	`out2:=variant_in,`
	`[...out32:=variant_in,]`
	`outelse:=variant_in)`
	`;`
		DEMUX copia el valor del punto asignado al parámetro IN en una o varias salidas. El valor del parámetro K selecciona la salida asignada como destino del valor IN. Si el valor de K es mayor que el número (OUT*n* - 1) el valor IN se copia en el punto asignado al parámetro ELSE.

[1] En KOP y FUP: haga clic en "???" y seleccione un tipo de datos en el menú desplegable.



Para agregar una salida, haga clic en el icono "Crear" o haga clic con el botón derecho del ratón en un conector de salida del parámetro OUT existente y seleccione el comando "Insertar salida".

Para eliminar una salida, haga clic con el botón derecho del ratón en el conector de salida de uno de los parámetros OUT existentes (si hay más salidas además de las dos originales) y seleccione el comando "Borrar".

Tabla 8-184 Tipos de datos para la instrucción DEMUX

Parámetro	Tipo de datos[1]	Descripción
K	UInt	Valor selector: • 0 selecciona OUT1 • 1 selecciona OUT2 • n selecciona OUTn
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Entrada
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Salidas
ELSE	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Sustituir salida si K es mayor que (OUTn - 1)

[1] La variable de entrada y las de salida deben tener el mismo tipo de datos.

Tabla 8-185 Estado de ENO para las instrucciones MUX y DEMUX

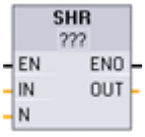
ENO	Condición	Resultado de OUT
1	No hay error	MUX: el valor de IN seleccionado se copia en OUT DEMUX: el valor de IN se copia en el parámetro OUT seleccionado
0	MUX: K es mayor que el número de entradas -1	• No hay ELSE: OUT permanece invariable, • Hay ELSE, valor de ELSE asignado a OUT
	DEMUX: K es mayor que el número de salidas -1	• No hay ELSE: Las salidas permanecen invariables, • Hay ELSE, el valor de IN se copia en ELSE

8.10 Desplazamiento y rotación

8.10.1 Instrucciones SHR (Desplazar a la derecha) y SHL (Desplazar a la izquierda)

Tabla 8-186 Instrucciones SHR y SHL

KOP / FUP	SCL	Descripción

	`out := SHR(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
	`out := SHL(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
		Utilice las instrucciones de desplazamiento (SHL y SHR) para desplazar el patrón de bits del parámetro IN. El resultado se asigna al parámetro OUT. El parámetro N especifica el número de posiciones de bit desplazadas:
		• SHR: Desplazar patrón de bits hacia la derecha
		• SHL: Desplazar patrón de bits hacia la izquierda

[1] En KOP y FUP: haga clic en "???" y seleccione los tipos de datos del menú desplegable.

Tabla 8-187 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Enteros	Patrón de bits que debe desplazarse
N	USInt, UDint	Número de bits que deben desplazarse
OUT	Enteros	Patrón de bits después del desplazamiento

- Con N=0 no hay desplazamiento. El valor IN se asigna a OUT.
- Los ceros se desplazan a los bits que quedan vacíos tras el desplazamiento.

- Si el número de posiciones que deben desplazarse (N) excede el número de bits en el valor de destino (8 para Byte, 16 para Word, 32 para DWord), todos los valores de bit originales se desplazarán hacia fuera y se reemplazarán por ceros (cero se asigna a OUT).
- ENO es siempre TRUE (verdadero) en las operaciones de desplazamiento.

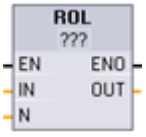
Tabla 8-188 Ejemplo: SHL para datos del tipo Word (palabra)

Desplazar los bits de una palabra a la izquierda insertando ceros en la derecha (N = 1)			
IN	1110 0010 1010 1101	Valor de OUT antes del primer desplazamiento:	1110 0010 1010 1101
		Después del primer desplazamiento a la izquierda:	1100 0101 0101 1010
		Después del segundo desplazamiento a la izquierda:	1000 1010 1011 0100
		Después del tercer desplazamiento a la izquierda:	0001 0101 0110 1000

8.10.2 Instrucciones ROR (Rotar a la derecha) y ROL (Rotar a la izquierda)

Tabla 8-189 Instrucciones ROR y ROL

KOP / FUP	SCL	Descripción

	`out := ROL(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
	`out := ROR(`
	`  in:=_variant_in_,`
	`  n:=_uint_in);`
		Utilice las instrucciones de rotación (ROR y ROL) para rotar el patrón de bits del parámetro IN. El resultado se deposita en el parámetro OUT. El parámetro N define el número de bits rotados.
		• ROR: rotar patrón de bits hacia la derecha
		• ROL: rotar patrón de bits hacia la izquierda

[1] En KOP y FUP: haga clic en "???" y seleccione los tipos de datos del menú desplegable.

Tabla 8-190 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	Enteros	Patrón de bits que debe rotarse
N	USInt, UDint	Número de bits que deben rotarse
OUT	Enteros	Patrón de bits después de la rotación

- Con N=0 no hay rotación. El valor IN se asigna a OUT.
- Los bits rotados hacia fuera en un lado del valor de destino se rotan hacia el otro lado, por lo que no se pierden los valores de bit originales.
- Si el número de bits que deben rotarse (N) excede el número de bits en el valor de destino (8 para Byte, 16 para Word, 32 para DWord), la rotación se efectúa de todas maneras.
- ENO es siempre TRUE (verdadero) tras ejecutarse las instrucciones de rotación.

Tabla 8-191 Ejemplo: ROR para datos del tipo Word (palabra)

Rotar bits desde la derecha a la izquierda (N = 1)			
IN	0100 0000 0000 0001	Valor de OUT antes de la primera rotación:	0100 0000 0000 0001
		Después de la primera rotación a la derecha:	1010 0000 0000 0000
		Después de la segunda rotación a la derecha:	0101 0000 0000 0000

Instrucciones avanzadas

9

9.1 Funciones de fecha, hora y reloj

9.1.1 Instrucciones de fecha y hora

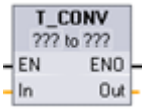
Utilice las instrucciones de fecha y hora para realizar cálculos de calendario y hora.

- T_CONV convierte un valor a o desde (tipos de datos de fecha y hora) y (tipos de datos de byte, palabra y palabra doble).
- T_ADD suma los valores de Time y DTL: (Time + Time = Time) o (DTL + Time = DTL)
- T_SUB resta los valores de Time y DTL: (Time - Time = Time) o (DTL - Time = DTL)
- T_DIFF da la diferencia entre dos valores de DTL en forma de valor de Time: DTL - DTL = Time
- T_COMBINE combina un valor Date y un valor Time_and_Date para crear un valor DTL.

Para más información sobre el formato de los datos DTL y Time, consulte el apartado sobre los tipos de datos Fecha y hora (Página 109).

Tabla 9-1 Instrucción T_CONV (Convertir y extraer tiempos)

KOP / FUP	Ejemplo de SCL	Descripción

	`out := DINT_TO_TIME( in:=_variant_in);`
	
	`out := TIME_TO_DINT( in:=_variant_in);`
		T_CONV convierte un valor a o desde (tipos de datos de fecha y hora) y (tipos de datos de byte, palabra y palabra doble).

[1] En cuadros KOP y FUP: Haga clic en "???" y seleccione los tipos de datos de origen/destino en el menú desplegable.

[2] En SCL: Arrastre T_CONV del árbol de instrucciones y suéltelo en el editor del programa y, a continuación, seleccione los tipos de datos de origen/destino.

Tabla 9-2 Tipos de datos válidos para las conversiones de T_CONV

Tipo de datos IN (u OUT)	Tipos de datos OUT (o IN)
TIME (milisegundos)	DInt, Int, SInt, UDInt, UInt, USInt, TOD Solo para SCL: Byte, Word, Dword
DATE (número de días desde el 1 de enero de 1990)	DInt, Int, SInt, UDInt, UInt, USInt, DTL Solo para SCL: Byte, Word, Dword
TOD (milisegundos desde medianoche: 24:00:00,000)	DInt, Int, SInt, UDInt, UInt, USInt, TIME, DTL Solo para SCL: Byte, Word, Dword

Nota

Uso de T_CONV para convertir un tamaño de datos mayor en un tamaño de datos menor

Los valores de datos se pueden truncar cuando convierte un tipo de datos mayor con más bytes en un tipo de datos menor con menos bytes. Si esto se produce, ENO se pone a 0.

Conversión a/de tipo de datos DTL

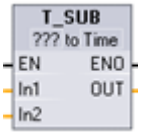
DTL (Date and Time Long) contiene los datos de año, mes, fecha y hora. Los datos de DTL se pueden convertir a/de tipos de datos DATE y TOD.
Sin embargo, la conversión de DTL con los datos de DATE solo afecta a los valores de año, mes y día. La conversión de DTL con datos de TOD solo afecta a los valores de hora, minutos y segundos.

Cuando T_CONV convierte a DTL, los elementos de datos no afectados en el formato DTL no cambiarán.

Tabla 9-3 Instrucciones T_ADD (Sumar tiempos) y T_SUB (Restar tiempos)

KOP / FUP	SCL	Descripción

	`out := T_ADD(`
	`in1:=_variant_in,`
	`in2:=_time_in);`
		T_ADD suma el valor de la entrada IN1 (tipos de datos DTL o Time) con el valor de la entrada IN2 Time. El parámetro OUT proporciona el resultado DTL o Time. Son posibles dos operaciones con estos tipos de datos, a saber:
		• Time + Time = Time
		• DTL + Time = DTL

	`out := T_SUB(`
	`in1:=_variant_in,`
	`in2:=_time_in);`
		T_SUB resta el valor IN2 Time del valor IN1 (DTL o Time). En el parámetro OUT se deposita el valor de diferencia como tipo de datos DTL o Time. Son posibles operaciones de dos tipos de datos.
		• Time - Time = Time
		• DTL - Time = DTL

[1] En KOP y FUP: haga clic en "???" y seleccione los tipos de datos del menú desplegable.

Tabla 9-4 Tipos de datos para los parámetros T_ADD y T_SUB

Parámetro y tipo		Tipo de datos	Descripción
IN1[1]	IN	DTL, Time	Valor DTL o Time
IN2	IN	Time	Valor Time que debe sumarse o restarse
OUT	OUT	DTL, Time	DTL o suma o diferencia Time

[1] Seleccione el tipo de datos de IN1 en la lista desplegable debajo del nombre de la instrucción. La selección del tipo de datos de IN1 ajusta también el tipo de datos del parámetro OUT.

Tabla 9-5 Instrucción T_DIFF (Diferencia de tiempo)

KOP / FUP	SCL	Descripción

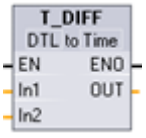
	`out := T_DIFF(`
	`in1:=_DTL_in,`
	`in2:=_DTL_in);`
		T_DIFF resta el valor DTL (IN2) del valor DTL (IN1). En el parámetro OUT se deposita el valor de diferencia como tipo de datos Time.
		• DTL - DTL = Time

Tabla 9-6 Tipos de datos para los parámetros T_DIFF

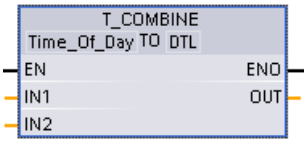
Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	DTL	Valor DTL
IN2	IN	DTL	Valor DTL que se debe restar
OUT	OUT	Time	Diferencia Time

Códigos de condición: ENO = 1 significa que no se ha producido ningún error. ENO = 0 y el parámetro OUT = 0 significan errores:

- El valor DTL no es válido
- El valor Time no es válido

Tabla 9-7 Instrucción T_COMBINE (Combinar tiempos)

KOP / FUP	SCL	Descripción

	`out :=`
	`CONCAT_DATE_TOD(`
	`In1 := _date_in,`
	`In2 := _tod_in);`
		T_COMBINE combina un valor Date y un valor Time_of_Day para crear un valor DTL.

[1] Observe que la instrucción T_COMBINE de las Instrucciones avanzadas se corresponde con la función CONCAT_DATE_TOD en SCL.

Tabla 9-8 Tipos de datos para los parámetros T_COMBINE

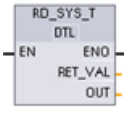
Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	Date	El valor Date que debe combinarse debe estar entre DATE#1990-01-01 y DATE#2089-12-31
IN2	IN	Time_of_Day	Valores Time_of_Day que deben combinarse
OUT	OUT	DTL	Valor DTL

9.1.2 Funciones de reloj

> **⚠ ADVERTENCIA**
>
> **Riesgo de ataques a las redes mediante la sincronización del Network Time Protocol (NTP)**
>
> Si un atacante puede acceder a las redes por sincronización del Network Time Protocol (NTP), posiblemente alterará el control del proceso cambiando la hora del sistema de la CPU. Las alteraciones en el control del proceso pueden causar la muerte, heridas graves o daños materiales.
>
> La función de cliente NTP de la CPU S7-1200 está desactivada por defecto y, si se activa, solo permite a las direcciones IP configuradas actuar como servidor NTP. La CPU desactiva esta función por defecto, y hay que configurar esta función para permitir correcciones de la hora del sistema de la CPU por control remoto.
>
> La CPU S7-1200 soporta alarmas horarias e instrucciones de reloj que dependen de la precisión de la hora del sistema de la CPU. Si se configura NTP y se acepta la sincronización horaria desde un servidor, hay que asegurarse de que el servidor es una fuente fiable. Si no se comprueba, se puede producir una brecha de seguridad que permita a un usuario desconocido alterar el control del proceso cambiando la hora del sistema de la CPU.
>
> Para más información sobre seguridad y recomendaciones, consulte nuestras "Guías operacionales sobre seguridad industrial" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) en la página de atención al cliente de Siemens.

Las instrucciones de reloj se utilizan para ajustar y leer el reloj del sistema de la CPU. El tipo de datos DTL (Página 109) se utiliza para proporcionar valores de fecha y hora.

Tabla 9-9 Instrucciones para la hora del sistema

KOP / FUP	SCL	Descripción
	`ret_val := WR_SYS_T( in:=_DTL_in_);`	WR_SYS_T (Ajustar la hora) ajusta el reloj de la CPU con un valor DTL en el parámetro IN. Este valor de hora no incluye la diferencia con respecto a la hora local ni tampoco al horario de verano.
	`ret_val := RD_SYS_T( out=>_DTL_out);`	RD_SYS_T (leer la hora) lee la hora actual del sistema de la CPU. Este valor de hora no incluye la diferencia con respecto a la hora local ni tampoco al horario de verano.

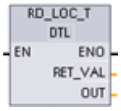
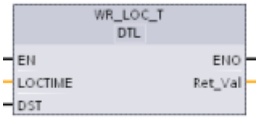
KOP / FUP	SCL	Descripción
	`ret_val := RD_LOC_T( out=>_DTL_out);`	RD_LOC_T (leer hora local) da la hora local actual de la CPU como tipo de datos DTL. Este valor de tiempo refleja la zona horaria local ajustada correctamente según el cambio de horario de verano/ invierno (si está configurado).
	`ret_val := WR_LOC_T( LOCTIME:=DTL_in_, DST:_in_;`	WR_LOC_T (Escribir hora local) establece la fecha y hora del reloj de la CPU. El usuario asigna la información de fecha y hora como hora local en LOCTIME con el tipo de datos de DTL. La instrucción utiliza la estructura de DB "TimeTransformationRule (Página 330)" para calcular la hora del sistema. La granularidad de la información horaria para la hora local y del sistema es específica del producto y es, como mínimo, de un milisegundo. Los valores de entrada en el parámetro LOCTIME que son menores que los admitidos por la CPU se redondean durante el cálculo de la hora del sistema. **Nota**: debe usar la configuración de dispositivos de la CPU para establecer las propiedades de "Hora" (zona horaria, activación de DST, inicio de DST y parada de DST). De lo contrario, WR_LOC_T no puede interpretar el cambio de hora de DST.

Tabla 9-10 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	DTL	Hora que debe ajustarse en el reloj del sistema de la CPU
OUT	OUT	DTL	RD_SYS_T: hora de sistema actual de la CPU RD_LOC_T: hora local actual, incluido cualquier ajuste para el horario de verano, si está configurado
LOCTIME	IN	DTL	WR_LOC_T: hora local
DST	IN	BOOL	WR_LOC_T: **D**aylight **S**aving **T**ime solo se evalúa durante la "hora doble" cuando el reloj cambia al horario de verano. • TRUE = horario de verano (primera hora) • FALSE = horario de invierno (segunda hora)
RET_VAL	OUT	Int	Código de condición de ejecución

- La hora local se calcula aplicando la zona horaria y las variaciones correspondientes al horario de verano/invierno ajustadas en los parámetros de la ficha de configuración del dispositivo "Hora".
- La configuración de la zona horaria es un offset respecto a la hora UTC o GTM.
- La configuración del horario de verano especifica el mes, semana, día y hora de comienzo del horario de verano.

- La configuración del horario estándar especifica el mes, semana, día y hora de comienzo del horario estándar.
- La diferencia con respecto a la zona horaria se aplica siempre al valor de hora del sistema. La diferencia con respecto al horario de verano se aplica únicamente si el horario de verano está en vigor.

Nota

Horario de invierno/verano y configuración del tiempo de inicio estándar

La propiedad "Hora del día" para el "Inicio del horario de verano" de la configuración de dispositivo de la CPU debe ser la hora local.

Códigos de condición: ENO = 1 significa que no se ha producido ningún error. ENO = 0 significa que ha ocurrido un error de ejecución. El código de condición se indica en la salida RET_VAL.

RET_VAL (W#16#....)	Descripción
0000	La hora local actual está en horario estándar.
0001	Se ha configurado el horario de verano y la hora local actual está en horario de verano.
8080	La hora local no está disponible o el valor LOCTIME no es válido.
8081	El valor de año no es válido o el valor de hora asignado por el parámetro LOCTIME no es válido.
8082	Valor de mes no válido (byte 2 en formato DTL)
8083	Valor de día no válido (byte 3 en formato DTL)
8084	Valor de hora no válido (byte 5 en formato DTL)
8085	Valor de minuto no válido (byte 6 en formato DTL)
8086	Valor de segundo no válido (byte 7 en formato DTL)
8087	Valor de nanosegundo no válido (bytes 8 a 11 en formato DTL)
8089	El valor de hora no existe (la hora ya ha pasado al cambiar al horario de verano)
80B0	Ha fallado el reloj de tiempo real
80B1	La estructura "TimeTransformationRule" no se ha definido.

9.1.3 Estructura de datos TimeTransformationRule

Descripción

Las reglas de cambio para el horario de invierno y el de verano se definen en la estructura TimeTransformationRule. Descripción de la estructura:

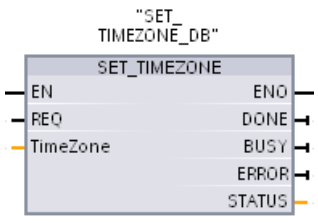
Nombre		Tipo de datos	Descripción
TimeTransformationRule		STRUCT	
	Bias	INT	Diferencia horaria entre hora local y UTC [minutos] Rango: -1439 a 1439
	DaylightBias	INT	Diferencia horaria entre horario de verano y horario de invierno [minutos] Rango: 0 a 60
	DaylightStartMonth	USINT	Mes de cambio a horario de verano Rango: 1 a 12

Nombre		Tipo de datos	Descripción
	DaylightStartWeek	USINT	Semana de cambio a horario de verano 1 = Primera aparición del día de la semana en el mes 5 = Última aparición del día de la semana en el mes
	DaylightStartWeekday	USINT	Día de la semana del cambio a horario de verano: 1 = domingo
	DaylightStartHour	USINT	Hora del cambio a horario de verano: Rango: 0 a 23
	DaylightStartMinute	USINT	Minuto del cambio a horario de verano Rango: 0 a 59
	StandardStartMonth	USINT	Mes del cambio a horario de invierno Rango: 1 a 12
	StandardStartWeek	USINT	Semana del cambio a horario de invierno 1 = Primera aparición del día de la semana en el mes 5 = Última aparición del día de la semana en el mes
	StandardStartWeekday	USINT	Día de la semana del cambio de horario de invierno: 1 = domingo
	StandardStartHour	USINT	Hora de la semana del cambio de horario de invierno Rango: 0 a 23
	StandardStartMinute	USINT	Minuto de la semana del cambio de horario de invierno Rango: 0 a 59
	TimeZoneName	STRING[80]	Nombre de la zona horaria: "(GMT+01:00) Ámsterdam, Berlín, Berna, Roma, Estocolmo, Viena"

9.1.4 SET_TIMEZONE (ajustar zona horaria)

Tabla 9-11 Instrucción SET_TIMEZONE

KOP / FUP	SCL	Descripción

	`"SET_TIMEZONE_DB"(`
	`REQ:=_bool_in,`
	`Timezone:=_struct_in,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		Ajusta la zona horaria local y los parámetros de horario de invierno/verano utilizados para convertir la hora del sistema de la CPU en hora local.

[1] En el ejemplo SCL, "SET_TIMEZONE_DB" es el nombre del DB de instancia.

Tabla 9-12 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	REQ=1: Ejecutar función
Timezone	IN	TimeTransformationRule	Reglas para la transformación de la hora del sistema en hora local
DONE	OUT	Bool	Función completada
BUSY	OUT	Bool	Función ocupada
ERROR	OUT	Bool	Error detectado
STATUS	OUT	Word	Resultado de la función / mensaje de error

Para configurar manualmente los parámetros de zona horaria en la CPU, utilice las propiedades "Hora del día" de la ficha "General" de la configuración del dispositivo.

Use la instrucción SET_TIMEZONE para ajustar la configuración de la hora local. Los parámetros de la estructura "TimeTransformationRule (Página 330)" asignan la zona horaria local y el momento de cambio automático entre el horario de invierno y verano.

Nota

Efecto de la instrucción SET_TIMEZONE en la memoria flash

La instrucción SET_TIMEZONE realiza operaciones de escritura en la memoria flash (memoria de carga interna o Memory Card). Para evitar que se reduzca la vida útil de la memoria flash, utilice la instrucción SET_TIMEZONE para actualizaciones poco frecuentes.

Códigos de condición: ENO = 1 significa que no se ha producido ningún error. ENO = 0 significa que ha ocurrido un error de ejecución. El código de condición se indica en la salida STATUS.

STATUS (W#16#....)	Descripción
0	No hay error
7000	No se está procesando ninguna tarea
7001	Comienza el procesamiento de la tarea. Parámetro BUSY = 1, DONE = 0
7002	Llamada intermedia (REQ irrelevante): Instrucción ya activa; BUSY tiene el valor "1".
808x	Error en el componente x-th: Por ejemplo 8084 indica que DaylightStartWeekif no es un valor de 1 a 5.

9.1.5 RTM (contador de horas de funcionamiento)

Tabla 9-13 Instrucción RTM

KOP / FUP	SCL	Descripción

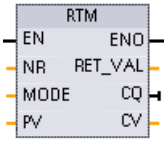
	`RTM(NR:=_uint_in_,`
	`MODE:=_byte_in_,`
	`PV:=_dint_in_,`
	`CQ=>_bool_out_,`
	`CV=>_dint_out_);`
		La instrucción RTM (Contador de horas de funcionamiento) puede inicializar, arrancar, parar y leer los contadores de horas de funcionamiento de la CPU.

Tabla 9-14 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
NR	IN	UInt	Número de contadores de horas de funcionamiento: (valores posibles: 0..9)
MODE	IN	Byte	RTM Número de modo de ejecución: • 0 = Recuperar datos (el estado se escribe entonces en CQ y el valor actual en CV) • 1 = Arrancar (en el último valor del contador) • 2 = Stop • 4 = Inicializar (al valor especificado en PV) • 5 = Inicializar (al valor especificado en PV) y después arrancar • 6 = Inicializar (al valor especificado en PV) y después parar • 7 = Guardar todos los valores RTM de la CPU en la MC (Memory Card)
PV	IN	DInt	Predeterminar valor de horas para el contador de horas de funcionamiento especificado
RET_VAL	OUT	Int	Resultado de la función / mensaje de error
CQ	OUT	Bool	Estado del contador de horas de funcionamiento (1 = contando)
CV	OUT	DInt	Valor actual de horas de funcionamiento para el contador especificado

La CPU utiliza un máximo de 10 contadores de horas de funcionamiento para conocer las horas de funcionamiento de subsistemas de control críticos. Hay que arrancar cada uno de los contadores individuales con una ejecución RTM para cada temporizador. Todos los contadores de horas de funcionamiento se paran cuando la CPU hace una transición de RUN a STOP. También es posible parar los temporizadores individualmente con una ejecución de RTM, modo 2.

Cuando la CPU hace una transición de STOP a RUN, es necesario reiniciar los temporizadores de horas con una ejecución de RTM para cada temporizador que se arranca. Una vez que el valor de un contador de horas de funcionamiento supera las 2147483647 horas, el contaje se detiene y se transmite el error "Desbordamiento". Hay que ejecutar la instrucción RTM una vez por cada temporizador que se reinicia, o bien modificar el temporizador.

Un fallo de alimentación de la CPU o una desconexión y reconexión provoca un proceso de cierre (power down), que guarda los valores actuales de horas de funcionamiento en una memoria remanente. Cuando la CPU vuelve a arrancar, los valores de horas de funcionamiento guardados se vuelven a cargar en los temporizadores, y los totales anteriores de horas de funcionamiento no se pierden. Los contadores de horas de funcionamiento deben reiniciarse para acumular horas de funcionamiento adicionales.

El programa también puede utilizar la ejecución de RTM en modo 7 para guardar los valores de contador de horas de funcionamiento en una Memory Card. Los estados de todos los temporizadores en el momento en que RTM se ejecuta en modo 7 se guardan en la Memory Card. Los valores memorizados pueden ser incorrectos con el tiempo, ya que los contadores de funcionamiento se arrancan y detienen durante una sesión de funcionamiento del programa. Los valores de la Memory Card deben actualizarse periódicamente para capturar eventos importantes en runtime. La ventaja que se obtiene de guardar los valores RTM en la Memory Card radica en que se puede insertar la Memory Card en una CPU de sustitución, en la que estarán disponibles los valores del programa y los valores RTM memorizados. Si los valores de

RTM no se guardaron en la Memory Card, los valores de temporizador se perderán (en una CPU de sustitución).

Nota

Debe evitarse un número excesivo de llamadas del programa para operaciones de escritura en Memory Card

Minimice las operaciones de escritura en Memory Card flash para ampliar la vida útil de las Memory Card.

Tabla 9-15 Códigos de condición

RET_VAL (W#16#....)	Descripción
0	No hay error
8080	Número de contador de horas de funcionamiento incorrecto
8081	Un valor negativo ha pasado al parámetro PV
8082	Desbordamiento del contador de horas de funcionamiento
8091	El parámetro de entrada MODE contiene un valor no válido
80B1	El valor no puede guardarse en la MC (MODE=7)

9.2 Cadena y carácter

9.2.1 Sinopsis del tipo de datos String

Tipo de datos STRING

Los datos String se almacenan como encabezado de 2 bytes seguido de 254 bytes de caracteres en código ASCII. Un encabezado String contiene dos longitudes. El primer byte contiene la longitud máxima que se indica entre corchetes cuando se inicializa una cadena o 254 (ajuste predeterminado). El segundo byte del encabezado es la longitud actual, es decir, el número de caracteres válidos de la cadena. La longitud actual debe ser menor o igual a la longitud máxima. El número de bytes almacenados que ocupa el formato String es 2 bytes mayor que la longitud máxima.

Inicialización de los datos String

Los datos de entrada y salida String deben inicializarse como cadenas válidas en la memoria antes de ejecutar cualquier instrucción con cadenas.

Datos String válidos

Una cadena válida tiene una longitud máxima que debe ser mayor que cero pero menor que 255. La longitud actual debe ser menor o igual a la longitud máxima.

Las cadenas no pueden asignarse a áreas de memoria I ni Q.

Encontrará más información en: Formato del tipo de datos String (Página 112).

9.2.2 S_MOVE (desplazar cadena de caracteres)

Tabla 9-16 Instrucción Desplazar cadena de caracteres

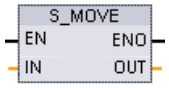
KOP / FUP	SCL	Descripción
	`out := in;`	Copiar la cadena IN fuente en la posición OUT. La instrucción S_MOVE no afecta a los contenidos de la cadena fuente.

Tabla 9-17 Tipos de datos para los parámetros

Parámetro	Tipo de datos	Descripción
IN	String	Cadena fuente
OUT	String	Dirección de destino

Si la longitud real de la cadena en la entrada IN excede la longitud máxima de una cadena guardada en la salida OUT, se copia la parte de la cadena IN que cabe en la cadena OUT.

9.2.3 Instrucciones de conversión de cadenas

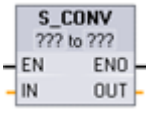
9.2.3.1 Instrucciones S_CONV, STRG_VAL y VAL_STRG (Convertir a/de cadena de caracteres y número)

Las siguientes instrucciones permiten convertir cadenas de caracteres numéricos en valores numéricos, y viceversa:

- S_CONV convierte una cadena numérica en un valor numérico, o viceversa.
- STRG_VAL convierte una cadena numérica en un valor numérico con opciones de formato.
- VAL_STRG convierte un valor numérico en una cadena numérica con opciones de formato.

S_CONV (convertir cadena de caracteres)

Tabla 9-18 Instrucciones de conversión de cadenas

KOP / FUP	SCL	Descripción
	`out := <Type>_TO_<Type>(in);`	Convierte una cadena de caracteres en el valor correspondiente o viceversa. La instrucción S_CONV no tiene opciones de formato de salida. Gracias a ello, la instrucción S_CONV es más simple pero menos flexible que las instrucciones STRG_VAL y VAL_STRG.

[1] En KOP y FUP: haga clic en "???" y seleccione el tipo de datos en la lista desplegable.

[2] En SCL: seleccione S_CONV en las Instrucciones avanzadas y responda las preguntas acerca de los tipos de datos para la conversión. Seguidamente, STEP 7 proporciona la instrucción de conversión adecuada.

Tabla 9-19 Tipos de datos (cadena en valor)

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Entrada que contiene la cadena de caracteres
OUT	OUT	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valor numérico resultante

La conversión del parámetro IN que contiene la cadena comienza en el primer carácter y continúa hasta el final de la cadena, o bien hasta que se encuentra el primer carácter que no sea "0" a "9", "+", "-" o ".". El resultado se deposita en la ubicación que indica el parámetro OUT. Si el valor numérico resultante no está comprendido en el rango del tipo de datos de OUT, el parámetro OUT se pone a 0 y ENO adopta el estado lógico FALSE (falso). De lo contrario, el parámetro OUT contendrá un resultado válido y ENO adoptará el estado lógico TRUE (verdadero).

Reglas de formato de las cadenas de entrada:

- Si se utiliza un punto decimal en la cadena IN, es preciso utilizar el carácter ".".
- Las comas "," utilizadas como separadores de miles a la izquierda del punto decimal están permitidas, aunque se ignoran.
- Los espacios iniciales se ignoran.

S_CONV (convertir cadena de caracteres)

Tabla 9-20 Tipos de datos (valor en cadena)

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Entrada que contiene el valor numérico
OUT	OUT	String, WString	Cadena de caracteres resultante

Un valor entero, entero sin signo o en coma flotante de la entrada IN se convierte en una cadena de caracteres correspondiente que se deposita en OUT. El parámetro OUT debe referenciar una cadena válida antes de ejecutarse la conversión. Una cadena válida consta de una longitud de cadena máxima en el primer byte, la longitud de cadena actual en el segundo byte y los

caracteres de la cadena actuales en los siguientes bytes. La cadena convertida sustituye los caracteres de la cadena OUT comenzando en el primer carácter y ajusta el byte de longitud actual de la cadena OUT. El byte de longitud máxima de la cadena OUT no se modifica.

El número de caracteres que se sustituyen depende del tipo de datos del parámetro IN y del valor numérico. El número de caracteres sustituidos no debe exceder la longitud de la cadena depositada en el parámetro OUT. La longitud máxima (primer byte) de la cadena OUT debe ser mayor o igual al número máximo esperado de caracteres convertidos. En la tabla siguiente se muestran ejemplos de conversión de valor en cadena con S_CONV:

Reglas de formato de las cadenas de salida:

- Los valores que se escriben en el parámetro OUT aparecen sin signo "+" inicial.
- Se utiliza la representación en coma fija (no la notación exponencial).
- El carácter de punto "." se utiliza para representar el punto decimal si el parámetro IN tiene el tipo de datos Real.
- Los valores se justifican a la derecha en la cadena de salida y van precedidos por caracteres de espacio que ocupan las posiciones de caracteres vacías.

Tabla 9-21 Longitudes de cadena máximas para cada tipo de datos

Tipo de datos IN	Posiciones de caracteres asignadas por S_CONV	Ejemplo de cadena convertida[1]	Longitud total de la cadena incluyendo los bytes de longitud máxima y actual
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] Los caracteres "x" representan los caracteres de espacio que ocupan las posiciones vacías en el campo justificado a la derecha que se ha asignado para el valor convertido.

STRG_VAL (convertir cadena de caracteres en valor numérico)

Tabla 9-22 Instrucción de conversión de cadena en valor

KOP / FUP	SCL	Descripción

	`"STRG_VAL"(`
	`in:=_string_in,`
	`format:=_word_in,`
	`p:=uint_in,`
	`out=>_variant_out);`
		Convierte una cadena de caracteres numéricos en un número entero o en coma flotante correspondiente.

[1] En KOP y FUP: haga clic en "???" y seleccione el tipo de datos en la lista desplegable.

Tabla 9-23 Tipos de datos para la instrucción STRG_VAL

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Cadena de caracteres ASCII que debe convertirse
FORMAT	IN	Word	Opciones de formato de salida
P	IN	UInt, Byte, USInt	IN: Índice al primer carácter que debe convertirse (primer carácter = 1)
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valor numérico convertido

La conversión comienza en el offset de carácter P de la cadena IN y continúa hasta su final, o bien hasta que se encuentra el primer carácter que no sea "+", "-", ".", ",", "e", "E" o "0" a "9". El resultado se deposita en la posición que indica el parámetro OUT.

Los datos String deben inicializarse antes de la ejecución como cadena válida en la memoria.

El parámetro FORMAT de la instrucción STRG_VAL se define a continuación. Los bits no utilizados deben ponerse a cero.

Tabla 9-24 Formato de la instrucción STRG_VAL

Bit 16							Bit 8	Bit 7							Bit 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

f = Formato de notación 1= Notación exponencial
0 = Notación en coma fija

r = Formato de punto decimal 1 = "," (coma)
0 = "." (período)

Tabla 9-25 Valores del parámetro FORMAT

FORMAT (W#16#)	Formato de notación	Representación del punto decimal
0000 (estándar)	Coma fija	"."
0001		","
0002	Exponencial	"."
0003		","
0004 a FFFF	Valores no válidos	

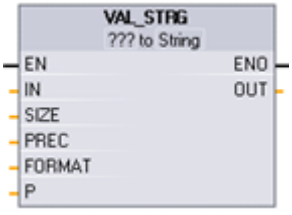
Reglas para la conversión de STRG_VAL:

- Si el carácter de punto "." se utiliza como punto decimal, las comas "," a la izquierda del punto decimal se interpretan como separadores de miles. Las comas están permitidas, aunque se ignoran.
- Si el carácter de coma "," se utiliza como punto decimal, los puntos "." a la izquierda del punto decimal se interpretan como separadores de miles. Estos puntos están permitidos, aunque se ignoran.
- Los espacios iniciales se ignoran.

VAL_STRG (convertir valor numérico en cadena de caracteres)

Tabla 9-26 Operación de conversión de valor en cadena

KOP / FUP	SCL	Descripción

	`"VAL_STRG"(`
	`in:=_variant_in,`
	`size:=_usint_in,`
	`prec:=_usint_in,`
	`format:=_word_in,`
	`p:=uint_in,`
	`out=>_string_out);`
		Convierte un valor entero, entero sin signo o en coma flotante en la cadena de caracteres correspondiente.

[1] En KOP y FUP: haga clic en "???" y seleccione el tipo de datos en la lista desplegable.

Tabla 9-27 Tipos de datos para la instrucción VAL_STRG

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Valor que debe convertirse
SIZE	IN	USInt	Número de caracteres que deben escribirse en la cadena OUT
PREC	IN	USInt	Precisión o tamaño de la parte fraccionaria. Esto no incluye el punto decimal.
FORMAT	IN	Word	Opciones de formato de salida
P	IN	UInt, Byte, USInt	IN: Índice al primer carácter de la cadena OUT que debe convertirse (primer carácter = 1)
OUT	OUT	String, WString	Cadena convertida

Esta instrucción convierte el valor que indica el parámetro IN en una cadena referenciada por el parámetro OUT. El parámetro OUT debe ser una cadena válida antes de ejecutar la conversión.

La cadena convertida sustituye los caracteres de la cadena OUT, comenzando en el contaje de offset de carácter P hasta el número de caracteres que indica el parámetro SIZE. El número de caracteres de SIZE debe caber en la longitud de la cadena OUT, contando desde la posición del carácter P. Si el parámetro SIZE es cero, los caracteres se sobrescriben sin limitación en la posición P de la cadena OUT. Esta instrucción resulta útil para integrar caracteres numéricos en una cadena de texto. Por ejemplo, la cifra "120" puede incorporarse a la cadena "Presión bomba = 120 psi".

El parámetro PREC indica la precisión o el número de dígitos de la parte fraccionaria de la cadena. Si el valor del parámetro IN es un número entero, PREC indicará la posición del punto decimal. Por ejemplo, si el valor de datos es 123 y PREC = 1, el resultado será "12.3". La precisión máxima soportada para el tipo de datos Real es 7 dígitos.

Si el parámetro P es mayor que el tamaño actual de la cadena OUT, se agregan espacios hasta la posición P y el resultado se añade al final de la cadena. La conversión finaliza cuando se alcanza la longitud máxima de la cadena OUT.

El parámetro FORMAT de la instrucción VAL_STRG se define a continuación. Los bits no utilizados deben ponerse a cero.

Tabla 9-28 Formato de la instrucción VAL_STRG

Bit 16							**Bit 8**	**Bit 7**							**Bit 0**
0	0	0	0	0	0	0	0	0	0	0	0	0	s	f	r

s = Carácter de signo	1= usar los signos "+" y "-" 0 = usar solo el signo "-"
f = Formato de notación	1= Notación exponencial 0 = Notación en coma fija
r = Formato de punto decimal	1 = "," (coma) 0 = "." (período)

Tabla 9-29 Valores del parámetro FORMAT

FORMAT (WORD)	**Carácter de signo**	**Formato de notación**	**Representación del punto decimal**
W#16#0000	Solo "-"	Coma fija	"."
W#16#0001			","
W#16#0002		Exponencial	"."
W#16#0003			","

FORMAT (WORD)	Carácter de signo	Formato de notación	Representación del punto decimal
W#16#0004	"+" y "-"	Coma fija	"."
W#16#0005			","
W#16#0006		Exponencial	"."
W#16#0007			","
W#16#0008 a W#16#FFFF	Valores no válidos		

Reglas de formato de la cadena del parámetro OUT:

- Si el tamaño de la cadena convertida es menor que el especificado, se insertan espacios en blanco iniciales en el extremo izquierdo de la cadena.
- Si el bit de signo del parámetro FORMAT es FALSE, los valores de los tipos de datos USINT y SINT se escriben en el búfer de salida sin el signo "+" inicial. El "-" signo se utiliza en caso necesario.
  <espacios iniciales><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Si el bit de signo es TRUE, los valores de los tipos de datos de entero con o sin signo se escriben en el búfer de salida con un carácter de signo inicial.
  <espacios iniciales><signo><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Si el parámetro FORMAT está ajustado a notación exponencial, los valores del tipo de datos Real se escriben en el búfer de salida de la siguiente manera:
  <espacios iniciales><signo><dígito> '.' <dígitos PREC>'E' <signo><dígitos sin cero inicial>
- Si el parámetro FORMAT está ajustado a notación en coma fija, los valores del tipo de datos entero, entero sin signo y real se escriben en el búfer de salida de la siguiente manera:
  <espacios iniciales><signo><dígitos sin ceros iniciales>'.'<dígitos PREC>
- Los ceros a la izquierda del punto decimal (con excepción del dígito adyacente a este) se suprimen.
- Los valores a la derecha del punto decimal se redondean para que se correspondan con el número de dígitos a la derecha del punto decimal que indica el parámetro PREC.
- La cadena de salida debe ser como mínimo tres bytes más grande que el número de dígitos a la derecha del punto decimal.
- Los valores se justifican a la derecha en la cadena de salida.

Condiciones que notifica ENO

Si la operación de conversión encuentra un error, la instrucción vuelve a los resultados siguientes:

- ENO se pone a 0.
- OUT se pone a 0, o como se indica en los ejemplos de conversión de una cadena en un valor.
- OUT no se modifica, o como se indica en los ejemplos cuando OUT es una cadena.

Tabla 9-30 Estado de ENO

ENO	Descripción
1	No hay error
0	Parámetro no permitido o no válido; p. ej. acceso a un DB que no existe
0	Cadena no permitida; la longitud máxima es 0 o 255
0	Cadena no permitida; la longitud actual excede la longitud máxima
0	El valor numérico convertido es demasiado grande para el tipo de datos de OUT indicado.
0	El tamaño máximo de la cadena del parámetro OUT debe ser lo suficientemente grande para aceptar el número de caracteres que indica el parámetro SIZE, comenzando en el parámetro P de posición de carácter.
0	Valor de P no permitido; P=0 o P es mayor que la longitud actual de la cadena
0	El parámetro SIZE debe ser mayor que el parámetro PREC.

Tabla 9-31 Ejemplo de conversión de cadena en valor con S_CONV

Cadena IN	Tipo de datos OUT	Valor OUT	ENO
"123"	Int o DInt	123	TRUE
"-00456"	Int o DInt	-456	TRUE
"123,45"	Int o DInt	123	TRUE
"+2345"	Int o DInt	2345	TRUE
"00123AB"	Int o 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"	N/A	0	FALSE
""	N/A	0	FALSE
"++123"	N/A	0	FALSE
"+-123"	N/A	0	FALSE

Tabla 9-32 Ejemplos de S_CONV (convertir valor en cadena)

Tipo de datos	Valor IN	Cadena OUT [1]	ENO
UInt	123	"xxx123"	TRUE
UInt	0	"xxxxx0"	TRUE
UDInt	12345678	"xxx12345678"	TRUE
Real	+9123,456	"xx+9,123456E+3"	TRUE

Tipo de datos	Valor IN	Cadena OUT [1]	ENO
LReal	+9123,4567890123	xx+9,1234567890123E +3"	TRUE
Real	-INF	"xxxxxxxxxxINF"	FALSE
Real	+INF	"xxxxxxxxxxINF"	FALSE
Real	NaN	"xxxxxxxxxxNaN"	FALSE

[1] Los caracteres "x" representan los caracteres de espacio que ocupan las posiciones vacías en el campo justificado a la derecha que se ha asignado para el valor convertido.

Tabla 9-33 Ejemplo: conversión de STRG_VAL

Cadena IN	FORMAT (W#16#....)	Tipo de datos OUT	Valor OUT	ENO
"123"	0000	Int o DInt	123	TRUE
"-00456"	0000	Int o DInt	-456	TRUE
"123,45"	0000	Int o DInt	123	TRUE
"+2345"	0000	Int o DInt	2345	TRUE
"00123AB"	0000	Int o 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" (o menor)	0002	Real	0,0	TRUE
"12345"	N/A	SInt	0	FALSE
"A123"	N/A	N/A	0	FALSE
""	N/A	N/A	0	FALSE
"++123"	N/A	N/A	0	FALSE
"+-123"	N/A	N/A	0	FALSE

Los siguientes ejemplos de conversión VAL_STRG se basan en una cadena OUT que se inicializa de la manera siguiente:

"Current Temp = xxxxxxxxxx C"
, donde el carácter "x" representa caracteres de espacio asignados al valor convertido.

Tabla 9-34 Ejemplo: conversión de VAL_STRG

Tipo de datos	Valor IN	P	SIZE	FORMAT (W#16#....)	PREC	Cadena 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	N/A	4	Current Temp = xxxxxx-INF C	FALSE
Real	+INF	16	10	N/A	4	Current Temp = xxxxxx+INF C	FALSE
Real	NaN	16	10	N/A	4	Current Temp = xxxxxxxNaN C	FALSE
UDInt	12345678	16	6	N/A	3	Current Temp = xxxxxxxxxx C	FALSE

9.2.3.2 Instrucciones Strg_TO_Chars y Chars_TO_Strg (Convertir a/de cadena de caracteres y Array of CHAR)

Strg_TO_Chars copia una cadena de caracteres ASCII en una matriz de bytes de caracteres.

Chars_TO_Strg copia una matriz de bytes de caracteres ASCII en una cadena de caracteres.

Nota

Solo los tipos de matriz basados en cero (Array [0..n] of Char) o (Array [0..n] of Byte) están permitidos como parámetro de entrada Chars para la instrucción Chars_TO_Strg o como parámetro IN_OUT Chars para la instrucción Strg_TO_Chars .

Tabla 9-35 Instrucción Strg_TO_Chars

KOP / FUP	SCL	Descripción

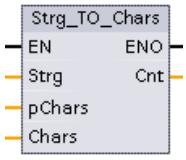
	`Strg_TO_Chars(`
	`  Strg:=_string_in_,`
	`  pChars:=_dint_in_,`
	`  Cnt=>_uint_out_,`
	`  Chars:=_variant_inout_)`
	`;`
		La cadena de entrada completa Strg se copia en una matriz de caracteres en el parámetro IN_OUT Chars.
		La operación sobrescribe los bytes que empiezan por el número de elemento especificado por el parámetro pChars.
		Pueden utilizarse cadenas de todas las longitudes máximas soportadas (1 ... 254).
		No se escribe un delimitador de fin: eso es responsabilidad del usuario. Para establecer un delimitador de fin justo después del último carácter escrito de la matriz, utilice el siguiente número de elemento de la matriz [pChars+Cnt].

Tabla 9-36 Tipos de datos para los parámetros (Strg_TO_Chars)

Parámetro y tipo		Tipo de datos	Descripción
Strg	IN	String, WString	Cadena fuente
pChars	IN	DInt	Número del elemento para el primer carácter de la cadena escrito en la matriz de destino
Chars	IN_OUT	Variante	El parámetro Chars es un puntero hacia una matriz basada en cero [0..n] de caracteres copiados de una cadena de entrada. La matriz puede declararse en un DB o como variables locales en la interfaz del bloque. Ejemplo: "DB1".MyArray apunta a valores de elementos MyArray [0..10] of Char en DB1.
Cnt	OUT	UInt	Recuento de caracteres copiados

Tabla 9-37 Instrucción Chars_TO_Strg

KOP / FUP	SCL	Descripción

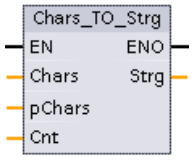
	`Chars_TO_Strg(`
	`  Chars:=_variant_in_,`
	`  pChars:=_dint_in_,`
	`  Cnt:=_uint_in_,`
	`  Strg=>_string_out_);`
		Toda una matriz de caracteres o una parte de ella se copia en una cadena.
		La cadena de salida debe estar declarada antes de ejecutar Chars_TO_Strg. La cadena se sobrescribe con la operación Chars_TO_Strg.
		Pueden utilizarse cadenas de todas las longitudes máximas soportadas (1 ... 254).
		El valor de longitud máxima de una cadena no cambia con la operación Chars_TO_Strg . El proceso de copia de una matriz a una cadena se detiene cuando se ha alcanzado la longitud máxima de la cadena.
		Un carácter nul '$00' o 16#00 en la matriz de caracteres funciona como delimitador y finaliza la copia de caracteres en la cadena.

Tabla 9-38 Tipos de datos para los parámetros (Chars_TO_Strg)

Parámetro y tipo		Tipo de datos	Descripción
Chars	IN	Variante	El parámetro Chars es un puntero hacia una matriz basada en cero [0..n] de caracteres que deben convertirse en una cadena. La matriz puede declararse en un DB o como variables locales en la interfaz del bloque. Ejemplo: "DB1".MyArray apunta a valores de elementos MyArray [0..10] of Char en DB1.
pChars	IN	Dint	Número de elemento del primer carácter en la matriz que debe copiarse. El elemento de matriz [0] es el valor predeterminado.
Cnt	IN	UInt	Recuento de caracteres que se copian: 0 significa todos
Strg	OUT	String, WString	Cadena de destino

Tabla 9-39 Estado de ENO

ENO	Descripción
1	No hay error
0	Chars_TO_Strg: Intento de copiar más bytes de caracteres en la cadena de salida de los que permite el byte de longitud máxima en la declaración de cadena
0	Chars_TO_Strg: El valor de carácter cero (16#00) se ha encontrado en el array de bytes de caracteres de entrada.
0	Strg_TO_Chars: Intento de copiar más bytes de caracteres en el array de salida de los que permite el límite de número de elementos

9.2.3.3 Instrucciones ATH y HTA (Convertir a/de cadena de caracteres ASCII y número hexadecimal)

Utilice las instrucciones ATH (ASCII a hexadecimal) y HTA (hexadecimal a ASCII) para realizar conversiones entre bytes de caracteres ASCII (0 a 9 y mayúsculas solo de A a F) y los correspondientes nibbles hexadecimales de 4 bits.

Tabla 9-40 Instrucción ATH

KOP / FUP	SCL	Descripción

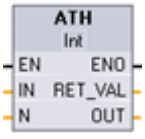
	`ret_val := ATH(`
	`  in:=_variant_in_,`
	`  n:=_int_in_,`
	`  out=>_variant_out_);`
		Convierte caracteres ASCII en dígitos hexadecimales comprimidos.

Tabla 9-41 Tipos de datos para la instrucción ATH

Tipo de parámetro		Tipo de datos	Descripción
IN	IN	Variante	Puntero a matriz de byte de caracteres ASCII
N	IN	UInt	Número de bytes de caracteres ASCII que deben convertirse

Tipo de parámetro		Tipo de datos	Descripción
RET_VAL	OUT	Word	Código de condición de ejecución
OUT	OUT	Variante	Puntero a la matriz de byte hexadecimal que debe convertirse

La conversión comienza en la posición especificada por el parámetro IN y continúa durante N bytes. El resultado se deposita en la posición que indica OUT. Solo pueden convertirse caracteres ASCII válidos 0 a 9, minúsculas de "a" a "f", y mayúsculas de A a F. Cualquier otro carácter se convertirá a cero.

Los caracteres ASCII codificados de 8 bits se convierten en nibbles hexadecimales de 4 bits. Los dos caracteres ASCII se pueden convertir en un solo byte que incluya dos nibbles hexadecimales de 4 bits.

Los parámetros IN y OUT especifican matrices de bytes y no datos String hexadecimales. Los caracteres ASCII se convierten y depositan en la salida hexadecimal en el mismo orden que se leen. Si existe un número impar de caracteres ASCII, se agregan ceros en el nibble situado más a la derecha del último dígito hexadecimal convertido.

Tabla 9-42 Ejemplos: conversión ASCII a hexadecimal (ATH)

Bytes de caracteres IN	N	Valor OUT	ENO
'0a23'	4	W#16#0A23	TRUE
'123AFx1a23'	10	16#123AF01023	FALSE
'a23'	3	W#16#A230	TRUE

Tabla 9-43 Instrucción HTA

KOP / FUP	SCL	Descripción

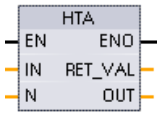
	`ret_val := HTA(`
	`  in:=_variant_in_,`
	`  n:=_uint_in_,`
	`  out=>_variant_out_);`
		Convierte dígitos hexadecimales comprimidos en los correspondientes bytes de caracteres ASCII.

Tabla 9-44 Tipos de datos para la instrucción HTA

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	Variante	Puntero a matriz de byte de entrada
N	IN	UInt	Número de bytes que deben convertirse (cada byte de entrada tiene dos nibbles de 4 bits y produce caracteres ASCII 2N)
RET_VAL	OUT	Word	Código de condición de ejecución
OUT	OUT	Variante	Puntero a matriz de byte de caracteres ASCII

La conversión comienza en la posición especificada por el parámetro IN y continúa durante N bytes. Cada nibble de 4 bits convierte a un carácter ASCII único de 8 bits y produce bytes de caracteres 2N ASCII de salida. Todos los bytes 2N de la salida se escriben como caracteres ASCII de 0 a 9 o mayúsculas A a F. El parámetro OUT especifica una matriz de bytes y no una cadena.

Cada nibble del byte hexadecimal se convierte en un carácter en el mismo orden en que se lee (el nibble situado más a la izquierda de un dígito hexadecimal se convierte primero, seguido por el nibble situado más a la derecha del mismo byte).

Tabla 9-45 Ejemplos: conversión hexadecimal a ASCII (HTA)

Valor IN	N	Bytes de caracteres OUT	ENO (ENO siempre TRUE tras ejecución de HTA)
W#16#0123	2	'0123'	TRUE
DW#16#123AF012	4	'123AF012'	TRUE

Tabla 9-46 Códigos de condición ATH and HTA

RET_VAL (W#16#....)	Descripción	ENO
0000	No hay error	TRUE
0007	Carácter de entrada ATH no válido: Se ha encontrado un carácter que no era un carácter ASCII 0-9, minúscula de "a" a "f" o mayúscula de A a F.	FALSE
8101	Puntero de entrada no permitido o no válido; p. ej. acceso a un DB que no existe.	FALSE
8120	La cadena de entrada tiene un formato no válido, es decir, máx= 0, máx=255, actual>máx o longitud en puntero < máx	FALSE
8182	El búfer de entrada es demasiado pequeño para N	FALSE
8151	Tipo de datos no permitido para búfer de entrada	FALSE
8301	Puntero de salida no permitido o no válido; p. ej. acceso a un DB que no existe.	FALSE
8320	La cadena de salida tiene un formato no válido, es decir, máx= 0, máx=255, actual>máx o longitud en puntero < máx	FALSE
8382	El búfer de salida es demasiado pequeño para N	FALSE
8351	Tipo de datos no permitido para búfer de salida	FALSE

9.2.4 Instrucciones con cadenas

El programa de control puede utilizar las siguientes instrucciones con cadenas y caracteres para crear avisos para displays de operador e históricos del proceso.

9.2.4.1 MAX_LEN (longitud máxima de una cadena de caracteres)

Tabla 9-47 Instrucción de longitud máxima

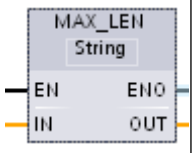
KOP / FUP	SCL	Descripción
	`out := MAX_LEN(in);`	MAX_LEN (longitud máxima de cadena) proporciona el valor de longitud máxima asignado a la cadena IN en la salida OUT. Si el error se produce durante el procesamiento de la instrucción, se obtendrá la longitud de cadena vacía. Los tipos de datos String y WString incluyen dos longitudes: el primer byte (o palabra) proporciona la longitud máxima y el segundo byte (o palabra) proporciona la longitud actual (es el número actual de caracteres válidos). • La longitud máxima de la cadena de caracteres se asigna para cada declaración String o WString entre corchetes. El número de bytes ocupados por un String es 2 bytes mayor que la longitud máxima. El número de palabras ocupadas por un WString es 2 palabras mayor que la longitud máxima. • La longitud actual representa el número de caracteres que se usan realmente. La longitud actual debe ser menor o igual a la longitud máxima. La longitud actual está en bytes para un String y en palabras para un WString. Use la instrucción MAX_LEN i para obtener la longitud máxima de la cadena de caracteres y la instrucción LEN para obtener la longitud actual de una cadena.

Tabla 9-48 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Cadena de entrada
OUT	OUT	DInt	Número máximo de caracteres permitidos para una cadena IN

9.2.4.2 LEN (determinar la longitud de una cadena de caracteres)

Tabla 9-49 Instrucciones de longitud

KOP / FUP	SCL	Descripción
	`out := LEN(in);`	LEN (longitud) proporciona la longitud actual de la cadena IN en la salida OUT. Una cadena vacía tiene una longitud cero.

Tabla 9-50 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Cadena de entrada
OUT	OUT	Int, DInt, Real, LReal	Número de caracteres válidos de la cadena IN

Tabla 9-51 Estado de ENO

ENO	Condición	OUT
1	No hay condición de cadena no válida	Longitud de cadena válida
0	La longitud actual de IN excede la longitud máxima de IN	La longitud actual se pone a 0
	La longitud máxima de IN no cabe centro del área de memoria asignada	
	La longitud máxima de IN es 255 (longitud no permitida)	

9.2.4.3 CONCAT (agrupar cadenas de caracteres)

Tabla 9-52 Instrucción Agrupar cadenas

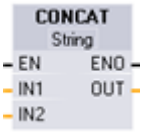
KOP / FUP	SCL	Descripción
	`out := CONCAT(in1, in2);`	CONCAT (agrupar cadenas) agrupa los parámetros de las cadenas IN1 y IN2 para proporcionar una cadena que se deposita en OUT. Tras la agrupación, la cadena IN1 es la parte izquierda y IN2 es la parte derecha de la cadena combinada.

Tabla 9-53 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	String, WString	Cadena de entrada 1
IN2	IN	String, WString	Cadena de entrada 2
OUT	OUT	String, WString	Cadena combinada (cadena 1 + cadena 2)

Tabla 9-54 Estado de ENO

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	La cadena resultante tras la concatenación excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT
	La longitud actual de IN1 excede la longitud máxima de IN1, la longitud actual de IN2 excede la longitud máxima de IN2, o la longitud actual de OUT excede la longitud máxima de OUT (cadena no válida)	La longitud actual se pone a 0
	La longitud máxima de IN1, IN2 o OUT no cabe dentro del área de memoria asignada	
	La longitud máxima de IN1 o IN2 es 255, o la longitud máxima de OUT es 0 o 255 (tipo de datos String)	
	La longitud máxima de IN1 o IN2 es 65534, o la longitud máxima de OUT es 0 o 65534 (tipo de datos WString)	

9.2.4.4 Instrucciones LEFT, RIGHT y MID (Leer los caracteres izquierdos, derechos o centrales de una cadena)

Tabla 9-55 Operaciones de subcadenas izquierda, derecha y central

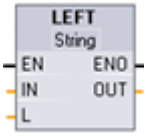
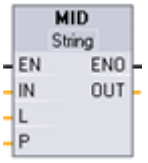
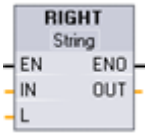
KOP / FUP	SCL	Descripción
	`out := LEFT(in, L);`	LEFT (Leer los caracteres izquierdos de una cadena) crea una subcadena formada por los primeros caracteres L del parámetro de cadena IN. • Si L es mayor que la longitud actual de la cadena IN, OUT devuelve la cadena IN completa. • Si la entrada contiene una cadena vacía, OUT devuelve una cadena vacía.
	`out := MID(in, L, p);`	MID (Leer los caracteres centrales de una cadena) provee la parte central de una cadena. La parte central de una cadena tiene L caracteres de longitud y empieza en la posición de caracteres P (inclusive) Si la suma de L y P excede la longitud actual del parámetro de la cadena IN, se devuelve una subcadena que comienza en la posición de carácter P y que continúa hasta el final de la cadena IN.
	`out := RIGHT(in, L);`	RIGHT (Leer los caracteres derechos de una cadena) provee los últimos caracteres L de una cadena. • Si L es mayor que la longitud actual de la cadena IN, el parámetro OUT devuelve la cadena IN completa. • Si la entrada contiene una cadena vacía, OUT devuelve una cadena vacía.

Tabla 9-56 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Cadena de entrada
L	IN	Int	Longitud de la subcadena que debe crearse: • LEFT utiliza el número de caracteres de la cadena situados más a la izquierda • RIGHT utiliza el número de caracteres de la cadena situados más a la derecha • MID utiliza el número de caracteres comenzando en la posición P de la cadena
P	IN	Int	Solo MID: Posición del primer carácter de subcadena que debe copiarse P= 1, para la posición de carácter inicial de la cadena IN
OUT	OUT	String, WString	Cadena de salida

Tabla 9-57 Estado de ENO

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	• L o P es menor o igual que 0 • P es mayor que la longitud máxima de IN • La longitud actual de IN excede la longitud máxima de IN o la longitud actual de OUT excede la longitud máxima de OUT • La longitud máxima de IN o OUT no cabe dentro del área de memoria asignada • La longitud máxima de IN o OUT es 0 o 255 (tipo de datos String) o 0 o 65534 (tipo de datos WString)	La longitud actual se pone a 0
	La longitud (L) de la subcadena que debe copiarse excede la longitud máxima de la cadena OUT.	Los caracteres se copian hasta alcanzarse la longitud máxima de OUT
	Solo MID: L o P es menor o igual que 0	La longitud actual se pone a 0
	Solo MID: P es mayor que la longitud máxima de IN	
	La longitud actual de IN1 excede la longitud máxima de IN1 o la longitud actual de IN2 excede la longitud máxima de IN2 (cadena no válida)	La longitud actual se pone a 0
	La longitud máxima de IN1, IN2 o OUT no cabe dentro del área de memoria asignada	
	La longitud máxima de IN1, IN2 o OUT es una longitud no permitida: 0 o 255 (tipo de datos String) o 0 o 65534 (tipo de datos WString)	

9.2.4.5 DELETE (borrar caracteres de una cadena)

Tabla 9-58 Instrucción Borrar caracteres de una cadena

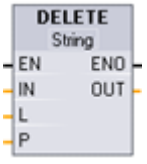
KOP / FUP	SCL	Descripción
	`out := DELETE(in, L, p);`	Borra L caracteres de la cadena IN. El borrado de caracteres comienza en la posición P (inclusive) y la subcadena resultante se deposita en el parámetro OUT. • Si L es igual a cero, la cadena de entrada se deposita en OUT. • Si la suma de L y P es mayor que la longitud de la cadena de entrada, la cadena se borra hasta el final.

Tabla 9-59 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN	IN	String, WString	Cadena de entrada
L	IN	Int	Número de caracteres que deben borrarse
P	IN	Int	Posición del primer carácter que debe borrarse: El primer carácter de la cadena IN tiene el número de posición 1
OUT	OUT	String, WString	Cadena de salida

Tabla 9-60 Estado de ENO

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P es mayor que la longitud actual de IN	IN se copia en OUT sin borrar caracteres
	La cadena resultante tras borrar los caracteres excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT
	L es menor que 0, o P es menor o igual que 0	La longitud actual se pone a 0
	La longitud actual de IN excede la longitud máxima de IN o la longitud actual de OUT excede la longitud máxima de OUT	
	La longitud máxima de IN o OUT no cabe dentro del área de memoria asignada	
	La longitud máxima de IN o OUT es 0 o 255	

9.2.4.6 INSERT (insertar caracteres en una cadena)

Tabla 9-61 Instrucción Insertar caracteres en una cadena

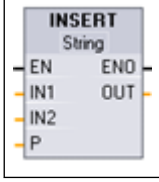
KOP / FUP	SCL	Descripción
	`out := INSERT(in1, in2, p);`	Inserta la cadena IN2 en la cadena IN1. La inserción comienza tras el carácter de la posición P.

Tabla 9-62 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	String, WString	Cadena de entrada 1
IN2	IN	String, WString	Cadena de entrada 2
P	IN	Int	Última posición de carácter en la cadena IN1 antes del punto de inserción de la cadena IN2. El primer carácter de la cadena IN1 tiene el número de posición 1.
OUT	OUT	String, WString	Cadena resultante

Tabla 9-63 Estado de ENO

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P excede la longitud de IN1	IN2 se agrupa con IN1 inmediatamente después del último carácter de IN1
	P es menor que 0	La longitud actual se pone a 0
	La cadena resultante tras la inserción excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT
	La longitud actual de IN1 excede la longitud máxima de IN1, la longitud actual de IN2 excede la longitud máxima de IN2, o la longitud actual de OUT excede la longitud máxima de OUT (cadena no válida)	La longitud actual se pone a 0
	La longitud máxima de IN1, IN2 o OUT no cabe dentro del área de memoria asignada	
	La longitud máxima de IN1 o IN2 es 255, o la longitud máxima de OUT es 0 o 255 (tipo de datos String)	
	La longitud máxima de IN1 o IN2 es 65534, o la longitud máxima de OUT es 0 o 65534 (tipo de datos WString)	

9.2.4.7 REPLACE (reemplazar caracteres de una cadena)

Tabla 9-64 Instrucción Reemplazar caracteres en una cadena

KOP / FUP	SCL	Descripción

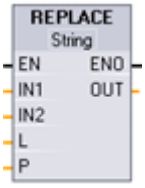
	`out := REPLACE(`
	`  in1:=_string_in_,`
	`  in2:=_string_in_,`
	`  L:=_int_in_,`
	`  p:=_int_in);`
		Reemplaza L caracteres en el parámetro de cadena IN1. La sustitución comienza en la posición de carácter P (inclusive) de la cadena IN1, y los caracteres de reemplazo provienen de la cadena IN2.

Tabla 9-65 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	String, WString	Cadena de entrada
IN2	IN	String, WString	Cadena de caracteres de reemplazo
L	IN	Int	Número de caracteres que deben reemplazarse
P	IN	Int	Posición del primer carácter que debe reemplazarse
OUT	OUT	String, WString	Cadena resultante

Si el parámetro L es igual a cero, la cadena IN2 se inserta en la posición P de la cadena IN1 sin que se borre ningún carácter de la cadena IN1.

Si P es igual a uno, los primeros caracteres L de la cadena IN1 se reemplazan por caracteres de la cadena IN2.

Tabla 9-66 Estado de ENO

ENO	Condición	OUT
1	No se han detectado errores	Caracteres válidos
0	P excede la longitud de IN1	IN2 se agrupa con IN1 inmediatamente después del último carácter de IN1
	P cabe en IN1, pero menos de L caracteres permanecen en IN1	IN2 reemplaza los caracteres finales de IN1 comenzando por la posición P
	La cadena resultante tras la sustitución excede la longitud máxima de la cadena OUT	Los caracteres de la cadena resultante se copian hasta alcanzarse la longitud máxima de OUT
	La longitud máxima de IN1 es 0	Los caracteres IN2 se copian en OUT
	L es menor que 0, o P es menor o igual que 0	La longitud actual se pone a 0
	La longitud actual de IN1 excede la longitud máxima de IN1, la longitud actual de IN2 excede la longitud máxima de IN2, o la longitud actual de OUT excede la longitud máxima de OUT	
	La longitud máxima de IN1, IN2 o OUT no cabe dentro del área de memoria asignada	
	La longitud máxima de IN1 o IN2 es 255, o la longitud máxima de OUT es 0 o 255 (tipo de datos String)	
	La longitud máxima de IN1 o IN2 es 65534, o la longitud máxima de OUT es 0 o 65534 (tipo de datos WString)	

9.2.4.8 FIND (buscar caracteres en una cadena)

Tabla 9-67 Instrucción Buscar caracteres en una cadena

KOP / FUP	SCL	Descripción

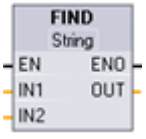
	`out := FIND(`
	`  in1:=_string_in_,`
	`  in2:=_string_in);`
		Proporciona la posición de carácter de la subcadena especificada en IN2 dentro de la cadena IN1. La búsqueda comienza en el lado izquierdo. La posición de carácter del primer resultado encontrado en la cadena IN2 se devuelve en OUT. Si la cadena IN2 no se encuentra en la cadena IN1, se devuelve cero.

Tabla 9-68 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
IN1	IN	String, WString	Buscar en esta cadena
IN2	IN	String, WString	Buscar esta cadena
OUT	OUT	Int	Posición de carácter de la cadena IN1 en el primer resultado de búsqueda

Tabla 9-69 Estado de ENO

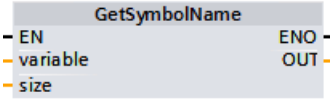
ENO	Condición	OUT
1	No se han detectado errores	Posición de carácter válida
0	IN2 es mayor que IN1	La posición de carácter se pone a 0
	La longitud actual de IN1 excede la longitud máxima de IN1 o la longitud actual de IN2 excede la longitud máxima de IN2 (cadena no válida)	
	La longitud máxima de IN1 o IN2 no cabe dentro del área de memoria asignada	
	La longitud máxima de IN1 o IN2 es 255 (tipo de datos String) o 65535 (tipo de datos WString)	

9.2.5 Información de tiempo de ejecución

9.2.5.1 GetSymbolName (leer nombre de una variable en el parámetro de entrada)

Tabla 9-70 Instrucción GetSymbolName

KOP / FUP	SCL	Descripción

	`OUT := GetSymbolName(`
	`  variable:=_parameter_in_,`
	`  size:=_dint_in_);`
		La instrucción GetSymbolName devuelve una cadena correspondiente al nombre de una variable de la interfaz del bloque.
		El programa puede llamar la instrucción varias veces con diferentes variables. El valor de proceso de la variable es irrelevante.
		La instrucción devuelve el nombre leído en el parámetro OUT.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción GetSymbolName:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
VARIABLE	Input	PARAMETER	Secciones de parámetro Input, Output, InOut	Variable de la interfaz del bloque local para la que debe devolverse un valor de cadena del nombre
SIZE	Input	DINT	I, Q, M, D, L	Limita el número de caracteres emitidos en el parámetro OUT: • SIZE > 0: GetSymbolName devuelve los primeros caracteres SIZE del nombre. • SIZE = 0: GetSymbolName devuelve el nombre completo. • SIZE < 0: GetSymbolName devuelve los últimos caracteres SIZE del nombre.
OUT	Return	WSTRING	I, Q, M, D, L	Salida del nombre de variable suministrado por el parámetro de entrada

Los parámetros de entrada de la interfaz del bloque se especifican en el parámetro VARIABLE. Para este parámetro debe utilizarse únicamente un parámetro de interfaz y no una variable PLC o de bloque de datos.

Para limitar la longitud del nombre de variable leído debe utilizarse el parámetro SIZE. Si la instrucción trunca el nombre, indica que el truncamiento en los caracteres "..." (carácter Unicode 16#2026) aparece al final del nombre. Tenga en cuenta que este carácter tiene una longitud de 1.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Ejemplo: Significado del parámetro SIZE

El siguiente ejemplo ilustra el significado del parámetro SIZE. El nombre de variable siguiente se lee desde la interfaz del bloque: "MyPLCTag" (las comillas dobles al principio y al final forman parte del nombre).

SIZE	Se devuelve GetSymbolName	Explicación
1	'...'	• Primer carácter de WSTRING:' • Identificador de que se ha truncado el nombre: ... • Último carácter de WSTRING:'
2	'"...'	• Primer carácter de WSTRING:' • Primer carácter del nombre e identificador de que se ha truncado el nombre:"... • Último carácter de WSTRING:'

SIZE	Se devuelve GetSymbolName	Explicación
3	'"M..."	• Primer carácter de WSTRING:' • Los dos primeros caracteres del nombre e identificador de que se ha truncado el nombre:"... "M... • Último carácter de WSTRING:'
6	'"MyPL..."	• Primer carácter de WSTRING:' • Los cinco primeros caracteres del nombre e identificador de que se ha truncado el nombre: "MyPL... • Último carácter de WSTRING:'
0	'"MyPLCTag"'	• Primer carácter de WSTRING:' • Todos los caracteres del nombre: "MyPLCTag" • Último carácter de WSTRING:'

Ejemplo: Leer un nombre simbólico

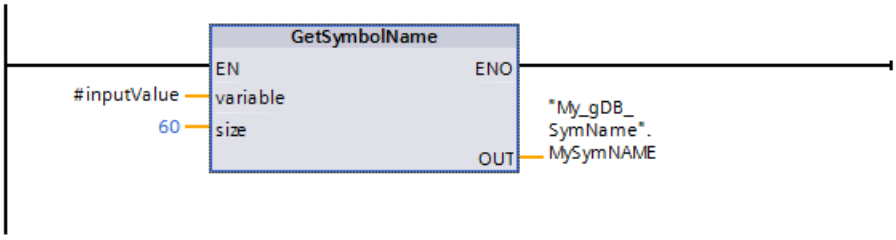
En el ejemplo siguiente se lee el nombre de una variable que está interconectada mediante el parámetro de entrada de un bloque.

Cree dos variables en un bloque de datos global para guardar los datos.

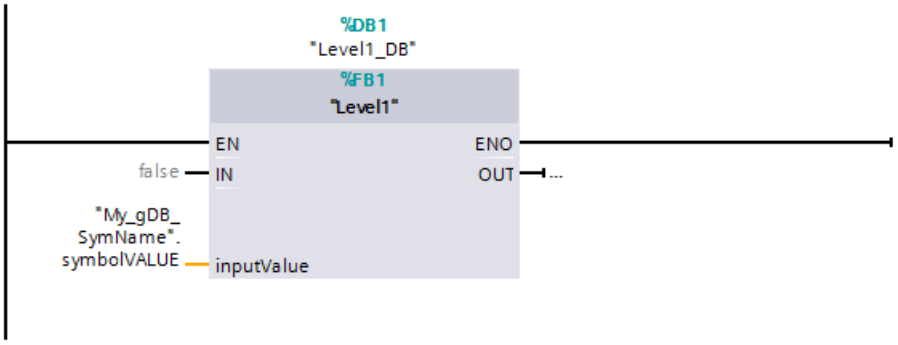
My_gDB_SymName

		Nombre	Tipo de datos	Valor de arranque
1		Static		
2		MySymNAME	WString	WSTRING#''
3		symbolVALUE	Byte	16#42

Cree un parámetro de entrada inputValue con el tipo de datos BYTE en el bloque Level1. Llame la instrucción GetSymbolName en el bloque Level1. Interconecte los parámetros de la instrucción del siguiente modo.



Interconecte el parámetro inputValue del bloque Level1 del siguiente modo.



La instrucción GetSymbolName se ejecuta en el bloque Level1. La interconexión del parámetro de entrada inputValue del bloque Level1 se examina mediante el parámetro de entrada VARIABLE de la instrucción. De este modo, la variable symbolVALUE se lee y se emite como cadena de caracteres en el parámetro de salida OUT ("MySymNAME"). De acuerdo con el valor del parámetro de entrada SIZE, la longitud de la cadena de caracteres está limitada a 60 caracteres.

My_gDB_SymName

		Nombre	Tipo de datos	Valor de arranque	Valor de observación
1		Static			
2		MySymNAME	WString	WSTRING#''	WSTRING#'"My_gDB_SymName".symbolVALUE'
3		symbolVALUE	Byte	16#42	16#42

9.2.5.2 GetSymbolPath (consultar el nombre global compuesto de la alimentación del parámetro de entrada)

Tabla 9-71 Instrucción GetSymbolPath

KOP / FUP	SCL	Descripción

GetSymbolPath
EN — ENO
variable — OUT
size
	`OUT := GetSymbolPath(`
	`  variable:=_parameter_in_,`
	`  size:=_dint_in_);`
		La instrucción GetSymbolPath lee el nombre global compuesto de un parámetro de entrada en la interfaz local de un bloque (FB o FC). El nombre incluye la ruta de almacenamiento y el nombre de la variable.
		El programa puede llamar la instrucción varias veces con diferentes variables. El valor de proceso de la variable es irrelevante.
		La instrucción devuelve el nombre leído en el parámetro OUT.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción GetSymbolPath:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
VARIABLE	Input	PARAMETER	Secciones de parámetro Input, Output, InOut	Selección de la interfaz local para la que debe leerse el nombre global de la alimentación del parámetro de entrada.
SIZE	Input	DINT	I, Q, M, D, L o constante	Limita el número de caracteres emitidos en el parámetro OUT. • SIZE > 0: GetSymbolPath devuelve los primeros caracteres SIZE del nombre. • SIZE = 0: GetSymbolPath devuelve el nombre completo. • SIZE < 0: GetSymbolPath devuelve los últimos caracteres SIZE del nombre.
OUT	Output	WSTRING	I, Q, M, D, L	Salida del nombre de variable de la alimentación del parámetro de entrada.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Utilización

Tenga en cuenta los consejos siguientes sobre el uso de la instrucción GetSymbolPath:

- Especifique la interfaz del bloque a través de la que se lee el nombre de la variable de entrada en el parámetro VARIABLE de la instrucción:
  - Si una variable de bloque de datos proporciona el parámetro de entrada, GetSymbolPath emite el nombre del DB, las estructuras que contiene y el nombre de la variable.
  - Si una variable PLC proporciona los parámetros de entrada, GetSymbolPath emite el nombre de la variable PLC.
  - Si una constante proporciona el parámetro de entrada, GetSymbolPath emite el valor de la constante.
- Para limitar la longitud del nombre de variable leído debe utilizarse el parámetro SIZE. Si el nombre se ha truncado, se indica en el carácter "..." (carácter Unicode 16#2026) al final del nombre. Tenga en cuenta que este carácter tiene una longitud de 1.

Ejemplo: Significado del parámetro SIZE

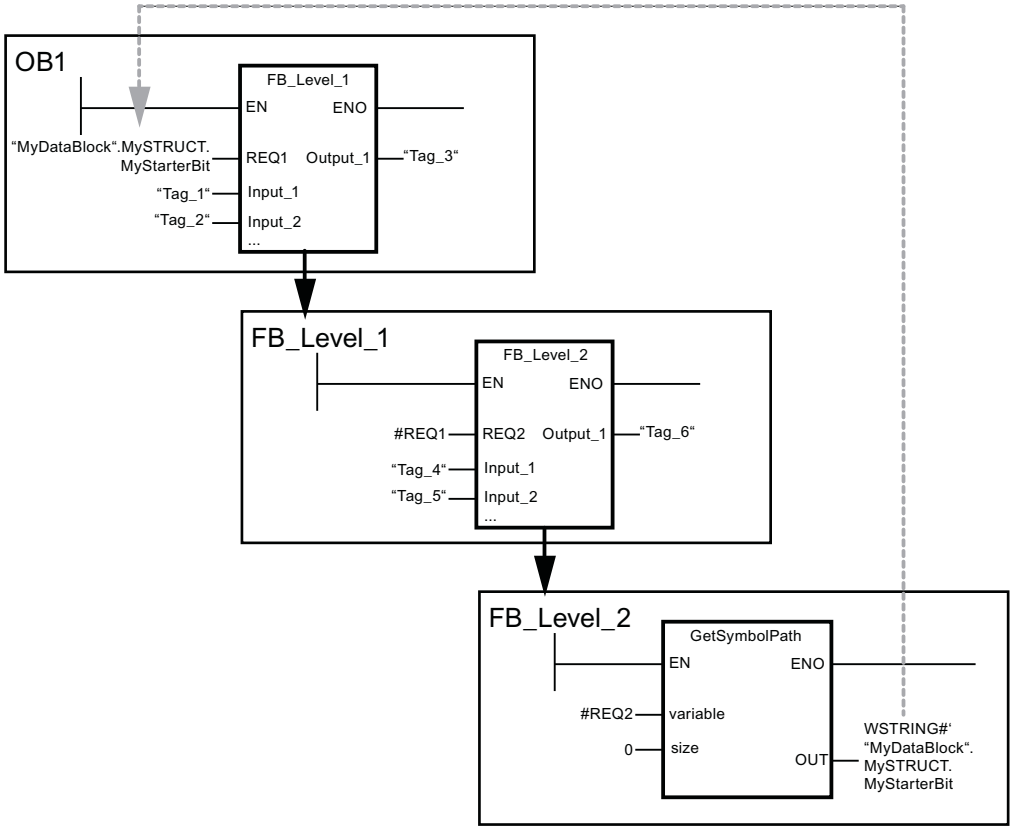
El siguiente ejemplo ilustra el significado del parámetro SIZE. GetSymbolPath ha leído el nombre de variable siguiente desde la interfaz del bloque: "MyPLCTag" (las comillas dobles al principio y al final forman parte del nombre).

SIZE	Se devuelve GetSymbol-Path	Explicación
1	'...'	• Primer carácter de WSTRING:' • Identificador de que se ha truncado el nombre: ... • Último carácter de WSTRING:'
2	'"...'	• Primer carácter de WSTRING:' • Primer carácter del nombre e identificador de que se ha truncado el nombre:"... • Último carácter de WSTRING:'
3	'"M...'	• Primer carácter de WSTRING:' • Los dos primeros caracteres del nombre e identificador de que se ha truncado el nombre:"... "M... • Último carácter de WSTRING:'
6	'"MyPL...'	• Primer carácter de WSTRING:' • Los cinco primeros caracteres del nombre e identificador de que se ha truncado el nombre: "MyPL... • Último carácter de WSTRING:'
0	'"MyPLCTag"'	• Primer carácter de WSTRING:' • Todos los caracteres del nombre: "MyPLCTag" • Último carácter de WSTRING:'

Ejemplo: Llamar GetSymbolPath en varios niveles de llamada del bloque

El ejemplo siguiente muestra el uso de GetSymbolPath en varios niveles de llamada:

- El bloque de organización OB1 llama el bloque FB_Level_1 que, a su vez, llama el bloque FB_Level_2.
- El bloque FB_Level_2 ejecuta GetSymbolPath para leer la ruta del parámetro en la interfaz REQ2.
- Puesto que la interfaz REQ1 proporciona REQ2, la instrucción determina la ruta del parámetro de entrada de REQ1.
- La variable MyStarterBit es el parámetro de entrada REQ1. El bit se encuentra en la estructura MySTRUCT del bloque de datos MyDatablock.
  GetSymbolPath lee esta información y emite la ruta ("MyDataBlock".MySTRUCT.MyStarterBit) en el parámetro OUT.



9.2.5.3 GetInstanceName (leer el nombre de la instancia de bloque)

Tabla 9-72 Instrucción GetInstanceName

KOP / FUP	SCL	Descripción
GetInstanceName - EN ENO - - size OUT -	`OUT := GetInstanceName(` `  size:=_dint_in_);`	La instrucción GetInstanceName se utiliza para leer el nombre del bloque de datos de instancia dentro de un bloque de función.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción GetInstanceName:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
SIZE	Input	DINT	I, Q, M, D, L o constante	Limita el número de caracteres emitidos en el parámetro OUT. • SIZE > 0: GetInstanceName devuelve los primeros caracteres SIZE del nombre. • SIZE = 0: GetInstanceName devuelve el nombre completo. • SIZE < 0: GetInstanceName devuelve los últimos caracteres SIZE del nombre.
OUT	Output	WSTRING	I, Q, M, D, L	Leer el nombre del bloque de datos de instancia

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Ejemplo: Significado del parámetro SIZE

Para limitar la longitud del nombre de instancia leído debe utilizarse el parámetro SIZE. Si la instrucción ha truncado el nombre, indica el truncamiento en el carácter "..." (carácter Unicode 16#2026) al final del nombre. Tenga en cuenta que este carácter tiene una longitud de 1.

El siguiente ejemplo ilustra el significado del parámetro SIZE. GetInstanceName ha leído el nombre de instancia siguiente desde la interfaz del bloque: "Level1_DB" (las comillas dobles al principio y al final forman parte del nombre).

SIZE	Se devuelve GetSymbolPath	Explicación
1	'...'	• Primer carácter de WSTRING:' • Identificador de que se ha truncado el nombre: ... • Último carácter de WSTRING:'
2	'"...'	• Primer carácter de WSTRING:' • Primer carácter del nombre e identificador de que se ha truncado el nombre:"... • Último carácter de WSTRING:'
3	'"L...'	• Primer carácter de WSTRING:' • Los dos primeros caracteres del nombre e identificador de que se ha truncado el nombre:"... "L... • Último carácter de WSTRING:'

SIZE	Se devuelve GetSymbol-Path	Explicación
6	'"Leve...'	• Primer carácter de WSTRING:' • Los cinco primeros caracteres del nombre e identificador de que se ha truncado el nombre: "Leve... • Último carácter de WSTRING:'
0	'"Level1_DB"'	• Primer carácter de WSTRING:' • Todos los caracteres del nombre: "Level1_DB" • Último carácter de WSTRING:'

GetInstanceName escribe el nombre del bloque de datos de instancia en el parámetro OUT La instrucción trunca el nombre si el nombre del bloque de datos de instancia es más grande que la longitud máxima de WSTRING.

Ejemplo: Leer el nombre de un bloque de datos de instancia

El ejemplo siguiente muestra cómo se lee el nombre de un bloque de datos de instancia.

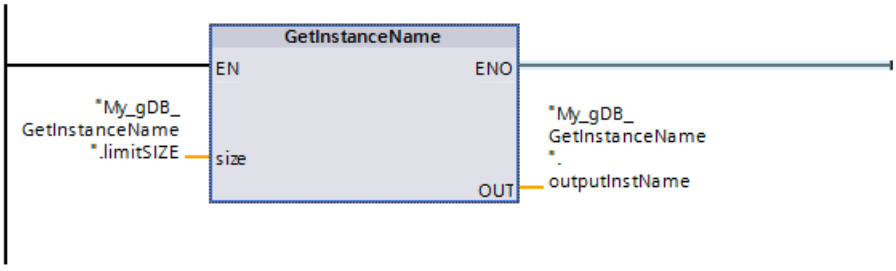
Cree dos variables en un bloque de datos global para guardar los datos.

Defina los parámetros de la instrucción del siguiente modo.

My_gDB_GetInstanceName

	Nombre	Tipo de datos	Valor de arranque
1	Static		
2	limitSIZE	DInt	0
3	outputInsName	WString	WSTRING#''

El bloque Level1_gin ejecuta la instrucción GetInstanceName, que determina el bloque de datos de instancia asociado del bloque Level1_gin y emite el nombre en forma de cadena de caracteres en el parámetro de salida OUT (outputInstName). De acuerdo con el valor 0 del parámetro de entrada SIZE (limitSIZE), la longitud de la cadena de caracteres es ilimitada.

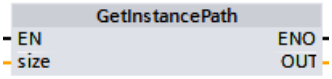


My_gDB_GetInstanceName

	Nombre	Tipo de datos	Valor de arranque	Valor de observación
1	Static			
2	limitSIZE	DInt	0	0
3	outputInsName	WString	WSTRING#''	WSTRING#'"Level_1_DB"'

9.2.5.4 GetInstancePath (consultar el nombre global compuesto de la instancia de bloque)

Tabla 9-73 Instrucción GetInstancePath

KOP / FUP	SCL	Descripción
	`OUT := GetInstancePath( size:=_dint_in_);`	La instrucción GetInstancePath se utiliza para leer el nombre global compuesto de la instancia del bloque dentro de un bloque de función. El nombre global compuesto de la instancia del bloque es la ruta de la jerarquía de llamada completa cuando el programa llama varias instancias.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción GetInstancePath:

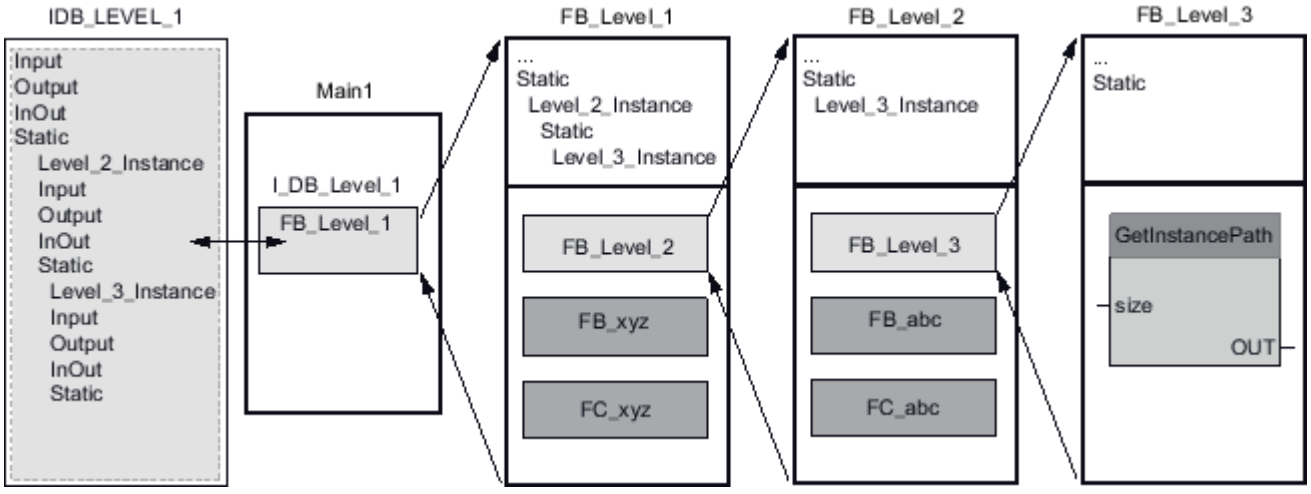
Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
SIZE	Input	DINT	I, Q, M, D, L o constante	Limita el número de caracteres emitidos en el parámetro OUT. • SIZE > 0: GetInstancePath devuelve los primeros caracteres SIZE del nombre. • SIZE = 0: GeInstancePath devuelve el nombre completo. • SIZE < 0: GetInstancePath devuelve los últimos caracteres SIZE del nombre.
OUT	Output	WSTRING	I, Q, M, D, L	Leer el nombre global de la instancia del bloque. Si el nombre global de la instancia del bloque es más grande que la longitud máxima de WSTRING (254 caracteres), GetInstancePath trunca el nombre.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Ejemplo: Llamar GetInstancePath para obtener la ruta de una llamada de FB multiinstancia

En el ejemplo siguiente, el bloque de función FB_Level_3 llama la instrucción GetInstancePath.

- El bloque de función FB_Level_3 guarda sus datos en el bloque de función FB_Level_2 que efectúa la llamada.
- El bloque de función FB_Level_2, a su vez, guarda sus datos en el bloque de función FB_Level_1 que efectúa la llamada.
- El bloque de función FB_Level_1, a su vez, guarda sus datos en su bloque de datos de instancia IDB_LEVEL_1. Debido al uso de multiinstancias, el bloque de datos de instancia de FB_Level_1 contiene todos los datos de los tres bloques de función.



La instrucción GetInstancePath devuelve los valores siguientes en este ejemplo, en función del valor del parámetro SIZE:

SIZE	Se devuelve GetInstance-Path	Explicación
1	'...'	• Primer carácter de WSTRING:' • Identificador de que se ha truncado el nombre: ... • Último carácter de WSTRING:'
2	'"...'	• Primer carácter de WSTRING:' • Primer carácter del nombre e identificador de que se ha truncado el nombre:"... • Último carácter de WSTRING:'
3	'"I...'	• Primer carácter de WSTRING:' • Los dos primeros caracteres del nombre e identificador de que se ha truncado el nombre:"... "I... • Último carácter de WSTRING:'
6	'"IDB_...'	• Primer carácter de WSTRING:' • Los cinco primeros caracteres del nombre e identificador de que se ha truncado el nombre: "IDB_... • Último carácter de WSTRING:'
0	'"IDB_LEVEL_1".Level_2_ Instance.Level_3_Instance'	• Primer carácter de WSTRING:' • Todos los caracteres del nombre: "IDB_LEVEL_1".Level_2_Instance.Level_3_Instance • Último carácter de WSTRING:'

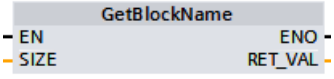
Nota

Uso de GetInstancePath en bloques de función con instancia individual

Si el bloque de función en el que se llama GetInstancePath guarda datos en su propio bloque de datos de instancia, GetInstancePath emite el nombre de la instancia individual como nombre global. El resultado en el parámetro OUT se corresponde en este caso con la instrucción GetInstanceName (Página 362).

9.2.5.5 GetBlockName (leer el nombre del bloque)

Tabla 9-74 Instrucción GetBlockName

KOP / FUP	SCL	Descripción
	`RET_VAL := GetBlockName(` `  size:=_dint_in_);`	La instrucción GetBlockName se utiliza para leer el nombre del bloque en el que se llama la instrucción.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción GetBlockName:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
SIZE	Input	UINT	I, Q, M, D, L o constante	Limita el número de caracteres emitidos en el parámetro RET_VAL. • SIZE > 0: GetBlockName devuelve los primeros caracteres SIZE del nombre. • SIZE = 0: GetBlockName devuelve el nombre completo. • SIZE < 0: GetBlockName devuelve los últimos caracteres SIZE del nombre.
RET_VAL	Output	WSTRING	I, Q, M, D, L	Leer el nombre del bloque de datos de instancia

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Ejemplo: Significado del parámetro SIZE

Para limitar la longitud del nombre del bloque en un determinado número de caracteres, especifique la longitud máxima en el parámetro SIZE. Si GetBlockName trunca el nombre, indica el truncamiento en el carácter "..." (carácter Unicode 16#2026) al final del nombre. Tenga en cuenta que este carácter tiene una longitud de 1.

El siguiente ejemplo ilustra el significado del parámetro SIZE. GetBlockName ha leído el nombre de bloque siguiente: Level1_gbn (las comillas dobles al principio y al final forman parte del nombre).

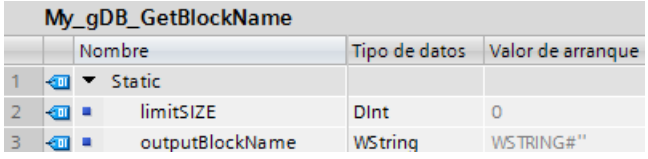
SIZE	Se devuelve GetBlockName	Explicación
1	'...'	• Primer carácter de WSTRING:' • Identificador de que se ha truncado el nombre: ... • Último carácter de WSTRING:'
2	'"...'	• Primer carácter de WSTRING:' • Primer carácter del nombre e identificador de que se ha truncado el nombre:"... • Último carácter de WSTRING:'
3	'"L...'	• Primer carácter de WSTRING:' • Los dos primeros caracteres del nombre e identificador de que se ha truncado el nombre:"... "L... • Último carácter de WSTRING:'
6	'"Leve...'	• Primer carácter de WSTRING:' • Los cinco primeros caracteres del nombre e identificador de que se ha truncado el nombre: "Leve... • Último carácter de WSTRING:'
0	'"Level1_gbn"'	• Primer carácter de WSTRING:' • Todos los caracteres del nombre: "Level1_gbn" • Último carácter de WSTRING:'

GetBlockName escribe el nombre del bloque en el parámetro RET_VAL. Si el nombre del bloque es más grande que la longitud máxima de WSTRING, trunca el nombre.

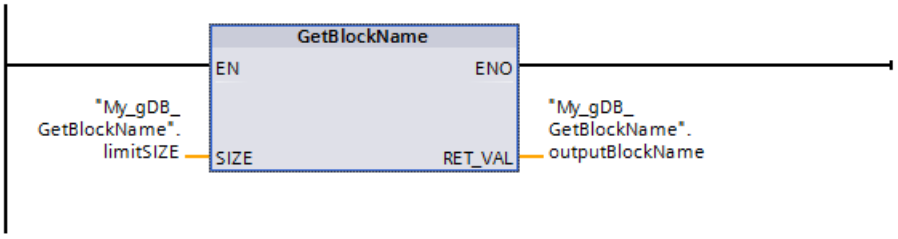
Ejemplo: Leer un nombre de bloque

El ejemplo siguiente muestra cómo se lee el nombre de un bloque.

1. Cree dos variables en un bloque de datos global para guardar los datos.



2. Defina los parámetros de la instrucción del siguiente modo:



El bloque Level1_gbn ejecuta la instrucción GetBlockName. GetBlockName lee el nombre del bloque Level1_gbn y emite el nombre en forma de cadena de caracteres en el parámetro de salida RET_VAL (outputBlockName). Puesto que el parámetro SIZE es 0 (limitSIZE), la longitud de la cadena de caracteres es ilimitada.

My_gDB_GetBlockName

	Nombre	Tipo de datos	Valor de arranque	Valor de observación
1	Static			
2	limitSIZE	DInt	0	0
3	outputBlockName	WString	WSTRING#''	WSTRING#'"Level_1_gbn"'

9.3 E/S distribuidas (PROFINET, PROFIBUS o AS-i)

9.3.1 Instrucciones de E/S descentralizadas

Las siguientes instrucciones E/S descentralizadas pueden utilizarse con PROFINET, PROFIBUS o AS-i:

- Instrucción RDREC (Página 370): Lee un registro con el número INDEX de un módulo o dispositivo.
- Instrucción WRREC (Página 370): Transfiere un registro con el número INDEX de un módulo o dispositivo definido en ID.
- Instrucción GETIO (Página 373): Lee de forma coherente todas las entradas de un esclavo DP normalizado / dispositivo PROFINET IO.
- Instrucción SETIO (Página 374): Transfiere de forma coherente los datos del rango de origen definido en el parámetro OUTPUTS al esclavo DP normalizado / dispositivo PROFINET IO direccionado.
- Instrucción GETIO_PART (Página 375): Lee de forma coherente una parte relacionada de las entradas de un módulo de periferia.
- Instrucción SETIO_PART (Página 376): Escribe de forma coherente los datos del área de origen delimitada por el parámetro OUTPUTS en las salidas de un módulo de periferia.
- Instrucción RALRM (Página 377): Permite recibir una alarma con toda la información correspondiente de un módulo o dispositivo y proporciona esta información a sus parámetros de salida.
- Instrucción DPRD_DAT (Página 389): Permite leer de forma coherente áreas de datos mayores que 64 bytes de un módulo o dispositivo con la instrucción DPRD_DAT.
- Instrucción DPWR_DAT (Página 389): Permite escribir de forma coherente áreas de datos mayores que 64 bytes en un módulo o dispositivo con la instrucción DPWR_DAT.

La instrucción D_ACT_PT (Página 380) permite desactivar y activar dispositivos PROFINET IO configurados de forma selectiva. También es posible determinar si cada dispositivo PROFINET IO asignado está activado o desactivado en esos momentos.

Nota

Nota: La instrucción D_ACT_DP solo puede utilizarse con dispositivos PROFINET IO. No es posible emplear la instrucción con esclavos PROFIBUS DP.

La instrucción DPNRM_DG (Página 396) permite leer los datos de diagnóstico actuales de un esclavo DP en el formato especificado en EN 50 170 volumen 2, PROFIBUS.

Nota

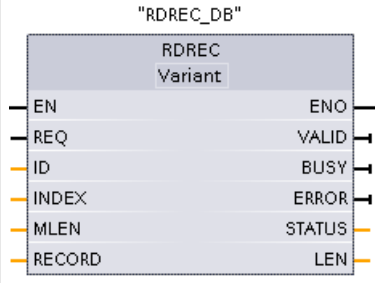
La instrucción DPNRM_DG solo puede utilizarse con PROFIBUS.

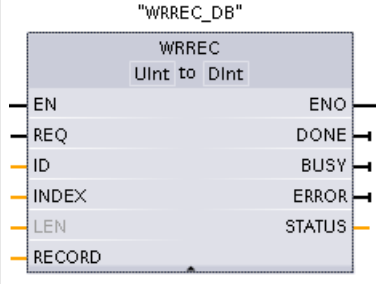
9.3.2 RDREC y WRREC (leer/escribir registro)

Las instrucciones RDREC (leer registro) y WRREC (escribir registro) se pueden utilizar con PROFINET, PROFIBUS y AS-i.

Tabla 9-75 Instrucciones RDREC y WRREC

KOP / FUP	SCL	Descripción

	`"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_);`
		Use la instrucción RDREC para leer un registro con el número INDEX de un componente direccionado por la ID, como un rack central o un componente descentralizado (PROFIBUS DP o PROFINET IO). Asigne el número máximo de bytes que deben leerse en MLEN. La longitud seleccionada del área de destino RECORD debe tener como mínimo la longitud de MLEN bytes.

	`"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_);`
		Use la instrucción WRREC para transferir un RECORD con el número de registro INDEX a un esclavo DP o dispositivo PROFINET IO direccionado por la ID, como un módulo del rack central o un componente descentralizado (PROFIBUS DP o PROFINET IO).
		Asigne la longitud en bytes del registro que debe transmitirse. Así, la longitud seleccionada del área de origen RECORD debe tener como mínimo la longitud de LEN bytes.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En los ejemplos SCL, "RDREC_DB" y "WRREC_DB" son los nombres de los DB de instancia.

Tabla 9-76 Tipos de datos RDREC y WRREC para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	REQ = 1: Transferir juego de datos
ID	IN	HW_IO (Word)	Dirección lógica del esclavo DP o componente PROFINET IO (módulo o submódulo): • Para un módulo de salida debe activarse el bit 15 (por ejemplo, para dirección 5: ID:= DW#16#8005). • Para un módulo combinado debe especificarse le menor de las dos direcciones. **Nota**: en V3.0, la ID de dispositivo puede determinarse de uno de los modos siguientes: • seleccionando lo siguiente en la "Vista de redes": – dispositivo (casilla gris) – "Propiedades" del dispositivo – "ID de hardware" **Nota**: no todos los dispositivos muestran sus IDs de hardware. • seleccionando lo siguiente en el menú "Árbol del proyecto": – variables PLC – tabla de variables predeterminada – tabla de constantes del sistema Se muestran todas las IDs de hardware configuradas para el dispositivo. **Nota**: en V4.0, la ID de dispositivo (identificador de hardware) para el módulo de interfaz se determina yendo a la tabla de variables y localizando el parámetro "Nombre de dispositivo [ENCABEZADO]" en Constantes de sistema.
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Número de registro
MLEN	IN	Byte, USInt, UInt	Longitud máxima en bytes de la información del registro que debe recuperarse (RDREC)
VALID	OUT	Bool	Se ha recibido un nuevo registro y es válido (RDREC). El bit VALID es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
DONE	OUT	Bool	El registro se ha transferido (WRREC). El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• BUSY = 1: El proceso de lectura (RDREC) o escritura (WRREC) todavía no ha terminado. • BUSY = 0: La transmisión del juego de datos ha finalizado.
ERROR	OUT	Bool	ERROR = 1: Se ha producido un error de lectura (RDREC) o escritura (WRREC). El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	DWord	Estado (Página 385) de bloque o información de error (Página 531)

Parámetro y tipo		Tipo de datos	Descripción
LEN	OUT (RDREC) IN (WRREC)	UInt	• Longitud de la información de registro recuperada (RDREC) • Longitud máxima en bytes del registro que debe transferirse (WRREC)
RECORD	IN_OUT	Variant	• Área de destino para el registro recuperado (RDREC) • Registro (WRREC)

Las instrucciones RDREC y WRREC funcionan de forma asíncrona, de modo que el procesamiento se extiende a lo largo de varias llamadas de la instrucción. Inicie la tarea llamando RDREC o WRREC con REQ = 1.

El estado de la tarea se muestra en el parámetro de salida BUSY y en los dos bytes centrales del parámetro de salida STATUS. La transferencia del registro se ha completado cuando el parámetro de salida BUSY se ha establecido en FALSE.

Un valor de TRUE (solo durante un ciclo) en el parámetro de salida VALID (RDREC) o DONE (WRREC) verifica que el registro se ha transferido correctamente al área de destino RECORD (RDREC) o al dispositivo de destino (WRREC). En caso de RDREC, el parámetro de salida LEN contiene la longitud de los datos recuperados en bytes.

El parámetro de salida ERROR (solo durante un ciclo cuando ERROR = TRUE) indica si se ha producido un error en una transmisión. En ese caso, el parámetro de salida STATUS (solo durante un ciclo cuando ERROR = TRUE) contiene la información de error.

Los registros están definidos por el fabricante del dispositivo de hardware. Consulte la documentación del dispositivo del fabricante de hardware para obtener información detallada acerca de un registro.

Es posible utilizar un máximo de cuatro instrucciones RDREC y cuatro instrucciones WRREC al mismo tiempo.

Nota

Si un esclavo DPV1 se configura mediante un archivo GSD (GSD vers. 3 y superior) y la interfaz DP del maestro DP está ajustada en "compatible con S7", no será posible leer juegos de datos de los módulos E/S en el programa de usuario con "RDREC" o escribir en los módulos E/S con "WRREC". En este caso, el maestro DP direcciona el slot equivocado (slot configurado + 3).

Remedio: pone la interfaz del maestro DP a "DPV1".

Nota

Las interfaces de las instrucciones "RDREC" y "WRREC" son idénticas a los FBs "RDREC" y "WRREC" definidos en la norma "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Nota

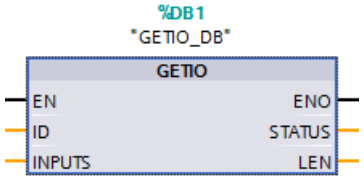
Si se utiliza "RDREC" o "WRREC" para leer o escribir un juego de datos para PROFINET IO, la CPU interpretará los valores negativos de los parámetros INDEX, MLEN y LEN como un entero de 16 bits sin signo.

9.3.3 GETIO (leer memoria imagen de proceso)

La instrucción "GETIO" se utiliza para leer de forma coherente las entradas de módulos o submódulos de esclavos DP o dispositivos PROFINET IO. La instrucción "GETIO" llama la instrucción "DPRD_DAT (Página 389)". Si no hay errores durante la transmisión de datos, los datos leídos se han introducido en el área de destino indicada en INPUTS.

Tabla 9-77 Instrucción GETIO (leer memoria imagen de proceso)

KOP / FUP	SCL	Descripción

	`"GETIO_DB"(`
	`id:=_uint_in_,`
	`status=>_dword_out_,`
	`len=>_int_out_,`
	`inputs:=_variant_inout_);`
		La instrucción "GETIO" se utiliza para leer de forma coherente todas las entradas de un esclavo DP normalizado / dispositivo PROFINET IO.

1 STEP 7 crea el DB automáticamente al insertar la instrucción.

2 En el ejemplo SCL, "GETIO_DB" es el nombre del DB de instancia.

El área de destino debe tener una longitud mayor o igual a la longitud del componente seleccionado.

Si se lee de un esclavo DP normalizado con una configuración modular o con varios identificadores DP, solo se puede acceder a los datos de un componente o identificador DP en la dirección inicial configurada con una llamada de "GETIO".

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "GETIO":

Parámetro	Declaración	Tipo de datos	Descripción
ID	IN	HW_SUBMO-DULE	Identificador de hardware del esclavo DP normalizado / dispositivo PROFINET IO
STATUS[1]	OUT	DWord	Contiene la información de error de "DPRD_DAT (Página 389)" en la forma DW#16#40xxxx00
LEN	OUT	Int	Cantidad de datos leídos en bytes
INPUTS	IN_OUT	Variant	Área de destino de los datos leídos: El área de destino debe tener una longitud mayor o igual a la longitud del esclavo DP normalizado / dispositivo PROFINET IO seleccionado. Puede utilizar los tipos de datos siguientes: • Tipos de datos de sistema y array de tipos de datos de sistema: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Tipos de datos personalizados (UDT) • Estructuras (STRUCT), pero solo en bloques de datos no optimizados (DB)

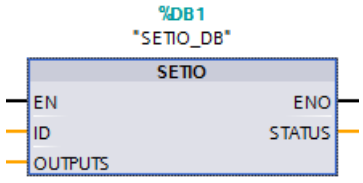
1 Utilice el tipo de datos DWord para visualizar los códigos de error de "GETIO".

9.3.4 SETIO (transferir memoria imagen de proceso)

La instrucción "SETIO" se utiliza para transferir de forma coherente los datos del rango de origen definido en el parámetro OUTPUTS a los módulos o submódulos direccionados de esclavos DP normalizados y dispositivos PROFINET IO. Si el área de direcciones relevante del esclavo DP normalizado / dispositivo PROFINET IO se ha configurado como rango coherente en una memoria imagen de proceso, los datos se transfieren a la memoria imagen de proceso. "SETIO" llama la instrucción "DPWR_DAT (Página 389)" durante la transferencia.

Tabla 9-78 Instrucción SETIO (leer memoria imagen de proceso)

KOP / FUP	SCL	Descripción

	`"SETIO_DB"(`
	`id:=_uint_in_,`
	`status=>_dword_out_,`
	`outputs:=_variant_inout_);`
		La instrucción "SETIO" se utiliza para transferir de forma coherente los datos del rango de origen definido en el parámetro OUTPUTS al esclavo DP normalizado / dispositivo PROFINET IO direccionado.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "SETIO_DB" es el nombre del DB de instancia.

El rango de origen debe tener una longitud mayor o igual a la longitud del componente seleccionado.

En el caso de un esclavo DP normalizado / dispositivo PROFINET IO con configuración modular o varios identificadores DP, solo es posible acceder a un identificador DP / componente por llamada de "SETIO".

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "SETIO":

Parámetro	Declaración	Tipo de datos	Descripción
ID	IN	HW_SUBMO-DULE	Identificador de hardware del esclavo DP normalizado / dispositivo PROFINET IO
STATUS[1]	OUT	DWord	Contiene la información de error de "DPWR_DAT (Página 389)" en la forma DW#16#40xxxx00
OUTPUTS	IN_OUT	Variant	Rango de origen de los datos que deben escribirse: El rango de origen debe tener una longitud mayor o igual a la longitud del esclavo DP normalizado / dispositivo PROFINET IO seleccionado. Puede utilizar los tipos de datos siguientes: • Tipos de datos de sistema y array de tipos de datos de sistema: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Tipos de datos personalizados (UDT) • Estructuras (STRUCT), pero solo en bloques de datos no optimizados (DB)

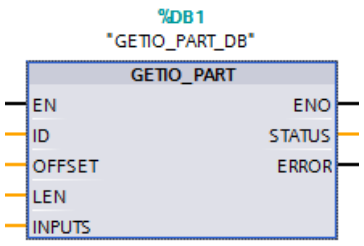
[1] Utilice el tipo de datos DWord para visualizar los códigos de error de "SETIO".

9.3.5 GETIO_PART (leer área de memoria imagen de proceso)

La instrucción "GETIO_PART" se utiliza para leer de forma coherente una parte de las entradas de módulos o submódulos de esclavos DP o dispositivos PROFINET IO. GETIO_PART llama la instrucción "DPRD_DAT (Página 389)".

Tabla 9-79 GETIO_PART (leer área de memoria imagen de proceso)

KOP / FUP	SCL	Descripción

	`"GETIO_PART_DB"(`
	`id:=_uint_in_,`
	`offset:=_int_in_,`
	`len:=_int_in_,`
	`status=>_dword_out_,`
	`error=>_bool_out_,`
	`inputs:=_variant_inout_);`
		La instrucción GETIO_PART se utiliza para leer de forma coherente una parte relacionada de las entradas de un módulo de periferia.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "GETIO_PART_DB" es el nombre del DB de instancia.

Utilice el parámetro de entrada ID para seleccionar el módulo de periferia por medio del identificador de hardware.

Utilice los parámetros OFFSET y LEN para especificar la parte del área de la memoria imagen de proceso que debe leerse. Si el área de entrada que abarcan OFFSET y LEN no está completamente cubierta por el módulo, el bloque devuelve el código de error DW#16#40**80B7**00.

La longitud del área de destino debe ser mayor o igual a la cantidad de datos que se leerán:

- Si no hay errores durante la transmisión de datos, ERROR recibe el valor FALSE. Los datos leídos se escriben en el área de destino definida en el parámetro INPUTS.
- Si se produce un error durante la transmisión de datos, ERROR recibe el valor TRUE. El parámetro STATUS recibe la información de error de DPRD_DAT.
- Si el área de destino es mayor que LEN, la instrucción escribe en los primeros bytes LEN del área de destino. ERROR recibe el valor FALSE.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción GETIO_PART:

Parámetro	Declaración	Tipo de datos	Descripción
ID	IN	HW_SUBMO-DULE	Identificador de hardware del módulo
OFFSET	IN	Int	Número del primer byte que debe leerse en la memoria imagen de proceso del componente (mínimo valor posible: 0)
LEN	IN	Int	Número de bytes que deben leerse
STATUS[1]	OUT	DWord	Contiene la información de error de DPRD_DAT (Página 389) en la forma DW#16#40xxxx00 si ERROR = TRUE

Parámetro	Declaración	Tipo de datos	Descripción
ERROR	OUT	Bool	Visualización del error: ERROR = TRUE si se produce un error cuando se llama DPRD_DAT (Página 389)
INPUTS	IN_OUT	Variant	Área de destino de los datos leídos: Si el área de destino es mayor que LEN, la instrucción escribe en los primeros bytes LEN del área de destino. Puede utilizar los tipos de datos siguientes: • Tipos de datos de sistema y array de tipos de datos de sistema: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Tipos de datos personalizados (UDT) • Estructuras (STRUCT), pero solo en bloques de datos no optimizados (DB)

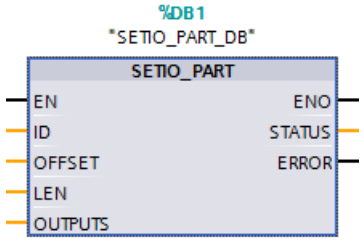
[1] Utilice el tipo de datos DWord para visualizar los códigos de error de GETIO_PART .

9.3.6 SETIO_PART (transferir área de memoria imagen de proceso)

La instrucción "SETIO_PART" se utiliza para escribir de forma coherente datos del área de origen delimitada por OUTPUTS en las salidas de módulos o submódulos de esclavos DP o dispositivos PROFINET IO. SETIO_PART llama la instrucción "DPWR_DAT (Página 389)".

Tabla 9-80 SETIO_PART (transferir área de memoria imagen de proceso)

KOP / FUP	SCL	Descripción

	`"SETIO_PART_DB"(`
	`id:=_uint_in_,`
	`offset:=_int_in_,`
	`len:=_int_in_,`
	`status=>_dword_out_,`
	`error=>_bool_out_,`
	`outputs:=_variant_inout_);`
		La instrucción SETIO_PART se utiliza para escribir de forma coherente los datos del área de origen delimitada por OUTPUTS en las salidas de un módulo de periferia.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "SETIO_PART_DB" es el nombre del DB de instancia.

Con el parámetro de entrada ID, se selecciona el módulo de periferia en base al identificador de hardware.

Con los parámetros OFFSET y LEN, se asigna la parte del área de la memoria imagen de proceso que debe escribirse para el componente direccionado por ID. Si el área de salida que abarcan OFFSET y LEN no está completamente cubierta por el módulo, el bloque devuelve el código de error DW#16#40**80B7**00.

La longitud del área de destino debe ser mayor o igual a la cantidad de datos que se leerán:

- Si no hay errores durante la transmisión de datos, ERROR recibe el valor FALSE.
- Si se produce un error durante la transmisión de datos, ERROR recibe el valor TRUE y STATUS recibe la información de error de DPWR_DAT.
- Si el área de origen es mayor que LEN, la instrucción transfiere los primeros bytes LEN de OUTPUTS. ERROR recibe el valor FALSE.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción SETIO_PART:

Parámetro	Declaración	Tipo de datos	Descripción
ID	IN	HW_SUBMO-DULE	Identificador de hardware del módulo de periferia
OFFSET	IN	Int	Número del primer byte que debe escribirse en la memoria imagen de proceso del componente (mínimo valor posible: 0)
LEN	IN	Int	Número de bytes que deben escribirse
STATUS[1]	OUT	DWord	Contiene la información de error de DPWR_DAT (Página 389) en la forma DW#16#40xxxx00 si ERROR = TRUE
ERROR	OUT	Bool	Visualización del error: ERROR = TRUE si se produce un error cuando se llama DPWR_DAT (Página 389)
OUTPUTS	IN_OUT	Variant	Rango de origen de los datos que deben escribirse: Si el área de origen es mayor que LEN, los primeros bytes LEN se transfieren de OUTPUTS. Puede utilizar los tipos de datos siguientes: • Tipos de datos de sistema y array de tipos de datos de sistema: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT • Tipos de datos personalizados (UDT) • Estructuras (STRUCT), pero solo en bloques de datos no optimizados (DB)

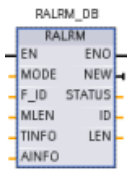
[1] Utilice el tipo de datos DWord para visualizar los códigos de error de SETIO_PART .

9.3.7 RALRM (recibir alarma)

La instrucción RALRM (Recibir alarma) puede usarse con PROFINET y PROFIBUS.

Tabla 9-81 Instrucción RALRM

KOP / FUP	SCL	Descripción

	`"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_);`
		Use la instrucción RALRM (leer alarma) para leer la información de una alarma de diagnóstico desde módulos o dispositivos de E/S PROFIBUS o PROFINET.
		La información de los parámetros de salida contiene la información de inicio del OB al que se ha llamado, así como información sobre el origen de la alarma.
		Llame a RALRM en un OB de alarma para devolver información sobre los eventos que han producido la alarma. En el S7-1200. Se soportan las siguientes alarmas de OB de diagnóstico: estado, actualización, perfil, alarma de error de diagnóstico, enchufe o desenchufe de módulos, fallo de rack o estación.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "RALRM_DB" es el nombre del DB de instancia.

Tabla 9-82 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
MODE	IN	Byte, USInt, SInt, Int	Estado operativo
F_ID	IN	HW_IO (Word)	La dirección inicial lógica del componente (módulo) del cual deben recibirse alarmas **Nota**: la ID de dispositivo puede determinarse de uno de los modos siguientes: • seleccionando lo siguiente en la "Vista de redes": – dispositivo (casilla gris) – "Propiedades" del dispositivo – "ID de hardware" **Nota**: No todos los dispositivos muestran sus IDs de hardware. • seleccionando lo siguiente en el menú "Árbol del proyecto": – variables PLC – tabla de variables predeterminada – tabla de constantes del sistema – Se muestran todas las IDs de hardware configuradas para el dispositivo.
MLEN	IN	Byte, USInt, UInt	Longitud máxima en bytes de la información de alarma de datos que debe recibirse. Si MLEN es 0, se permitirá la recepción de tanta información de alarma de datos como esté disponible en el área de destino de AINFO.
NEW	OUT	Bool	Se ha recibido una nueva alarma.
STATUS	OUT	DWord	Estado de la instrucción RALRM. Consulte "Parámetro STATUS para RDREC, WRREC y RALRM" (Página 385) para obtener más información.
ID	OUT	HW_IO (Word)	Identificador de hardware del módulo de E/S que ha provocado la alarma de diagnóstico. **Nota**: consulte el parámetro F_ID para obtener una explicación de cómo determinar la ID de dispositivo.
LEN	OUT	DWord, UInt, UDInt, DInt, Real, LReal	Longitud de la información de alarma de AINFO recibida.
TINFO	IN_OUT	Variant	Información de la tarea: Área de destino para inicio de OB e información de administración. La longitud de TINFO siempre es de 32 bytes.
AINFO	IN_OUT	Variant	Información de alarma: Área de destino para información de encabezado e información de alarma adicional. Para AINFO debe preverse una longitud equivalente como mínimo a los bytes de MLEN si MLEN es mayor que 0. La longitud de AINFO es variable.

Nota

Si se llama "RALRM" en un OB cuyo evento de arranque no es una alarma E/S, la instrucción ofrecerá información reducida en sus salidas en correspondencia.

Asegúrese de utilizar diferentes DBs instancia al llamar "RALRM" en OBs distintos. Utilice un DB de instancia independiente para cada evento de arranque del OB para evaluar datos de una llamada de "RALRM" fuera del OB de alarma asociado.

Nota

La interfaz de la instrucción "RALRM" es idéntica al FB "RALRM" definido en la norma "PROFIBUS Guideline PROFIBUS Communication and Proxy Function Blocks according to IEC 61131-3".

Llamar RALRM

La instrucción RALRM se puede llamar en tres modos de operación diferentes (MODE).

Tabla 9-83 Modos de operación de la instrucción RALRM

MODE	Descripción
0	• ID contiene la ID de hardware del módulo de E/S que ha disparado la alarma. • El parámetro de salida NEW se establece en TRUE. • LEN produce una salida de 0. • AINFO y TINFO no se actualizan con ninguna información.
1	• ID contiene la ID de hardware del módulo de E/S que ha disparado la alarma. • El parámetro de salida NEW se establece en TRUE. • LEN produce una salida con la cantidad de bytes de los datos AINFO que se devuelven. • AINFO y TINFO se actualizan con la información de alarma.
2	Si la ID de hardware asignada al parámetro de entrada F_ID ha activado la alarma, entonces: • ID contiene la ID de hardware del módulo de E/S que ha disparado la alarma. Debe ser igual que el valor de F_ID. • El parámetro de salida NEW se establece en TRUE. • LEN produce una salida con la cantidad de bytes de los datos AINFO que se devuelven. • AINFO y TINFO se actualizan con la información de alarma.

Nota

Si se asigna un área de destino demasiado corta para TINFO o AINFO, RALRM no puede devolver la información completa.

MLEN puede limitar la cantidad de datos de AINFO que se devuelven.

Consulte los parámetros de AINFO y TINFO de la Ayuda on line de STEP 7 para obtener información sobre cómo interpretar los datos de TINFO y AINFO.

Datos de bloques de organización de TInfo

En la tabla se muestra cómo se organizan los datos de TInfo para la instrucción RALRM:

Igual para OB: estado, actualización, perfil, alarma de error de diagnóstico, presencia de módulo, fallo de rack o estación	0	SI_Format	OB_Class	OB_Nr
	4	LADDR		

TI_Submodule - OB: estado, actualización, perfil	4		Slot	
	8	Indicador	0	
TI_DiagnosticInterrupt - OB: Alarma de error de diagnóstico	4		IO_State	
	8	Canal	MultiError	0
TI_PlugPullModule - OB: Presencia de módulos	4		Event_Class	Fault_ID
	8	0	0	
TI_StationFailure - OB: Fallo del rack o estación	4		Event_Class	Fault_ID
	8	0	0	
Igual para OB: estado, actualización, perfil, alarma de error de diagnóstico, presencia de módulo, fallo de rack o estación	12	0		
	16			
	20	Dirección	slv_prfl	intr_type
	24	flags1 / flags2	ID	
	28[1]	Fabricante	Instancia	

[1] Los bytes 28 - 31 (fabricante e instancia) no se utilizan con PROFIBUS.

Nota

Consulte el sistema de información online de STEP 7 para obtener información más detallada sobre los datos de TINFO.

9.3.8 D_ACT_DP (activar/desactivar dispositivos PROFINET IO)

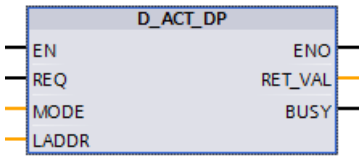
La instrucción "D_ACT_DP" permite desactivar y activar dispositivos PROFINET IO configurados de forma selectiva. También es posible determinar si cada dispositivo PROFINET IO asignado está activado o desactivado en esos momentos.

Nota

La instrucción D_ACT_DP solo puede utilizarse con dispositivos PROFINET IO. No es posible emplear la instrucción con esclavos PROFIBUS DP.

Tabla 9-84 Instrucción D_ACT_DP

KOP / FUP	SCL	Descripción

	`"D_ACT_DP_DB"(`
	`req:=_bool_in_,`
	`mode:=_usint_in_,`
	`laddr:=_uint_in_,`
	`ret_val=>_int_out_,`
	`busy=>_bool_out_);`
		La instrucción D_ACT_DP se utiliza para desactivar y activar dispositivos PROFINET IO configurados y determinar si cada dispositivo PROFINET IO asignado está activado o desactivado en esos momentos.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "D_ACT_DP_SFB_DB" es el nombre del DB de instancia.

No es posible desactivar/activar una pasarela del tipo IE/PB Link PN IO por medio de la instrucción D_ACT_DP. Si, de todas formas, se utiliza D_ACT_DP en la citada pasarela, la CPU devuelve el valor W#16#8093 (no hay ningún objeto de hardware que pueda activarse o desactivarse para la dirección especificada en LADDR).

Nota

La tarea de desactivación o activación requiere varias ejecuciones por el punto de control del ciclo. Por consiguiente, no es posible esperar a que finalice una tarea de este tipo en un bucle programado.

Descripción del funcionamiento

D_ACT_DP es una instrucción asíncrona, con lo que el procesamiento de la tarea abarca varias ejecuciones de D_ACT_DP. Inicie la tarea llamando D_ACT_DP con REQ = 1.

Los parámetros de salida RET_VAL y BUSY indican el estado de la tarea.

Aplicación

Si se configuran dispositivos PROFINET IO en una CPU que realmente no existe o que actualmente no se necesita, esta CPU sigue accediendo a los dispositivos PROFINET IO a intervalos regulares. Después de desactivar los dispositivos la CPU deja de acceder a ellos. Los respectivos eventos de error no se producen más.

Ejemplos

Desde el punto de vista de un fabricante de máquinas hay muchas opciones de dispositivos posibles en la fabricación de máquinas en serie. Sin embargo, cada máquina suministrada incluye únicamente una combinación de opciones determinada.

El fabricante configura cada una de estas opciones posibles para la máquina como un dispositivo PROFINET IO. El fabricante lo hace para crear y mantener un programa de usuario común que tenga todas las opciones posibles. D_ACT_DP se utiliza para desactivar todos los dispositivos PROFINET IO que no están presentes al arrancar la máquina.

Una situación similar se da con las máquinas herramienta que disponen de muchas opciones de herramientas pero que en realidad solo utilizan algunas de ellas en un momento determinado. Estas herramientas se implementan como dispositivos PROFINET IO. Con D_ACT_DP, el

programa de usuario activa las herramientas que se necesitan actualmente y desactiva las que se necesitarán más adelante.

Identificación de una tarea

Si se ha empezado una tarea de desactivación o activación y se vuelve a llamar D_ACT_DP antes de que finalice la tarea, el comportamiento de la instrucción depende de si la nueva llamada incluye la misma tarea o no. Si el parámetro de entrada LADDR coincide, la llamada se interpreta como subsiguiente.

Desactivar dispositivos PROFINET IO

Cuando se desactiva un dispositivo PROFINET con D_ACT_DP, las salidas del proceso del mismo se ponen a los valores sustitutivos configurados o a "0" (estado seguro). El controlador PROFINET IO asignado no sigue direccionando este componente. Los LED de error del controlador PROFINET IO o la CPU no identifican los dispositivos PROFINET IO desactivados como averiados o ausentes.

La CPU actualiza con "0" la memoria imagen de proceso de las entradas de los dispositivos PROFINET IO desactivados. Por consiguiente, la CPU trata los dispositivos PROFINET IO desactivados simplemente como averiados.

Si se accede directamente desde el programa a los datos de usuario de un dispositivo PROFINET IO previamente desactivado, el comportamiento del sistema depende del tratamiento de errores seleccionado para el bloque:

- Si el tratamiento de errores global está activado, el sistema introduce un evento de arranque para el error de acceso en el búfer de diagnóstico y permanece en RUN.
- Si el tratamiento de errores local de bloques está activado, el sistema introduce una causa del error en la estructura del error. Se accede a la causa del error por medio de la instrucción GET_ERROR_ID (Página 304).
  Un error de acceso de lectura devuelve "0". Encontrará más información sobre el tratamiento de errores en "Prioridades y colas de espera para la ejecución de eventos" (Página 87).

Si se intenta acceder a un dispositivo PROFINET IO desactivado por medio de una instrucción (p. ej. "RD_REC (Página 370)"), en RET_VAL se recibe la misma información de error que para un dispositivo PROFINET IO no disponible.

Si una estación PROFINET IO falla después de haberla desactivado con D_ACT_DP, el sistema operativo no detecta el fallo.

Activar dispositivos PROFINET IO

Cuando se reactiva un dispositivo PROFINET IO con D_ACT_DP, el controlador PROFINET IO asociado configura el componente y lo parametriza (igual que cuando retorna una estación PROFINET IO que había fallado). Esta activación termina cuando el componente está en disposición de transferir datos de usuario.

Si se intenta activar un dispositivo PROFINET IO al que no puede accederse (p. ej. porque estaba físicamente separado del bus) con una instrucción D_ACT_DP, la instrucción devuelve el código de error W#16#80A7 una vez ha finalizado el tiempo de parametrización configurado para periferia descentralizada. El dispositivo PROFINET IO se activa y, puesto que no es posible acceder a él, se muestra un aviso en correspondencia en el diagnóstico del sistema.

Si, más adelante, el dispositivo PROFINET IO vuelve a estar accesible, el comportamiento del sistema será el estándar.

Nota

La activación de un dispositivo PROFINET IO requiere cierto tiempo. Si desea cancelar una tarea de activación en curso, inicie D_ACT_DP con el mismo valor para LADDR y MODE = 2. Debe repetir la llamada de D_ACT_DP con MODE = 2 hasta que se muestre la cancelación correcta de la tarea de activación con RET_VAL = 0.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción D_ACT_DP:

Parámetro	Declaración	Tipo de datos	Descripción
REQ	IN	Bool	Parámetro de control disparado por nivel REQ = 1: Ejecutar activación o desactivación
MODE	IN	USInt	Identificador de la tarea Valores posibles: • 0: Solicitar información sobre si el componente direccionado está activado o desactivado (salida utilizando el parámetro RET_VAL) • 1: Activar el dispositivo PROFINET IO • 2: Desactivar el dispositivo PROFINET IO
LADDR	IN	HW_DEVICE	Identificador de hardware del dispositivo PROFINET IO (HW_Device) El número puede tomarse de las propiedades del dispositivo PROFINET IO en la Vista de redes o de la ficha "Constantes de sistema" de la tabla de variables estándar. Si tanto el identificador del diagnóstico de dispositivo como el identificador de las transiciones de estado operativo se especifican allí, hay que utilizar el código del diagnóstico de dispositivo.
RET_VAL	OUT	Int	Si se produce un error mientras el programa ejecuta la instrucción, el valor de retorno contiene un código de error.
BUSY	OUT	Bool	Código activo: • BUSY = 1: La tarea sigue activa. • BUSY = 0: La tarea ha finalizado.

Parámetro RET_VAL

Código de error* (W#16#...)	Explicación
0000	Tarea finalizada sin errores.
0001	El dispositivo PROFINET IO está activo (este código de error solo es posible si MODE = 0).
0002	El dispositivo PROFINET IO está desactivado (este código de error solo es posible si MODE = 0).

Código de error* (W#16#...)	**Explicación**
7000	Primera llamada con REQ = 0: La tarea especificada en LADDR no está activa; BUSY tiene el valor "0".
7001	Primera llamada con REQ = 1. El programa ha disparado la tarea especificada en LADDR. BUSY tiene el valor "1".
7002	Llamada intermedia (REQ irrelevante). La tarea activada sigue activa; BUSY tiene el valor "1".
8090	• No se ha configurado un módulo con la dirección especificada en LADDR. • La CPU funciona como esclavo I / I-device y se ha especificado una dirección de este esclavo I / I-device en LADDR.
8092	La desactivación del dispositivo PROFINET IO direccionado actualmente (MODE = 2) no puede cancelarse activándolo (MODE = 1). Active el componente más adelante.
8093	La dirección especificada en LADDR no pertenece a ningún dispositivo PROFINET IO que pueda activarse o desactivarse, o bien el parámetro MODE es desconocido.
8094	Se ha intentado activar un dispositivo que es un posible interlocutor de un puerto de cambio de herramientas. Sin embargo, actualmente ya hay otro dispositivo activado en este puerto de cambio de herramientas. El dispositivo activado permanece activo.
80A0	Error en la comunicación entre la CPU y el controlador IO.
80A1	No es posible asignar parámetros al componente direccionado. (Este código de error solo es posible si MODE = 1.) Nota: Si este componente vuelve a fallar durante la parametrización del dispositivo activado, la instrucción D_ACT_DP proporciona la información de error. Si la parametrización de un solo módulo falla, D_ACT_DP devuelve la información de error W#16#0000.
80A3	El controlador PROFINET IO correspondiente no soporta esta función.
80A4	La CPU no soporta esta función para un controlador PROFINET IO externo.
80A6	Error de slot en el dispositivo PROFINET IO; no es posible acceder a todos los datos de usuario (este código de error solo está disponible si MODE = 1). Nota: D_ACT_DP solo devuelve esta información de error si el componente activado vuelve a fallar después de la parametrización y antes de que finalice la ejecución de la instrucción D_ACT_DP. Si solo hay un módulo disponible, D_ACT_DP devuelve la información de error W#16#0000.
80A7	Tiempo excedido durante la activación: El dispositivo remoto no está accesible, o bien se ha ajustado un tiempo de parametrización demasiado corto para la periferia centralizada y descentralizada. El estado del dispositivo remoto es "activado", pero no está accesible.
80AA	Activación con errores en el dispositivo PROFINET IO: Diferencias en la configuración
80AB	Activación con errores en el dispositivo PROFINET IO: Error de parametrización
80AC	Activación con errores en el dispositivo PROFINET IO: Mantenimiento necesario
80C1	D_ACT_DP se ha iniciado y continúa con otra dirección (este código de error es posible si MODE = 1 y MODE = 2).
80C3	• Error de recurso temporal: La CPU está procesando el número máximo posible de tareas de activación y desactivación (8). (Este código de error solo es posible si MODE = 1 y MODE = 2.) • La CPU está ocupada recibiendo una configuración modificada. Actualmente no es posible activar/desactivar dispositivos PROFINET IO.
80C6	PROFINET: las tareas no recopiladas por el usuario se rechazan al reiniciar.

Código de error* **(W#16#...)**	**Explicación**
Información general del error	Encontrará información sobre cómo acceder al error en la instrucción GET_ERROR_ID (Página 304).
* Los códigos de error en el editor de programas se pueden mostrar como valores enteros o hexadecimales.	

9.3.9 Parámetro STATUS para RDREC, WRREC y RALRM

El parámetro de salida STATUS contiene información de error que se interpreta como ARRAY[1...4] OF BYTE con la siguiente estructura:

Tabla 9-85 Matriz de salida STATUS

Elemento de matriz	**Nombre**	**Descripción**
STATUS[1]	Function_Num	• B#16#00, si no hay errores • ID de función de DPV1-PDU: Si se produce un error, , B#16#80 se combina lógicamente con O (para leer registro: B#16#DE; para escribir registro: B#16#DF). Si no se utiliza ningún elemento de protocolo DPV1, se transferirá B#16#C0 .
STATUS[2]	Error_Decode	Ubicación del ID de error
STATUS[3]	Error_Code_1	ID de error
STATUS[4]	Error_Code_2	Extensión de ID de error específica del fabricante

Tabla 9-86 Valores de STATUS[2]

Error_decode (B#16#....)	**Fuente**	**Descripción**
00 a 7F	CPU	Sin errores o alarmas
80	DPV1	Error según CEI 61158-6
81 a 8F	CPU	B#16#8x muestra un error en el "enésimo" parámetro de llamada de la instrucción.
FE, FF	Perfil DP	Error específico de perfil

Tabla 9-87 Valores de STATUS[3]

Error_decode (B#16#....)	**Error_code_1 (B#16#....)**	**Explicación (DVP1)**	**Descripción**
00	00		Sin errores o alarmas
70	00	Reservado, rechazar	Llamada inicial; sin transferencia de registro activa
	01	Reservado, rechazar	Llamada inicial; transferencia de registro iniciada
	02	Reservado, rechazar	Llamada intermedia; la transferencia de registro ya está activa

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explicación (DVP1)	Descripción
80	90	Reservado, permitir	Dirección inicial lógica no válida
	92	Reservado, permitir	Tipo no permitido para puntero Variant
	93	Reservado, permitir	El componente DP direccionado vía ID o F_ID no está configurado.
	96		"RALRM (Página 377)" no puede suministrar la información de arranque del OB, la información de administración, de encabezado o de alarma adicional. Para los siguientes OB, se puede utilizar la instrucción "DPNRM_DG (Página 396)" para leer de forma asíncrona la trama actual del aviso de diagnóstico del esclavo DP relevante (información de dirección procedente de la información de arranque del OB): • Alarma de proceso (Página 78) • Estado (Página 84), Actualización (Página 84) o Perfil (Página 85) • Alarma de error de diagnóstico (Página 80) • Presencia de módulos (Página 82)
	A0	Error de lectura	Acuse negativo al leer del módulo
	A1	Error de escritura	Acuse negativo al escribir del módulo
	A2	Fallo del módulo	Error de protocolo DP en nivel 2 (por ejemplo, fallo de esclavo o problemas de bus)
	A3	Reservado, permitir	• PROFIBUS DP: Error de protocolo DP con Direct-Data-Link-Mapper o interfaz de usuario/usuario • PROFINET IO: Error general CM
	A4	Reservado, permitir	Fallo de comunicación en el bus de comunicación
	A5	Reservado, permitir	-
	A7	Reservado, permitir	Esclavo DP o módulo ocupado (error temporal).
	A8	Conflicto de versiones	Esclavo DP o módulo notifica versiones no compatibles.
	A9	Función no soportada	Función no soportada por esclavo DP o módulo
	AA a AF	Específico de usuario	El esclavo DP o módulo devuelve un error específico de fabricante en su aplicación. Consulte la documentación del fabricante del esclavo DP o módulo.
	B0	Índice no válido	El registro no se conoce en el módulo; número de registro no permitido ≥ 256
	B1	Error de longitud de escritura	La información de longitud en el parámetro RECORD es incorrecta. • Con "RALRM": error de longitud en AINFO **Nota**: consulte la Ayuda online de STEP 7 para acceder directamente a información sobre cómo interpretar los búferes "AINFO" devueltos. • Con "RDREC (Página 370)" y "WRREC (Página 370)": error de longitud en "MLEN"
	B2	Slot no válido	El slot configurado no está ocupado.
	B3	Conflicto de tipos	El tipo de módulo real no coincide con el especificado.
	B4	Área no válida	El esclavo DP o módulo notifica un acceso a un área no válida.

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explicación (DVP1)	Descripción
	B5	Conflicto de estado	El esclavo DP o módulo no está listo
	B6	Acceso denegado	El esclavo DP o módulo deniega el acceso.
	B7	Rango no válido	El esclavo DP o módulo notifica un rango no válido de un parámetro o valor.
	B8	Parámetro no válido	El esclavo DP o módulo notifica un parámetro no válido.
	B9	Tipo no válido	El esclavo DP o módulo notifica un tipo no válido: • Con "RDREC (Página 370)": búfer demasiado pequeño (no es posible leer partes de un campo) • Con "WRREC (Página 370)": búfer demasiado pequeño (no es posible escribir en partes de un campo)
	BA a BF	Específico de usuario	El esclavo DP o módulo devuelve un error específico de fabricante al acceder. Consulte la documentación del fabricante del esclavo DP o módulo.
	C0	Conflicto de restricción de lectura	• Con "WRREC (Página 370)": Los datos sólo se pueden escribir cuando la CPU se encuentra en estado operativo STOP. **Nota**: Ello significa que los datos no pueden escribirse con el programa de usuario. Sólo se pueden escribir datos online con una programadora o un PC. • Con "RDREC (Página 370)": el módulo enruta el registro, pero no hay datos o los datos sólo pueden leerse cuando la CPU está en estado operativo STOP. **Nota**: Si los datos sólo pueden leerse cuando la CPU está en modo STOP, no es posible una evaluación por parte del programa de usuario. En ese caso, sólo pueden leerse datos online con una programadora o un PC.
	C1	Conflicto de restricción de escritura	El módulo aún no ha procesado los datos de la solicitud anterior de escritura en el módulo.
	C2	Recurso ocupado	Actualmente el módulo está procesando el número máximo de tareas posibles para una CPU.
	C3	Recurso no disponible	Los recursos requeridos están ocupados en este momento.
	C4		Error temporal interno. No ha sido posible realizar la tarea. Repetir la tarea. Si se produce este error, compruebe la instalación en busca de fuentes de perturbación eléctrica.
	C5		Esclavo DP o módulo no disponible
	C6		La transferencia del registro se ha cancelado debido a la cancelación por clase de prioridad
	C7		Tarea cancelada debido a rearranque en caliente o en frío del maestro DP.
	C8 a CF		El esclavo DP o módulo devuelve un error de recurso específico de fabricante. Consulte la documentación del fabricante del esclavo DP o módulo.
	Dx	Específico de usuario	Específico de esclavo DP. Véase la descripción del esclavo DP.
81	00 a FF		Error en el parámetro de llamada inicial (con "RALRM (Página 377)": MODE)
	00		Estado operativo no válido
82	00 a FF		Error en el segundo parámetro de llamada

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Explicación (DVP1)	Descripción
88	00 a FF		Error en el octavo parámetro de llamada (con "RALRM (Página 377)": TINFO) **Nota**: consulte la Ayuda online de STEP 7 para acceder directamente a información sobre cómo interpretar los búferes "TINFO" devueltos.
	01		ID de sintaxis errónea
	23		Estructura de cantidad excedida o área de destino demasiado pequeña
	24		ID de rango errónea
	32		Número de DB/DI fuera del rango de usuario
	3A		El número de DB/DI es CERO para la ID de área DB/DI o el DB/DI especificado no existe.
89	00 a FF		Error en el noveno parámetro de llamada (con "RALRM (Página 377)": AINFO) **Nota**: consulte la Ayuda online de STEP 7 para acceder directamente a información sobre cómo interpretar los búferes "AINFO" devueltos.
	01		ID de sintaxis errónea
	23		Estructura de cantidad excedida o área de destino demasiado pequeña
	24		ID de rango errónea
	32		Número de DB/DI fuera del rango de usuario
	3A		El número de DB/DI es CERO para la ID de área DB/DI o el DB/DI especificado no existe.
8A	00 a FF		Error en el 10.º parámetro de llamada
8F	00 a FF		Error en el 15.º parámetro de llamada
FE, FF	00 a FF		Error específico de perfil

Elemento de matriz STATUS[4]

Con errores DPV1, el maestro DP transfiere en STATUS[4] a la CPU y a la instrucción. Sin un error de DPV1, este valor se pone a 0 con las siguientes excepciones para RDREC:

- STATUS[4] contiene la longitud del área de destino de RECORD, si MLEN > la longitud del área de destino de RECORD.
- STATUS[4]=MLEN, si la longitud real del registro < MLEN < la longitud del área de destino de RECORD.
- STATUS[4]=0, si STATUS[4] > 255; debe estar activado

En PROFINET IO, STATUS[4] tiene el valor 0.

9.3.10 Otras

9.3.10.1 DPRD_DAT y DPWR_DAT (leer/escribir datos coherentes)

La instrucción DPRD_DAT (leer datos coherentes) se utiliza para leer uno o varios bytes de datos de forma coherente y la instrucción DPWR_DAT (escribir datos coherentes) se utiliza para transferir uno o varios bytes de datos de forma coherente. Las instrucciones DPRD_DAT y DPWR_DAT se pueden utilizar con PROFINET y PROFIBUS.

Tabla 9-88 Instrucciones DPRD_DAT y DPWR_DAT

KOP / FUP	SCL	Descripción

	`ret_val := DPRD_DAT(`
	`laddr:=_word_in_,`
	`record=>_variant_out_);`
		La instrucción DPRD_DAT se utiliza para leer uno o más bytes de datos de módulos o submódulos de una de las siguientes ubicaciones:
		• E/S básicas locales
		• Esclavo DP
		• dispositivo PROFINET I/O
		La CPU transfiere los datos leídos de forma coherente. Si no ocurre ningún error durante la transferencia de datos, la CPU introduce los datos leídos en el área de destino especificada por el parámetro RECORD. El área de destino debe tener la misma longitud que la configurada con STEP 7 para el módulo seleccionado. Al ejecutar la instrucción DPRD_DAT solo se puede acceder a los datos de un módulo o submódulo. La transferencia comienza en la dirección inicial configurada.

	`ret_val := DPWR_DAT(`
	`laddr:=_word_in_,`
	`record:=_variant_in_);`
		Utilice la instrucción DPWR_DAT para transferir los datos de RECORD de forma coherente a las siguientes ubicaciones:
		• módulo o submódulo direccionado en la base local
		• esclavo DP normalizado
		• dispositivo PROFINET I/O
		El área de origen debe tener la misma longitud que la configurada con STEP 7 para el módulo o submódulo seleccionado.

- La CPU S7-1200 soporta la lectura o escritura coherentes de 1, 2 o 4 bytes en la periferia descentralizada. La instrucción DPRD_DAT se utiliza para leer de forma coherente datos de longitudes diferentes de 1, 2 o 4 bytes y la instrucción DPWR_DAT para escribirlos de forma coherente.
- Estas instrucciones se pueden utilizar para áreas de datos de 1 o más bytes. Si el acceso se deniega, se devuelve el código de error W#16#8090.
- PROFINET soporta hasta 1024 bytes de datos coherentes. No es necesario utilizar estas instrucciones para realizar una transferencia coherente entre S7-1200 y los dispositivos PROFINET.

Nota

Si se utiliza las instrucciones DPRD_DAT y DPWR_DAT con datos coherentes, hay que retirar los datos coherentes de la actualización automática de la memoria imagen de proceso.
Véase "Conceptos de PLC: Ejecución del programa de usuario" (Página 67) para obtener más información.

Tabla 9-89 Parámetros

Parámetro	Declaración	Tipo de datos	Descripción
LADDR	IN	HW_IO (Word)	ID de hardware del módulo del cual se van a leer los datos. (DPRD_DAT) ID de hardware del módulo en el cual se van a escribir los datos. (DPWR_DAT) Encontrará la ID de hardware en la vista de dispositivos o las constantes de sistema de las propiedades del módulo.
RECORD	OUT	Variant	Área de destino para los datos de usuario que se han leído (DPRD_DAT) o área de origen de los datos de usuario que deben escribirse (DPWR_DAT). Este debe tener exactamente la misma longitud que la configurada con STEP 7 para el módulo seleccionado.
RET_VAL	OUT	Int	Si se produce un error mientras la función está activa, el valor de retorno contiene un código de error.

Operaciones DPRD_DAT

Utilice el parámetro LADDR para seleccionar el módulo del esclavo DP normalizado / dispositivo PROFINET IO. Si se produce un error de acceso en el módulo direccionado, se emite el código de error W#16#8090.

Utilice el parámetro RECORD para definir el rango de destino de los datos leídos:

- El rango de destino debe tener por lo menos la misma longitud que las entradas del módulo seleccionado. Solo se transfieren las entradas, los otros bytes se ignoran. Si se lee de un esclavo DP normalizado con un diseño modular o con varios identificadores DP, solo se puede acceder a los datos de un módulo del identificador de hardware configurado por cada llamada de la instrucción DPRD_DAT. Si se selecciona un rango de destino demasiado pequeño, se emite el código de error W#16#80B1 en el parámetro RET_VAL.
- Se pueden usar los tipos de datos siguientes: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Está permitido usar estos tipos de datos en una estructura de tipo de datos personalizado (UDT) del tipo ARRAY o STRUCT.
- El tipo de datos STRING no se soporta.
- Si no hay errores durante la transmisión de datos, los datos leídos se introducen en el rango de destino definido en el parámetro RECORD.

Operaciones DPWR_DAT

Utilice el parámetro LADDR para seleccionar el módulo del esclavo DP normalizado / dispositivo PROFINET IO. Si se produce un error de acceso en el módulo direccionado, se emite el código de error W#16#8090.

Utilice el parámetro RECORD para definir el rango de origen de los datos que deben escribirse:

- El rango de origen debe tener por lo menos la misma longitud que las salidas del módulo seleccionado. Solo se transfieren las salidas, los otros bytes se ignoran. Si el rango de origen en el parámetro RECORD es mayor que las salidas del módulo configurado, solo se transfieren los datos hasta la longitud máxima de las salidas. Si el rango de origen en el parámetro RECORD es menor que las salidas del módulo configurado, se emite el código de error W#16#80B1 en el parámetro RET_VAL.
- Se pueden usar los tipos de datos siguientes: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Está permitido usar estos tipos de datos en una estructura de tipo de datos personalizado (UDT) del tipo ARRAY o STRUCT.
- El tipo de datos STRING no se soporta.
- Los datos se transfieren de forma asíncrona, de modo que el proceso de escritura se completa cuando se completa la instrucción.

Códigos de error

Tabla 9-90 Códigos de error DPRD_DAT y DPWR_DAT

Código de error [1]	Descripción
0000	No ha ocurrido ningún error
8090	Se da uno de los siguientes casos: • No ha configurado un módulo para la dirección base lógica especificada. • Ha ignorado la restricción referente a la longitud de los datos coherentes. • Ha introducido la dirección inicial en el parámetro LADDR en formato hexadecimal.
8092	El parámetro RECORD soporta los siguientes tipos de datos: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.
8093	No hay ningún módulo DP o dispositivo PROFINET IO del que se pueden leer (DPRD_DAT) o en el que se pueden escribir (DPWR_DAT) datos coherentes en la dirección lógica especificada en LADDR.
80A0	Error de acceso detectado durante el acceso a los dispositivos I/O (DPRD_DAT).
80B1	La longitud del área de destino (DPRD_DAT) o de origen (DPWR_DAT) especificada coincide con la longitud de datos de usuario configurada con STEP 7 Basic.
80B2	Error de sistema con módulo interfaz DP externo (DPRD_DAT) y (DPWR_DAT)

[1] Utilice el tipo de datos DWord para visualizar los códigos de error de DPRD_DAT y DPWR_DAT.

Nota

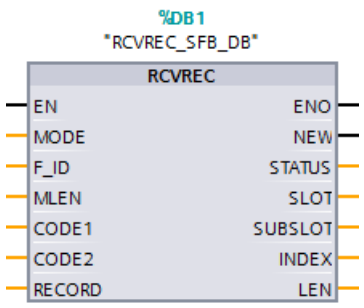
Si se accede a esclavos DPV1, la información de error de estos puede transmitirse del maestro DP a la instrucción.

9.3.10.2 RCVREC (recibir registro con I-device/esclavo I)

Un I-device puede recibir un registro de un controlador superior. La recepción tiene lugar en el programa de usuario con la instrucción RCVREC (recibir registro).

Tabla 9-91 Instrucción RCVREC

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción RCVREC se utiliza para recibir un registro de un controlador superior.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "RCVREC_SFB_DB" es el nombre del DB de instancia.

La instrucción tiene los siguientes estados operativos:

- Comprobar si el I-device tiene una orden para recibir un registro
- Poner el registro a disposición de los parámetros de salida
- Enviar una respuesta al controlador superior

El estado operativo que ejecuta la instrucción se determina mediante el parámetro de entrada MODE.

El I-device debe estar en estado RUN o ARRANQUE.

Con MLEN se especifica el número máximo de bytes que deben recibirse. La longitud seleccionada del rango de destino RECORD debe tener como mínimo la longitud de MLEN bytes.

Si se recibe un registro (MODE = 1 o MODE = 2), el parámetro de salida NEW indica que el registro está almacenado en RECORD. Asegúrese de que RECORD tenga la longitud suficiente. El parámetro de salida LEN contiene la longitud real del registro recibido en bytes.

Ponga CODE1 y CODE2 a cero para la respuesta positiva al controlador superior. Si el registro recibido debe rechazarse, introduzca la respuesta negativa al controlador superior en Error Code 1 de CODE1 y en Error Code 2 de CODE2.

Nota

Si el I-device ha recibido una orden para recibir un registro, hay que acusar la entrega de esta orden en un tiempo determinado. Una vez acusada, hay que enviar una respuesta al controlador superior dentro de este tiempo. En caso contrario, el I-device sufre un error de tiempo excedido que hace que el sistema operativo del I-device envíe una respuesta negativa al controlador superior. Encontrará información sobre el valor de tiempo en las especificaciones de la CPU.

El parámetro de salida STATUS recibe la información de error después de haberse producido un error.

Estados operativos

El estado operativo de la instrucción RCVREC se determina con el parámetro de entrada MODE. Este paso se explica en la tabla siguiente:

MODE	Significado
0	Comprobar si hay una orden para recibir un registro Si en el I-device hay un registro de un controlador superior, la instrucción solo escribe en los parámetros de salida NEW, SLOT, SUBSLOT, INDEX y LEN. Si se llama la instrucción varias veces con MODE = 0, el parámetro solo hace referencia una misma orden.
1	Recibir un registro para cualquier subslot del I-device Si en el I-device hay un registro de un controlador superior para cualquier subslot del I-device, la instrucción escribe en el parámetro de salida y transfiere el registro al parámetro RECORD.
2	Recibir un registro para un subslot concreto del I-device Si en el I-device hay un registro de un controlador superior para un subslot concreto del I-device, la instrucción escribe en el parámetro de salida y transfiere el registro al parámetro RECORD.
3	Enviar una respuesta positiva al controlador superior La instrucción comprueba la orden del controlador superior para recibir un registro, acepta el registro existente y envía un acuse positivo al controlador superior.
4	Enviar una respuesta negativa al controlador superior La instrucción comprueba la orden del controlador superior para recibir un registro, rechaza el registro existente y envía un acuse negativo al controlador superior. Introduzca el motivo del rechazo en los parámetros de entrada CODE1 y CODE2.

Nota

Después de recibir un registro (NEW = 1) hay que llamar la instrucción RCVREC dos veces para garantizar su procesamiento completo. Esto debe realizarse en el orden siguiente:

- Primera llamada con MODE = 1 o MODE = 2
- Segunda llamada con MODE = 3 o MODE = 4

Parámetros

La tabla siguiente muestra los parámetros de la instrucción RCVREC:

Parámetro	Declaración	Tipo de datos	Descripción
MODE	IN	Int	Modo
F_ID	IN	HW_SUBMODULE	Subslot en el área de transferencia del I-device para el registro que debe recibirse (solo relevante si MODE = 2). La palabra "high" siempre se pone a cero.
MLEN	IN	Int	Longitud máxima en bytes del registro que debe recibirse
CODE1	IN	Byte	Cero (si MODE = 3) y/o Error Code 1 (si MODE = 4)
CODE2	IN	Byte	Cero (si MODE = 3) y/o Error Code 2 (si MODE = 4)

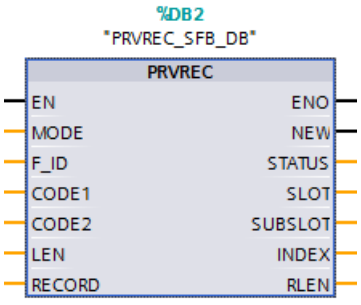
Paráme-tro	Declara-ción	Tipo de datos	Descripción
NEW	OUT	Bool	• MODE = 0: Se ha recibido un nuevo registro • MODE = 1 o 2: El registro se ha transferido a RECORD
STATUS	OUT	DWord	Información de error. Encontrará más información en "Parámetro STATUS" (Página 385).
SLOT	OUT	HW_SUBMODULE	Idéntico a F_ID
SUBSLOT	OUT	HW_SUBMODULE	Idéntico a F_ID
INDEX	OUT	UInt	Número del registro recibido
LEN	OUT	UInt	Longitud del registro recibido
RECORD	IN_OUT	Variant	Rango de destino del registro recibido

9.3.10.3 PRVREC (proveer registro con I-device/esclavo I)

Un I-device puede recibir una orden de un controlador superior para proveer un registro. El I-device provee el registro en el programa de usuario con la instrucción PRVREC (proveer registro).

Tabla 9-92 Instrucción PRVREC

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción PRVREC se utiliza para recibir una orden de un controlador superior para proveer un registro.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "PRVREC_SFB_DB" es el nombre del DB de instancia.

La instrucción tiene los siguientes estados operativos:

- Comprobar si el I-device tiene una orden para proveer un registro
- Transferir el registro solicitado al controlador superior
- Enviar una respuesta al controlador superior

El estado operativo que ejecuta la instrucción se determina mediante el parámetro de entrada MODE.

El I-device debe estar en estado RUN o ARRANQUE.

Introduzca el número máximo de bytes que debe tener el registro que se enviará con LEN. La longitud seleccionada del rango de destino RECORD debe tener como mínimo la longitud de LEN bytes.

Si hay una orden para proveer un registro (MODE = 0), el parámetro de salida NEW se pone a TRUE.

Si se acepta la orden para proveer un registro, escriba RECORD para la respuesta positiva al controlador superior con el registro solicitado y cero para CODE1 y CODE2. Si la orden para proveer un registro debe rechazarse, introduzca la respuesta negativa al controlador superior en Error Code 1 de CODE1 y en Error Code 2 de CODE2.

Nota

Si el I-device ha recibido una orden para proveer un registro, hay que acusar la entrega de esta orden en un tiempo determinado. Una vez acusada, hay que enviar una respuesta al controlador superior dentro de este tiempo. En caso contrario, el I-device sufre un error de tiempo excedido que hace que el sistema operativo del I-device envíe una respuesta negativa al controlador superior. Encontrará información sobre el valor de tiempo en las especificaciones de la CPU.

El parámetro de salida STATUS recibe la información de error después de haberse producido un error.

Estados operativos

El estado operativo de la instrucción PRVREC se determina con el parámetro de entrada MODE. Este paso se explica en la tabla siguiente:

MODE	Significado
0	Comprobar si existe una orden para proveer un registro Si en el I-device hay una orden de un controlador superior para proveer un registro, la instrucción solo escribe en los parámetros de salida NEW, SLOT, SUBSLOT, INDEX y RLEN. Si se llama la instrucción varias veces con MODE = 0, el parámetro solo hace referencia una misma orden.
1	Recibir una orden para proveer un registro para cualquier subslot del I-device Si en el I-device hay una orden de este tipo de un controlador superior para cualquier subslot del I-device, la instrucción escribe en el parámetro de salida.
2	Recibir una orden para proveer un registro para un subslot concreto del I-device Si en el I-device hay una orden de este tipo de un controlador superior para un subslot concreto del I-device, la instrucción escribe en el parámetro de salida.
3	Proveer el registro y enviar una respuesta positiva al controlador superior La instrucción comprueba la orden del controlador superior para proveer un registro, proporciona el registro solicitado a RECORD y envía un acuse positivo al controlador superior.
4	Enviar una respuesta negativa al controlador superior La instrucción comprueba la orden del controlador superior para proveer un registro, rechaza la orden y envía un acuse negativo al controlador superior. Introduzca el motivo del rechazo en los parámetros de entrada CODE1 y CODE2.

Nota

Después de recibir una orden (NEW = 1) hay que llamar la instrucción PRVREC dos veces para garantizar su procesamiento completo. Esto debe realizarse en el orden siguiente:

- Primera llamada con MODE = 1 o MODE = 2
- Segunda llamada con MODE = 3 o MODE = 4

Parámetros

La tabla siguiente muestra los parámetros de la instrucción PRVREC:

Parámetro	Declaración	Tipo de datos	Descripción
MODE	IN	Int	Modo
F_ID	IN	HW_SUBMODULE	Subslot en el área de transferencia del I-device para el registro que debe enviarse (solo relevante si MODE = 2). La palabra "high" siempre se pone a cero.
CODE1	IN	Byte	Cero (si MODE = 3) y/o Error Code 1 (si MODE = 4)
CODE2	IN	Byte	Cero (si MODE = 3) y/o Error Code 2 (si MODE = 4)
LEN	IN	UInt	Longitud máxima en bytes del registro que debe enviarse
NEW	OUT	Bool	El controlador superior ha solicitado el nuevo registro.
STATUS	OUT	DWord	Información de error. Encontrará más información en "Parámetro STATUS" (Página 385).
SLOT	OUT	HW_SUBMODULE	Idéntico a F_ID
SUBSLOT	OUT	HW_SUBMODULE	Idéntico a F_ID
INDEX	OUT	UInt	Número del registro que debe enviarse
RLEN	OUT	UInt	Longitud del registro que debe enviarse
RECORD	IN_OUT	Variant	Registro provisto

9.3.10.4 DPNRM_DG (leer datos de diagnóstico de un esclavo DP)

La instrucción DPNRM_DG (Leer datos de diagnóstico de un esclavo DP) puede utilizarse con PROFIBUS.

Tabla 9-93 Instrucción DPNRM_DG

KOP / FUP	SCL	Descripción

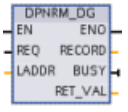
	`ret_val := DPNRM_DG(`
	`  req:=_bool_in_,`
	`  laddr:=_word_in_,`
	`  record=>_variant_out_,`
	`  busy=>_bool_out_);`
		Utilice la instrucción DPNRM_DG para leer los datos de diagnóstico actuales de un esclavo DP en el formato especificado por EN 50 170 volumen 2, PROFIBUS. Los datos leídos se introducen en el área de destino indicada en RECORD tras una transferencia sin errores.

Tabla 9-94 Tipos de datos de la instrucción DPNRM_DG para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	REQ=1: Solicitud de lectura
LADDR	IN	HW_DPSLAVE	Dirección de diagnóstico configurada para el esclavo DP: tiene que ser la dirección de la estación y no la del dispositivo I/O. Seleccione la estación (y no la imagen del dispositivo) en la vista "Redes" de la "Configuración de dispositivos" para determinar la dirección de diagnóstico. Introduzca las direcciones en formato hexadecimal. Por ejemplo, la dirección de diagnóstico 1022 significa LADDR:=W#16#3FE.

Parámetro y tipo		Tipo de datos	Descripción
RET_VAL	OUT	Int	Si se produce un error mientras la función está activa, el valor de retorno contiene un código de error. Si no se produce ningún error, la longitud real de los datos transferidos se deposita en RET_VAL.
RECORD	OUT	Variant	Área de destino para los datos de diagnóstico leídos. La longitud mínima del registro que debe leerse (o del área de destino) es de 6 bytes. La longitud máxima del registro que debe transferirse es de 240 bytes. Los esclavos estándar pueden proporcionar más de 240 bytes de datos de diagnóstico, hasta un máximo de 244 bytes. En ese caso, los primeros 240 bytes se transfieren al área de destino y en los datos se activa el bit de desbordamiento.
BUSY	OUT	Bool	BUSY=1: la tarea de lectura todavía no se ha completado

La tarea de lectura se inicia asignando 1 al parámetro de entrada REQ en la llamada de la instrucción DPNRM_DG. La tarea de lectura se ejecuta de forma asíncrona, lo que significa que requiere varias llamadas de la instrucción DPNRM_DG. El estado de la tarea se indica en los parámetros de salida RET_VAL y BUSY.

Tabla 9-95 Estructura de datos del diagnóstico de esclavo

Byte	Descripción
0	Estado de estación 1
1	Estado de estación 2
2	Estado de estación 3
3	Número de estación maestra
4	ID del fabricante (byte high)
5	ID del fabricante (byte low)
6 ...	Información de diagnóstico adicional específica de esclavo

Tabla 9-96 Códigos de error de la instrucción DPNRM_DG

Código de error	Descripción	Restricción
0000	No hay error	-
7000	Primera llamada con REQ=0: Ninguna transferencia de datos activa; BUSY tiene el valor 0.	-
7001	Primera llamada con REQ=1: Ninguna transferencia de datos activa; BUSY tiene el valor 1.	E/S descentralizadas
7002	Llamada intermedia (REQ irrelevante): Transferencia de datos ya activa; BUSY tiene el valor 1.	E/S descentralizadas
8090	Dirección lógica base especificada no válida: No ha dirección base.	-
8092	El parámetro RECORD admite los tipos de datos siguientes: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.	-

Código de error	Descripción	Restricción
8093	• Esta instrucción no está permitida para el módulo especificado en LADDR (se permiten módulos DP S7 para S7-1200). • LADDR especifica el dispositivo I/O en lugar de especificar la estación. Seleccione la estación (y no la imagen del dispositivo) en la vista "Redes" de la "Configuración de dispositivos" para determinar la dirección de diagnóstico de LADDR.	-
80A2	• Error de protocolo DP en nivel 2 (por ejemplo, fallo de esclavo o problemas de bus) • Para ET200S los registros no pueden leerse en modo DPV0.	E/S descentralizadas
80A3	Error de protocolo DP con interfaz de usuario/usuario	E/S descentralizadas
80A4	Problema de comunicación en el bus de comunicación	El error se produce entre la CPU y el modulo de interfaz DP externo.
80B0	• La instrucción no es posible para el tipo de módulo. • El modulo no reconoce el registro. • El número de registro 241 no está permitido.	-
80B1	La longitud especificada en el parámetro RECORD es incorrecta.	Longitud especificada > longitud de registro
80B2	El slot configurado no está ocupado.	-
80B3	El tipo de módulo real no coincide con el requerido	-
80C0	No hay información de diagnóstico.	-
80C1	El módulo aún no ha procesado los datos de la tarea de escritura anterior para el mismo registro.	-
80C2	Actualmente el módulo está procesando el número máximo de tareas posibles para una CPU.	-
80C3	Los recursos requeridos (memoria, etc.) están ocupados en este momento.	-
80C4	Error temporal interno. La tarea no se ha podido procesar. Repetir la tarea. Si este error se produce frecuentemente, compruebe el sistema en busca de fuentes de interferencia eléctrica.	-
80C5	E/S descentralizadas no disponibles	E/S descentralizadas
80C6	La transferencia del registro se ha interrumpido debido a una interrupción por clase de prioridad (rearranque o ejecución de fondo)	E/S descentralizadas
8xyy[1]	Códigos de error generales	

Consulte "Instrucciones avanzadas, E/S descentralizadas: Información de error para RDREC, WRREC y RALRM" (Página 385) para más información sobre códigos de error generales.

9.4 PROFIenergy

PROFIenergy es un perfil independiente del fabricante y del dispositivo para la gestión de la energía con PROFINET. Con el fin de reducir el consumo eléctrico durante las pausas de producción y las interrupciones imprevistas, existe la posibilidad de apagar equipos de una forma coordinada y centralizada por medio de PROFIenergy.

El controlador PROFINET IO desconecta los dispositivos PROFINET/módulos de potencia utilizando comandos especiales del programa de usuario. No se requiere hardware adicional. Los dispositivos PROFINET interpretan directamente los comandos de PROFIenergy.

La CPU S7-1200 no soporta la funcionalidad de controlador PE. La CPU S7-1200 solo puede funcionar como unidad PROFIenergy (con funcionalidad de I-device).

Controlador PROFIenergy (controlador PE)

El controlador PE es un PLC de nivel superior (p. ej. un S7-1500) que activa o desactiva el estado de reposo de los dispositivos de nivel inferior. El controlador PE desactiva y reactiva componentes de producción individuales o bien líneas de producción completas desde el programa de usuario. Los dispositivos de nivel inferior reciben comandos del programa de usuario a través de las instrucciones correspondientes (bloques de función).

El programa de usuario envía los comandos por medio del protocolo de comunicación PROFINET. El comando PE puede ser un comando de control para conmutar una unidad PE al modo de ahorro de energía o bien un comando para leer un estado o valor medido.

Utilice la instrucción PE_I_DEV para solicitar datos de un módulo. El programa de usuario debe determinar la información que solicita el controlador PE y recuperarla del módulo de energía por medio de registros. El módulo propiamente dicho no soporta directamente los comandos PE. El módulo guarda la información sobre la medición de energía en un área compartida y la CPU de nivel inferior (p. ej. un S7-1200) dispara la instrucción PE_I_DEV para devolverla al controlador PE.

Unidad PROFIenergy (unidad PE)

La unidad PE (p. ej. un S7-1200) recibe los comandos PROFIenergy del controlador PE (p. ej. un S7-1500) y los ejecuta en correspondencia (p. ej. devolviendo un valor medido o activando el modo de ahorro de energía). La implementación de la unidad PE en un dispositivo apto para PROFIenergy depende del dispositivo y del fabricante.

Información de referencia

Encontrará más información sobre PROFIenergy en la Ayuda en pantalla de STEP 7 (TIA Portal). Encontrará ejemplos de uso de las instrucciones PROFIenergy en la entrada "PROFIenergy - Ahorrar energía con SIMATIC S7 (http://support.automation.siemens.com/WW/view/es/41986454)" de Industry Online Support.

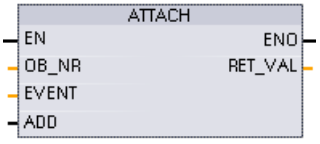
9.5 Alarmas

9.5.1 Instrucciones ATTACH y DETACH (Asignar OB a evento de alarma/deshacer asignación)

Las instrucciones ATTACH y DETACH permiten activar y desactivar subprogramas controlados por eventos de alarma.

Tabla 9-97 Instrucciones ATTACH y DETACH

KOP / FUP	SCL	Descripción

	`ret_val := ATTACH(`
	`  ob_nr:=_int_in_,`
	`  event:=_event_att_in_,`
	`  add:=_bool_in_);`
		ATTACH habilita la ejecución de un subprograma de OB de alarma para un evento de alarma de proceso.

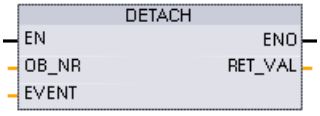
	`ret_val := DETACH(`
	`  ob_nr:=_int_in_,`
	`  event:=_event_att_ in);`
		DETACH deshabilita la ejecución de un subprograma de OB de alarma para un evento de alarma de proceso.

Tabla 9-98 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_ATT	Identificador del bloque de organización: Seleccione uno de los OBs de alarma de proceso que fueron creados utilizando la función "Agregar nuevo bloque". Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los OBs disponibles.
EVENT	IN	EVENT_ATT	Identificador del evento: Seleccione uno de los eventos de alarma de proceso habilitados en la configuración de dispositivos PLC para las entradas digitales o los contadores rápidos. Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los eventos disponibles.
ADD (solo ATTACH)	IN	Bool	• ADD = 0 (predeterminado): Este evento reemplaza a todas las asignaciones de eventos anteriores de este OB. • ADD = 1: Este evento se agrega a asignaciones de eventos anteriores de este OB.
RET_VAL	OUT	Int	Código de condición de ejecución

Eventos de alarma de proceso

La CPU soporta los siguientes eventos de alarma de proceso:

- Eventos de flanco ascendente: primeras 12 entradas digitales de la CPU integradas (de DIa.0 a DIb.3) y todas las entradas digitales de SB
  - Un flanco ascendente ocurre cuando la entrada digital cambia de OFF a ON como reacción a una modificación de la señal de un aparato de campo conectado a la entrada.
- Eventos de flanco descendente: primeras 12 entradas digitales de la CPU integradas (de DIa.0 a DIb.3) y todas las entradas digitales de SB
  - Un flanco descendente ocurre cuando la entrada digital cambia de ON a OFF.
- Eventos en los que el valor actual del contador rápido (HSC) = valor de referencia (CV = RV) (HSC 1 a 6)
  - Una alarma CV = RV de un HSC se genera cuando el contaje actual cambia de un valor adyacente al valor que concuerda exactamente con un valor de referencia establecido previamente.
- Eventos de cambio de sentido del HSC (HSC 1 a 6)
  - Un evento de cambio de sentido ocurre cuando se detecta que el sentido de contaje del HSC cambia de ascendente a descendente, o viceversa.
- Eventos de reset externo del HSC (HSC 1 a 6)
  - Algunos modos del HSC permiten asignar una entrada digital como reset externo para poner a cero el valor de contaje del HSC. Un evento de reset externo del HSC ocurre cuando esta entrada cambia de OFF a ON.

Habilitar eventos de alarma de proceso en la configuración de dispositivos

Las alarmas de proceso deben habilitarse durante la configuración de dispositivos. La casilla de habilitación de eventos se debe activar en la configuración de eventos para un canal de entrada digital o un HSC si este evento debe asignarse durante la configuración o en runtime.

Opciones de las casillas de verificación en la configuración de dispositivos PLC:

- Entrada digital
  - Habilitar detección de flancos ascendentes
  - Habilitar detección de flancos descendentes
- Contador rápido (HSC)
  - Habilitar este contador rápido para su uso
  - Generar alarma si el valor del contador es igual al valor de referencia
  - Generar alarma en caso de evento de reset externo
  - Generar alarma en caso de evento de cambio de sentido

Agregar OBs de alarma de proceso nuevos al programa de usuario

De forma predeterminada, ningún OB está asignado a un evento cuando este se habilita por primera vez. Esto se indica en la etiqueta "<no conectado>" de la configuración de dispositivos para "Alarma de proceso:". Solo los OB de alarma de proceso pueden asignarse a un evento de alarma de proceso. Todos los OB de alarma de proceso existentes se muestran en la lista desplegable "Alarma de proceso:". Si no se lista ningún OB, es preciso crear un OB de "Alarma de proceso" como se indica a continuación. En la rama "Bloques de programa" del árbol del proyecto:

1. Haga doble clic en "Agregar nuevo bloque", seleccione "Bloque de organización (OB)" y elija "Hardware interrupt" (Alarma de proceso).
2. Opcionalmente, puede cambiar el nombre del OB, elegir el lenguaje de programación (KOP, FUP o SCL) y seleccionar el número de bloque (cambiar a modo manual y seleccionar un número de bloque diferente del propuesto).
3. Edite el OB y agregue la reacción programada que debe ejecutarse cuando ocurra el evento. Puede llamar FC y FB desde este OB, hasta la profundidad de anidamiento máxima. La profundidad de anidamiento máxima es de cuatro para programas de seguridad. Para el resto de programas, la profundidad de anidamiento máxima es de seis.

Parámetro OB_NR

Todos los nombres de OB de alarma de proceso existentes se muestran en la lista desplegable "Alarma de proceso:" de la configuración de dispositivos y en la lista desplegable OB_NR de los parámetros ATTACH /DETACH.

Parámetro EVENT

Si está habilitado un evento de alarma de proceso, se le asigna un nombre predeterminado y unívoco. El nombre del evento puede cambiarse editándolo en el cuadro "Nombre de evento:", pero debe ser un nombre unívoco. Los nombres de eventos se convierten en nombres de variables en la tabla de variables "Constantes" y aparecen en la lista desplegable del parámetro EVENT, en los cuadros de las instrucciones ATTACH y DETACH. El valor de la variable es un número interno utilizado para identificar el evento.

Funcionamiento general

Todo evento de hardware puede asignarse a un OB de alarma de proceso que se pondrá en cola de espera para ser ejecutado cuando ocurra el evento de alarma de proceso. El OB puede asignarse al evento durante la configuración o en runtime.

Es posible asignar o cancelar la asignación de un OB a un evento habilitado al realizar la configuración. Para asignar un OB a un evento durante la configuración hay que utilizar la lista desplegable "Alarma de proceso:" (haga clic en la flecha hacia abajo de la derecha) y seleccionar un OB en la lista de OB de alarma de proceso disponibles. Seleccione el nombre de OB deseado en esta lista, o bien elija "<no conectado>" para cancelar la asignación.

También es posible asignar o deshacer la asignación de un evento de alarma de proceso habilitado en runtime. Las instrucciones de programa ATTACH o DETACH se utilizan en tiempo de ejecución (varias veces si se desea) para asignar un evento de alarma activado al OB adecuado o quitar su asignación. Si actualmente no hay ningún OB asignado (bien porque se ha

seleccionado "<no conectado>" en la configuración de dispositivos bien como resultado de ejecutar una instrucción DETACH), el evento de alarma de proceso activado se ignora.

Operación DETACH

La instrucción DETACH se utiliza para quitar la asignación de un evento concreto o de todos los eventos de un OB en particular. Si se ha especificado un EVENT, solo se quita la asignación de este evento al OB_NR especificado; si hay otros eventos asignados a este OB_NR, permanecerán asignados. Si no se especifica ningún EVENT, se deshace la asignación de todos los eventos asignados actualmente al OB_NR.

Códigos de condición

Tabla 9-99 Códigos de condición

RET_VAL (W#16#....)	ENO	Descripción
0000	1	No hay error
0001	1	No hay ninguna asignación que pueda deshacerse (solo DETACH)
8090	0	El OB no existe
8091	0	Tipo de OB incorrecto
8093	0	El evento no existe

9.5.2 Alarmas cíclicas

9.5.2.1 SET_CINT (ajustar parámetros de alarma cíclica)

Tabla 9-100 SET_CINT (Ajustar parámetros de alarma cíclica)

KOP / FUP	SCL	Descripción

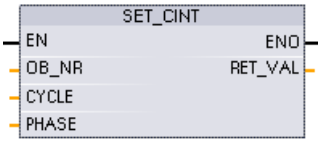
	`ret_val := SET_CINT(`
	`  ob_nr:=_int_in_,`
	`  cycle:=_udint_in_,`
	`  phase:=_udint_in_);`
		Ajustar el OB de alarma cíclica específico al comienzo de la ejecución cíclica que interrumpe el ciclo del programa.

Tabla 9-101 Tipos de datos para los parámetros

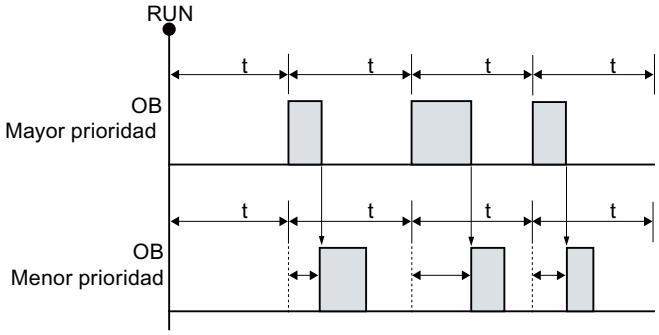
Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_CYCLIC	Número de OB (acepta el nombre simbólico)
CYCLE	IN	UDInt	Intervalo de tiempo en microsegundos
PHASE	IN	UDInt	Desfase, en microsegundos
RET_VAL	OUT	Int	Código de condición de ejecución

Ejemplos: parámetro de tiempo

- Si el tiempo CYCLE = 100 us, el OB de alarma referenciado por OB_NR interrumpirá el programa cíclico cada 100 us. El OB de alarma se ejecuta y a continuación devuelve el control de ejecución al ciclo del programa en el punto de la interrupción.
- Si el tiempo CYCLE = 0, el evento de alarma se desactiva y el OB de alarma no se ejecuta.
- El tiempo PHASE (desfase) es un retardo específico que se produce antes de que comience el intervalo de tiempo CYCLE. Se puede utilizar el desfase para control el tiempo de ejecución de OBs de menor prioridad.

Si se llaman OBs de prioridad menor y mayor en el mismo intervalo de tiempo, el OB de menor prioridad solo se llama una vez que ha finalizado el procesamiento del OB de mayor prioridad. El tiempo de inicio de la ejecución para el OB de menor prioridad puede cambiar en función del tiempo de procesamiento de OBs de mayor prioridad.

Llamada de OB sin desfase



Para reiniciar la ejecución de un OB de menor prioridad en un ciclo de tiempo fijo, el tiempo de desfase debe ser mayor que el tiempo de procesamiento de OBs de mayor prioridad.

Llamada de OB con desfase

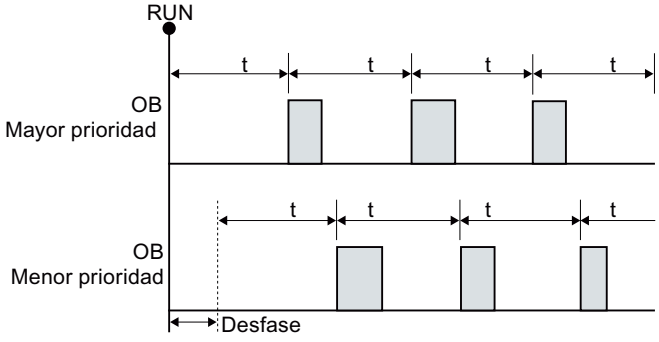


Tabla 9-102 Códigos de condición

RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	El OB no existe o es del tipo incorrecto
8091	Tiempo de ciclo no válido

RET_VAL (W#16#....)	Descripción
8092	Tiempo de desfase no válido
80B2	El OB no tiene ningún evento asignado

9.5.2.2 QRY_CINT (consultar parámetros de alarma cíclica)

Tabla 9-103 **QRY_CINT (consultar parámetros de alarma cíclica)**

KOP / FUP	SCL	Descripción

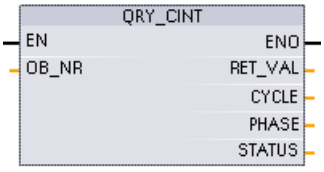
	`ret_val := QRY_CINT(`
	`  ob_nr:=_int_in_,`
	`  cycle=>_udint_out_,`
	`  phase=>_udint_out__,`
	`  status=>_word_out_);`
		Obtener los parámetros y el estado de ejecución de un OB de alarma cíclica. Los valores que se devuelven existían en el momento de ejecutar QRY_CINT.

Tabla 9-104 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_CYCLIC	Número de OB (acepta el nombre simbólico como OB_MyOBName)
RET_VAL	OUT	Int	Código de condición de ejecución
CYCLE	OUT	UDInt	Intervalo de tiempo en microsegundos
PHASE	OUT	UDInt	Desfase, en microsegundos
STATUS	OUT	Word	Código de estado de alarma cíclica: • Bits 0 a 4, véase la tabla STATUS siguiente • Otros bits, siempre 0

Tabla 9-105 Parámetro STATUS

Bit	Valor	Descripción
0	0	Durante RUN de la CPU
	1	Durante el arranque
1	0	La alarma está habilitada.
	1	Alarma deshabilitada vía la instrucción DIS_IRT.
2	0	La alarma no está activa o ha transcurrido.
	1	La alarma está activa.
4	0	El OB identificado por OB_NR no existe.
	1	El OB identificado por OB_NR existe.
Otros bits		Siempre 0

Si se produce un error, RET_VAL muestra el código de error apropiado y el parámetro STATUS es 0.

Tabla 9-106 Parámetro RET_VAL

RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	El OB no existe o es del tipo incorrecto.
80B2	El OB no tiene ningún evento asignado.

9.5.3 Alarmas horarias:

⚠ ADVERTENCIA

Riesgo de ataques a las redes mediante la sincronización del Network Time Protocol (NTP)

Si un atacante puede acceder a las redes por sincronización del Network Time Protocol (NTP), posiblemente alterará el control del proceso cambiando la hora del sistema de la CPU. Las alteraciones en el control del proceso pueden causar la muerte, heridas graves o daños materiales.

La función de cliente NTP de la CPU S7-1200 está desactivada por defecto y, si se activa, solo permite a las direcciones IP configuradas actuar como servidor NTP. La CPU desactiva esta función por defecto, y hay que configurar esta función para permitir correcciones de la hora del sistema de la CPU por control remoto.

La CPU S7-1200 soporta alarmas horarias e instrucciones de reloj que dependen de la precisión de la hora del sistema de la CPU. Si se configura NTP y se acepta la sincronización horaria desde un servidor, hay que asegurarse de que el servidor es una fuente fiable. Si no se comprueba, se puede producir una brecha de seguridad que permita a un usuario desconocido tomar un control limitado del proceso cambiando la hora del sistema de la CPU.

Para más información sobre seguridad y recomendaciones, consulte nuestras "Operational Guidelines for Industrial Security" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) en la página del Siemens Industry Online Support.

9.5.3.1 SET_TINTL (Ajustar alarma horaria)

Tabla 9-107 SET_TINTL (Ajustar alarma horaria)

KOP / FUP	SCL	Descripción

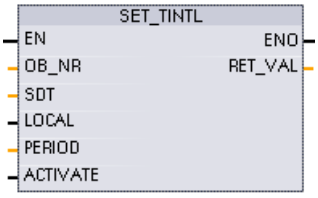
	`ret_val := SET_TINTL(`
	`OB_NR:=_int_in_,`
	`SDT:=_dtl_in_,`
	`LOCAL:=_bool_in_`
	`PERIOD:=_word_in_`
	`ACTIVATE:=_bool_in_);`
		Ajuste una alarma de fecha y hora. El OB de alarma de programa se puede ajustar para una ejecución o para una ejecución recurrente con un periodo de tiempo asignado.

Tabla 9-108 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_TOD (INT)	Número de OB (acepta el nombre simbólico)
SDT	IN	DTL	Fecha y hora de inicio: Los segundos y milisegundos se ignoran y se pueden poner a 0.
LOCAL	IN	Bool	0 = Usar hora del sistema 1 = Usar hora local (si la CPU se configura para hora local, en caso contrario use la hora del sistema)
PERIOD	IN	Word	El periodo de la fecha y hora de inicio para el evento de alarma recurrente. • W#16#0000 = Único • W#16#0201 = Cada minuto • W#16#0401 = Cada hora • W#16#1001 = Diario • W#16#1201 = Semanal • W#16#1401 = Mensual • W#16#1801 = Anual • W#16#2001 = Final de mes
ACTIVATE	IN	Bool	0 = Debe ejecutarse ACT_TINT para activar el evento de alarma. 1 = Se activa el evento de alarma.
RET_VAL	OUT	Int	Código de condición de ejecución

El programa puede utilizar SET_TINTL para ajustar un evento de alarma horaria que ejecutará el OB de alarma asignado. La fecha y hora de inicio se ajustan en el parámetro SDT y el período de las alarmas repetitivas (p. ej. cada día o cada semana) se ajusta en el parámetro PERIOD. Si se ajusta un período de repetición mensual, la fecha de inicio debe ser un día entre el 1 y el 28. Los días del 29 al 31 no deben utilizarse porque no existen en febrero. Si desea que haya un evento de alarma al final de cada mes, use el final de mes para el parámetro PERIOD.

El valor de día de la semana en los datos DTL del parámetro SDT se pasa por alto. Para ajustar la fecha y hora actual de una CPU, utilice la función "Ajustar la hora" en la vista "Online y

diagnóstico" de una CPU online. Debe ajustar el mes, el día del mes y el año. STEP 7 calcula el periodo de alarma en función del reloj de fecha y hora de la CPU.

Nota

La primera hora del día no existe al cambiar del horario de verano al de invierno (horario de verano). Use la hora de inicio dentro de la segunda hora o bien use una alarma de retardo adicional durante la primera hora.

Tabla 9-109 Código de condición

RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	Parámetro OB_NR no válido
8091	Parámetro de hora de inicio de SDT no válido: (por ejemplo, una hora de inicio dentro de la hora omitida al inicio del horario de verano)
8092	Parámetro PERIOD no válido
80A1	La hora de inicio corresponde a una hora pasada. (Este código de error se produce con PERIOD = W #16#0000.)

9.5.3.2 CAN_TINT (Cancelar alarma horaria)

Tabla 9-110 CAN_TINT (Cancelar alarma de fecha y hora)

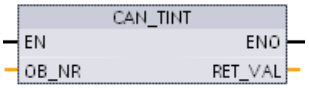
KOP / FUP	SCL	Descripción
	`ret_val:=CAN_TINT(_int_in);`	Cancela el evento de alarma de fecha y hora de inicio del OB de alarma especificado.

Tabla 9-111 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_TOD (INT)	Número de OB (acepta el nombre simbólico)
RET_VAL	OUT	Int	Código de condición de ejecución

Tabla 9-112 Códigos de condición

RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	Parámetro OB_NR no válido
80A0	No hay fecha/hora de inicio ajustada para el OB de alarma

9.5.3.3 ACT_TINT (Activar alarma horaria)

Tabla 9-113 ACT_TINT (Activar una alarma de fecha y hora)

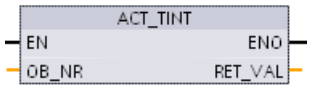
KOP / FUP	SCL	Descripción
	`ret_val:=ACT_TINT(_int_in_);`	Activa el evento de alarma de fecha y hora de inicio de un OB de alarma especificado.

Tabla 9-114 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_TOD (INT)	Número de OB (acepta el nombre simbólico)
RET_VAL	OUT	Int	Código de condición de ejecución

Tabla 9-115 Códigos de condición

RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	Parámetro OB_NR no válido
80A0	No se ha ajustado la fecha y hora de inicio para el OB de alarma horaria pertinente.
80A1	La hora activa es una hora que ya ha pasado. El error solo se produce cuando el OB de alarma se ajusta para que se ejecute una sola vez.

9.5.3.4 QRY_TINT (Consultar estado de alarma horaria)

Tabla 9-116 QRY_TINT (Consultar una alarma de fecha y hora)

KOP / FUP	SCL	Descripción

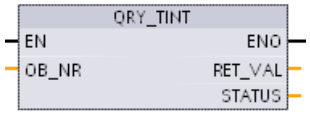
	`ret_val:=QRY_TINT(`
	`  OB_NR:=_int_in_,`
	`  STATUS=>_word_out_);`
		Consulta el estado del evento de alarma de fecha y hora de un OB de alarma especificado.

Tabla 9-117 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_TOD (INT)	Número de OB (acepta el nombre simbólico) del OB de alarma que se consulta
RET_VAL	OUT	Int	Código de condición de ejecución
STATUS	OUT	Word	Estado del OB de alarma especificado

Tabla 9-118 Parámetro STATUS

Bit	Valor	Descripción
0	0	En RUN
	1	En el arranque
1	0	La alarma está habilitada.
	1	La alarma está deshabilitada.
2	0	La alarma no está activa o ha transcurrido.
	1	La alarma está activa.
4	0	El parámetro OB_NR no existe.
	1	Existe un OB con el parámetro OB_NR asignado.
6	1	La alarma de fecha y hora usa la hora local.
	0	La alarma de fecha y hora usa la hora del sistema.
Otros		Siempre 0

Tabla 9-119 Código de condición

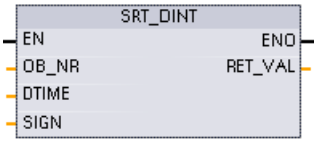
RET_VAL (W#16#....)	Descripción
0000	No hay error
8090	Parámetro OB_NR no válido

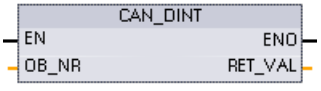
9.5.4 Alarmas de retardo

El procesamiento de las alarmas de retardo se puede iniciar y cancelar con las instrucciones SRT_DINT y CAN_DINT o se puede consultar el estado de la alarma con la instrucción QRY_DINT. Toda alarma de retardo es un evento único que ocurre al cabo del tiempo de retardo indicado. Si el evento de retardo se cancela antes de que transcurra el tiempo de retardo, no ocurrirá la alarma.

Tabla 9-120 Instrucciones SRT_DINT, CAN_DINT y QRY_DINT

KOP / FUP	SCL	Descripción

	`ret_val := SRT_DINT(`
	`ob_nr:=_int_in_,`
	`dtime:=_time_in_,`
	`sign:=_word_in_);`
		SRT_DINT inicia una alarma de retardo que ejecuta un OB una vez transcurrido el tiempo de retardo especificado en el parámetro DTIME.

	`ret_val := CAN_DINT(` `ob_nr:=_int_in_);`	CAN_DINT cancela una alarma de retardo ya iniciada. El OB de alarma de retardo no se ejecuta en este caso.

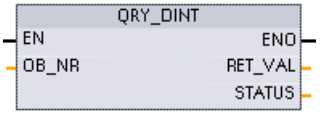
	`ret_val := QRY_DINT(`
	`ob_nr:=_int_in_,`
	`status=>_word_out_)`
	`;`
		QRY_DINT consulta el estado de la alarma de retardo especificada en el parámetro OB_NR.

Tabla 9-121 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
OB_NR	IN	OB_DELAY	Bloque de organización (OB) que debe ejecutarse tras un tiempo de retardo: Seleccione uno de los OBs de alarma de retardo creados utilizando la función "Agregar nuevo bloque" del árbol del proyecto. Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda para ver los OBs disponibles.
DTIME [1]	IN	Time	Tiempo de retardo (1 hasta 60000 ms)
SIGN [1]	IN	Word	No se utiliza en S7-1200: Se acepta cualquier valor. Debe asignarse un valor para prevenir errores.
RET_VAL	OUT	Int	Código de condición de ejecución
STATUS	OUT	Word	Instrucción QRY_DINT: Estado del OB de alarma de retardo especificado, véase la tabla siguiente

[1] Solo para SRT_DINT

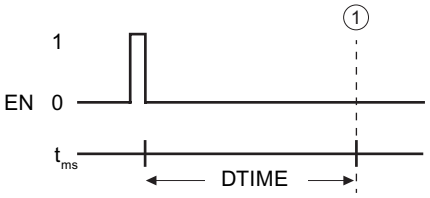
Funcionamiento

Si EN=1, la instrucción SRT_DINT inicia el temporizador de retardo interno (DTIME). Cuando termina el retardo, la CPU genera un alarma de programa que dispara la ejecución del OB de alarma de retardo asociado. Es posible cancelar una alarma de retardo en curso antes de que se produzca el retardo especificado ejecutando la instrucción CAN_DINT. No puede haber más de cuatro eventos de alarma de retardo activos.

Nota

SRT_DINT pone en marcha el temporizador de retardo cada ciclo cuando EN=1. Confirme EN=1 como un disparo en lugar de ajustar simplemente EN=1 para que inicie el retardo.

Cronograma de la instrucción SRT_DINT:



① Se ejecuta la alarma de retardo

Agregar OBs de alarma de retardo al proyecto

Los OB de alarma de retardo son los únicos que pueden asignarse a las instrucciones SRT_DINT y CAN_DINT. Un proyecto nuevo no contiene aún OBs de alarma de retardo. Es preciso agregar OB de alarma de retardo al proyecto. Para crear un OB de alarma de retardo, proceda del siguiente modo:

1. Haga doble clic en "Agregar nuevo bloque" en la rama "Bloques de programa" del árbol del proyecto, seleccione "Bloque de organización (OB)" y elija "Time delay interrupt" (Alarma de retardo).
2. Es posible cambiar el nombre del OB, así como seleccionar el lenguaje de programación y el número de bloque. Cambie al modo de numeración manual si desea asignar un número de bloque diferente del propuesto automáticamente.
3. Edite el OB de alarma de retardo y cree la reacción programada que debe ejecutarse cuando ocurra el evento de timeout de retardo. Puede llamar otros bloques lógicos FC y FB desde el OB de alarma de retardo. La profundidad de anidamiento máxima es de cuatro para programas de seguridad. Para el resto de programas, la profundidad de anidamiento máxima es de seis.
4. Los nuevos nombres de los OB de alarma de retardo estarán disponibles al editar el parámetro OB_NR de las instrucciones SRT_DINT y CAN_DINT.

Parámetro STATUS de QRY_DINT

Tabla 9-122 Si hay un error (REL_VAL <> 0), entonces STATUS = 0.

Bit	Valor	Descripción
0	0	En RUN
	1	En el arranque
1	0	La alarma está habilitada.
	1	La alarma está deshabilitada.
2	0	La alarma no está activa o ha transcurrido.
	1	La alarma está activa.
4	0	Un OB con un número de OB indicado en OB_NR no existe.
	1	Un OB con un número de OB indicado en OB_NR existe.
Otros bits		Siempre 0

Códigos de condición

Tabla 9-123 Códigos de condición para SRT_DINT, CAN_DINT y QRY_DINT

RET_VAL (W#16#...)	Descripción
0000	No ha ocurrido ningún error
8090	Parámetro OB_NR incorrecto
8091	Parámetro DTIME incorrecto
80A0	La alarma de retardo no se ha iniciado.

9.5.5 Instrucciones DIS_AIRT y EN_AIRT (Retardar/habilitar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad)

Las instrucciones DIS_AIRT y EN_AIRT sirven para deshabilitar y habilitar el procesamiento de alarmas.

Tabla 9-124 Instrucciones DIS_AIRT y EN_AIRT

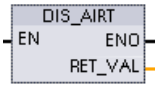
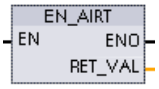
KOP / FUP	SCL	Descripción
	`DIS_AIRT();`	DIS_AIRT retarda el procesamiento de eventos de alarma nuevos. DIS_AIRT se puede ejecutar más de una vez en un OB.
	`EN_AIRT();`	EN_AIRT habilita el procesamiento de eventos de alarma deshabilitados anteriormente con la instrucción DIS_AIRT. Toda ejecución de DIS_AIRT debe ser cancelada por una ejecución de EN_AIRT. Las ejecuciones de EN_AIRT deben ocurrir en un mismo OB, o bien en una FC o FB llamado desde el mismo OB, antes de poder habilitar las alarmas nuevamente para este OB.

Tabla 9-125 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
RET_VAL	OUT	Int	Número de retardos = número de ejecuciones de DIS_AIRT en la cola de espera.

El sistema operativo cuenta las ejecuciones de DIS_AIRT. Cada ejecución permanece activa hasta que es cancelada específicamente por una instrucción EN_AIRT o hasta que se termina de procesar el OB actual. Ejemplo: Si se han deshabilitado alarmas cinco veces con cinco ejecuciones de DIS_AIRT, puede cancelarlas con cinco ejecuciones de EN_AIRT antes de que las alarmas se rehabiliten.

Una vez que los eventos de alarma sean habilitadas nuevamente, se procesan las alarmas que ocurrieron cuando estaba activa la instrucción DIS_AIRT, o bien se procesan tan pronto como se haya ejecutado el OB actual.

El parámetro RET_VAL indica el número de veces que se ha inhibido el procesamiento de alarmas. Este es el número de ejecuciones de DIS_AIRT en la cola de espera. El procesamiento de alarmas solo se puede habilitar nuevamente cuando el parámetro RET_VAL = 0.

9.6 Alarmas

9.6.1 Gen_UsrMsg (crear avisos de diagnóstico de usuario)

Tabla 9-126 Instrucción Gen_UsrMsg

KOP / FUP	SCL	Descripción

	`ret_val :=Gen_UsrMsg(`
	`  Mode:=_uint_in_,`
	`  TextID:=_uint_in_,`
	`  TextListID:=_uint_in_,`
	`  AssocValues:=_struct_inout_);`
		La instrucción "Gen_UsrMsg" se utiliza para crear un aviso de diagnóstico de usuario que puede ser tanto entrante como saliente. Mediante los avisos de diagnóstico de usuario es posible escribir una entrada de usuario en el búfer de diagnóstico y enviar un aviso en correspondencia.
		La entrada en el búfer de diagnóstico se crea de forma síncrona. La transmisión del aviso es asíncrona.
		Si se produce un error mientras se ejecuta una instrucción, se emite en el parámetro RET_VAL.

Contenido del aviso

Una lista de textos define el contenido del aviso:

- Defina la lista de textos que desee utilizar con el parámetro TextListID. Para ello, abra el cuado de diálogo "Listas de textos" en el árbol del proyecto. Visualice la columna "ID" en el cuadro de diálogo "Listas de textos". Aplique la ID en el parámetro TextListID.
- Utilice el parámetro TextID para seleccionar la entrada de la lista de textos que debe escribirse en el búfer de diagnóstico. Para ello, seleccione una entrada en el cuadro de diálogo "Listas de textos" aplicando un número de las columnas "Rango desde / Rango hasta" en el parámetro TextID. Hay que utilizar el mismo número de las columnas "Rango desde" y "Rango hasta" para la entrada en la lista de textos.

Encontrará información detallada sobre las listas de textos en el sistema de información de STEP 7.

Definir valores asociados

La entrada en la lista de textos define los nuevos valores asociados que deben agregarse al aviso:

- Incluya la información siguiente en la entrada de la lista de textos para definir valores asociados:
  @<N.° del valor asociado><tipo de elemento><especificación del formato>@
- Utilice el tipo de datos de sistema AssocValues para especificar el valor asociado que debe agregarse al generar el aviso.

Encontrará información detallada sobre la estructura de los valores asociados en el sistema de información de STEP 7.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "Gen_UsrMsg":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
Mode	Input	UInt	I, Q, M, D, L o constante	Parámetros para seleccionar el estado del aviso: • 1: aviso entrante • 2: aviso saliente
TextID	Input	UInt	I, Q, M, D, L o constante	ID de la entrada en la lista de textos que debe utilizarse para el texto del aviso.
TextListID	Input	UInt	I, Q, M, D, L o constante	ID de la lista de textos que contiene la entrada en la lista de textos.
Ret_Val	Return	Int	I, Q, M, D, L	Código de error de la instrucción.
AssocValues	InOut	VARIANT	D, L	Puntero al tipo de datos de sistema AssocValues que permite definir los valores asociados.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Parámetro AssocValues

Utilice el tipo de datos de sistema AssocValues para definir los valores asociados que deben enviarse. Se admiten como máximo ocho valores asociados. Introduzca el tipo de datos "AssocValues" como bloque de datos para crear la estructura.

Los valores asociados se seleccionan introduciendo sus números para los parámetros Value[x]. Tenga en cuenta lo siguiente:

- La instrucción Gen_UsrMsg considera los valores de TextID y TextListID valores asociados que deben enviarse. Como resultado, "1" y "2" se preasignan como números para el direccionamiento de valores asociados. No utilice los números "1" ni "2" para direccionar valores asociados.
- Direccione el valor asociado del parámetro Value [1] como número "3", el del parámetro Value [2] como número "4" y así sucesivamente.

Byte	Parámetro	Tipo de datos	Valor inicial	Descripción	Número del valor asociado
0..1	Value[1]	UINT	0	Primer valor asociado del aviso.	3
2..3	Value[2]	UINT	0	Segundo valor asociado del aviso.	4
4..5	Value[3]	UINT	0	...	5

Byte	Parámetro	Tipo de datos	Valor inicial	Descripción	Número del valor asociado
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	Octavo valor asociado del aviso.	10

Parámetro RET_VAL

La tabla siguiente define los valores de salida del parámetro RET_VAL . Consulte también Códigos de error comunes para las instrucciones avanzadas (Página 531).

Código de error* (W#16#...)	Explicación
0000	Sin errores
8080	Valor no soportado en el parámetro MODE.
80C1	Escasez de recursos debido a un exceso de llamadas paralelas.
8528	El parámetro 5 (AssocValues) no está alineado a un byte.
853A	El parámetro 5 (AssocValues) hace referencia a un punto no válido.
* El código de error puede visualizarse en el editor de programas como valor entero o hexadecimal.	

9.7 Diagnóstico (PROFINET o PROFIBUS)

9.7.1 Instrucciones de diagnóstico

Las siguientes instrucciones de diagnóstico se pueden utilizar tanto con PROFINET como con PROFIBUS:

- Instrucción RD_SINFO (Página 417): Lee la información de arranque del OB actual
- Instrucción LED (Página 427): Lee el estado de los LED en una unidad periférica descentralizada.
- Instrucción Get_IM_Data (Página 428): Comprueba los datos de identificación y mantenimiento (I&M) del módulo o submódulo especificado.
- Instrucción Get_Name (Página 429): Lee el nombre de un dispositivo PROFINET IO, esclavo PROFIBUS o esclavo AS-i.
- Instrucción GetStationInfo (Página 435): Lee la dirección IP o MAC de un dispositivo PROFINET IO ubicado en el sistema IO local o de un dispositivo PROFINET IO ubicado en un sistema IO de nivel inferior (conectado usando módulos CP/CM).

- Instrucción DeviceStates (Página 445): Recupera los estados operativos de una unidad periférica descentralizada dentro de un subsistema de E/S.
- Instrucción ModuleStates (Página 450): Recupera los estados operativos de los módulos de una unidad periférica descentralizada.
- Instrucción GET_DIAG (Página 455): Lee la información de diagnóstico de un dispositivo especificado.

Nota

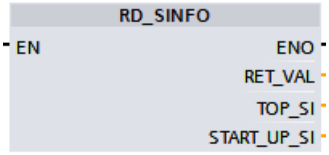
La instrucción GetStationInfo solo puede utilizarse con dispositivos PROFINET IO. No es posible emplear la instrucción con esclavos PROFIBUS DP.

9.7.2 RD_SINFO (leer información de arranque del OB actual)

Descripción

Tabla 9-127 Instrucción RD_SINFO

KOP / FUP	SCL	Descripción

	`ret_val := RD_SINFO(`
	`TOP_SI=>_variant_out_,`
	
	`START_UP_SI=>_variant_out_)`
	`;`
		La instrucción "RD_SINFO" se utiliza para leer la información de arranque de los OB siguientes:
		• El último OB llamado que todavía no ha terminado de ejecutarse
		• El último OB de arranque que ha iniciado la CPU
		No hay sello de tiempo en ninguno de los casos. Si la llamada es en el OB 100, OB 101 u OB 102, se devolverán dos avisos idénticos de información de arranque.

Parámetro

La tabla siguiente muestra los parámetros de la instrucción "RD_SINFO":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
RET_VAL	Return	INT	I, Q, M, D, L	Información de error
TOP_SI	Output	VARIANT	D, L	Información de arranque del OB actual
START_UP_SI	Output	VARIANT	D, L	Información de arranque del último OB de arranque que se inició

Encontrará más información detallada sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

SDT del parámetro TOP_SI

La tabla siguiente muestra los SDT posibles del parámetro TOP_SI:

Bloques de organización (OB)	Tipos de datos de sistema (SDT)	Números de tipos de datos de sistema
Cualquiera	SI_classic*	592*
	SI_none	593
ProgramCycleOB	SI_ProgramCycle	594
TimeOfDayOB	SI_TimeOfDay	595
TimeDelayOB	SI_Delay	596
CyclicOB	SI_Cyclic	597
ProcessEventOB	SI_HWInterrupt	598
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

* El SDT SI_classic no es válido para el S7-1200. La CPU S7-1200 devuelve un RET_VAL de #16#8081 si el parámetro TOP_SI es del tipo SI_classic.

SDT del parámetro START_UP_SI

La tabla siguiente muestra los SDT posibles del parámetro START_UP_SI:

Tipos de datos de sistema (SDT)	Números de tipos de datos de sistema
SI_classic*	592
SI_none	593
SI_Startup	613

* El SDT SI_classic no es válido para el S7-1200. La CPU S7-1200 devuelve un RET_VAL de #16#8083 si el parámetro START_UP_SI es del tipo SI_classic.

Estructuras

Las tablas siguientes definen los elementos de las diferentes estructuras:

Tabla 9-128 Estructura de SI_classic

Elemento de estructura	Tipo de datos	Descripción
EV_CLASS	BYTE	• Bits 0 a 3: ID de evento • Bits 4 a 7: clase de evento
EV_NUM	BYTE	Número de evento
PRIORITY	BYTE	Número de clase de prioridad (Significado de B#16#FE: OB no disponible o desactivado o no puede iniciarse en el estado operativo actual)
NUM	BYTE	Número del OB
TYP2_3	BYTE	ID de datos 2_3: identifica la información introducida en ZI2_3
TYP1	BYTE	ID de datos 1: identifica la información introducida en ZI1
ZI1	WORD	Información adicional 1
ZI2_3	DWORD	Información adicional 2_3

Tabla 9-129 Estructura de SI_none

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)

Tabla 9-130 Estructura de SI_ProgramCycle

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 1	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Initial_Call	BOOL	Para OB_Class = 1, 30, 52, 61, 65
Remanence	BOOL	Para OB_Class = 1

Tabla 9-131 Estructura de SI_TimeOfDay

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 10	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
CaughtUp	BOOL	Para OB_Class = 10
SecondTime	BOOL	Para OB_Class = 10

Tabla 9-132 Estructura de SI_Delay

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 20	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Sign	WORD	Para OB_Class = 20

Tabla 9-133 Estructura de SI_Cyclic

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 30	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Initial_Call	BOOL	Para OB_Class = 1, 30, 52, 61, 65
Event_Count	INT	Para OB_Class = 30, 51, 52, 61, 65, 91, 92

Tabla 9-134 Estructura de SI_HWInterrupt

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 40	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_IO	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
USI	WORD	Para OB_Class = 40
IChannel	USINT	Para OB_Class = 40
EventType	BYTE	Para OB_Class = 40

Tabla 9-135 Estructura de SI_Submodule

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_IO	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Slot	UINT	Para OB_Class = 55, 56, 57
Specifier	WORD	Para OB_Class = 55, 56, 57

Tabla 9-136 Estructura de SI_SynchCycle

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 61	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Initial_Call	BOOL	Para OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Para OB_Class = 61, 91, 92
PIP_Output	BOOL	Para OB_Class = 61, 91, 92
IO_System	USINT	Para OB_Class = 61, 91, 92
Event_Count	INT	Para OB_Class = 30, 51, 52, 61, 65, 91, 92
SyncCycleTime	LTIME	Tiempo de ciclo calculado

Tabla 9-137 Estructura de SI_IORedundancyError

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 70	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_ANY	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Para OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Para OB_Class = 70, 80, 83, 85, 86

Tabla 9-138 Estructura de SI_CPURedundancyError

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 72	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Switch_Over	BOOL	Para OB_Class = 72

Tabla 9-139 Estructura de SI_TimeError

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 80	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Fault_ID	BYTE	Para OB_Class = 70, 80, 83, 85, 86
Csg_OBnr	OB_ANY	Para OB_Class = 80
Csg_Prio	UINT	Para OB_Class = 80

Tabla 9-140 Estructura de SI_DiagnosticInterrupt

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 82	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
IO_State	WORD	Para OB_Class = 82
LADDR	HW_ANY	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Channel	UINT	Para OB_Class = 82
MultiError	BOOL	Para OB_Class = 82

Tabla 9-141 Estructura de SI_PlugPullModule

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 83	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_IO	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92

Elemento de estructura	Tipo de datos	Descripción
Event_Class	BYTE	Para OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Para OB_Class = 70, 80, 83, 85, 86

Tabla 9-142 Estructura de SI_AccessError

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 85	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_IO	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Para OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Para OB_Class = 70, 80, 83, 85, 86
IO_Addr	UINT	Para OB_Class = 85
IO_LEN	UINT	Para OB_Class = 85

Tabla 9-143 Estructura de SI_StationFailure

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 86	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LADDR	HW_IO	Para OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Para OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Para OB_Class = 70, 80, 83, 85, 86

Tabla 9-144 Estructura de SI_Servo

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 91	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Initial_Call	BOOL	Para OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Para OB_Class = 61, 91, 92
PIP_Output	BOOL	Para OB_Class = 61, 91, 92
IO_System	USINT	Para OB_Class = 61, 91, 92

Elemento de estructura	Tipo de datos	Descripción
Event_Count	INT	Para OB_Class = 30, 51, 52, 61, 65, 91, 92
Synchronous	BOOL	

Tabla 9-145 Estructura de SI_Ipo

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 92	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
Initial_Call	BOOL	Para OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Para OB_Class = 61, 91, 92
PIP_Output	BOOL	Para OB_Class = 61, 91, 92
IO_System	USINT	Para OB_Class = 61, 91, 92
Event_Count	INT	Para OB_Class = 30, 51, 52, 61, 65, 91, 92
Reduction	UINT	Para OB_Class = 92

Tabla 9-146 Estructura de SI_Startup

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT := 100	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
LostRetentive	BOOL	Para OB_Class = 100
LostRTC	BOOL	Para OB_Class = 100

Tabla 9-147 Estructura de SI_ProgIOAccessError

Elemento de estructura	Tipo de datos	Descripción
SI_Format	USINT	• 16#FF = no hay información • 16#FE = información de arranque optimizada
OB_Class	USINT	Clase de OB para "no hay información" o "información de arranque optimizada"
OB_Nr	UINT	Número del OB (1 ... 32767)
BlockNr	UINT	Para OB_Class = 121, 122
Reaction	USINT	Para OB_Class = 121, 122
Fault_ID	BYTE	Para OB_Class = 121, 122
BlockType	USINT	Para OB_Class = 121, 122
Area	USINT	Para OB_Class = 121, 122

Elemento de estructura	Tipo de datos	Descripción
DBNr	DB_ANY	Para OB_Class = 121, 122
Csg_OBNr	OB_ANY	Para OB_Class = 121, 122
Csg_Prio	USINT	Para OB_Class = 121, 122
Width	USINT	Para OB_Class = 121, 122

Nota

Si se ha creado con la propiedad de bloque "Estándar", los elementos especificados para la estructura de SI_classic tienen el mismo contenido que las variables temporales de un OB.

Sin embargo, hay que tener en cuenta que las variables temporales de los diferentes OB pueden tener nombres distintos y tipos de datos diferentes. También debe tenerse en cuenta que la interfaz de llamada de cada OB incluye información adicional relacionada con la fecha y hora de la petición del OB.

Los bits 4 a 7 del elemento de estructura EV_CLASS contienen la clase de evento. Aquí son posibles los valores siguientes:

- 1: Eventos de arranque de OB estándar
- 2: Eventos de arranque de OB de error síncrono
- 3: Eventos de arranque de OB de error asíncrono

El elemento de estructura PRIORITY proporciona la clase de prioridad del OB actual.

Aparte de estos dos elementos también es relevante NUM . NUM contiene el número del OB actual o del OB de arranque que se inició en último lugar.

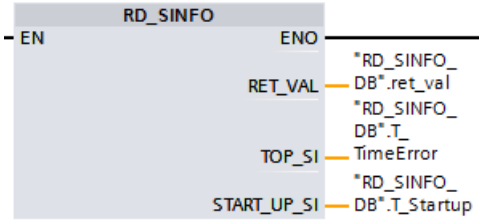
Parámetro RET_VAL

En la tabla siguiente se explican los valores del parámetro RET_VAL:

Código de error* (W#16#...)	Explicación
8081	La información de arranque del OB actual no se corresponde con el tipo de datos de sistema especificado
8083	La información de arranque del último OB de arranque iniciado no se corresponde con el tipo de datos de sistema especificado
* El código de error puede visualizarse en el editor de programas como valor entero o hexadecimal.	

Ejemplo

Un OB de error de tiempo (OB 80) es el último OB llamado que todavía no ha terminado de procesarse. Un OB de arranque (OB 100) es el último OB de arranque iniciado. La llamada de la instrucción para leer la información de arranque es la siguiente, siendo RD_SINFO_DB el bloque de datos que contiene variables de los SDT para tipos de OB:



La tabla siguiente muestra la asignación entre los elementos de estructura del parámetro TOP_SI de la instrucción "RD_SINFO" y las variables locales asignadas del OB 80.

Elemento de estructura TOP_SI	Tipo de datos	OB 80 - Variable local asociada	Tipo de datos
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

La tabla siguiente muestra la asignación entre los elementos de estructura del parámetro START_UP_SI de la instrucción "RD_SINFO" y las variables locales asignadas del OB 100.

Elemento de estructura START_UP_SI	Tipo de datos	OB 100 - Variable local	Tipo de datos
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 (leer estado del LED)

Tabla 9-148 Instrucción LED

KOP / FUP	SCL	Descripción

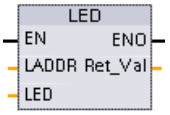
	`ret_val := LED(`
	`  laddr:=_word_in_,`
	`  LED:=_uint_in_);`
		Utilice la instrucción LED para leer el estado de los LED de la CPU (Página 1197). El estado de LED especificado se devuelve en la salida RET_VAL.

Tabla 9-149 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción		
LADDR	IN	HW_IO	Identificador de la CPU[1]		
LED	IN	UInt	Número de identificador LED		
			1	RUN/STOP	Color 1 = verde, color 2 = amarillo
			2	Error	Color 1 = rojo
			3	Mantenimiento	Color 1 = amarillo
RET_VAL	OUT	Int	Estado del LED		

[1] Para el identificador de la CPU conectada, seleccione Local~Common de la lista desplegable del parámetro.

Tabla 9-150 Estado de RET_VAL

RET_VAL (W#16#...)	Descripción	
Estado de LED 0 a 9	0	El DB no existe
	1	Off
	2	Color 1 encendido (permanente)
	3	Color 2 encendido (permanente)
	4	Color 1 parpadea a 2 Hz
	5	Color 2 parpadea a 2 Hz
	6	Color 1 & 2 parpadean de forma alterna a 2 Hz
	9	El estado del LED no está disponible
8091	El dispositivo identificado por LADDR no existe	
8092	El dispositivo identificado por LADDR no soporta LEDs	
8093	Identificador LED no definido	
80Bx	La CPU identificada por LADDR no soporta la instrucción LED	

9.7.4 Get_IM_Data (leer datos de identificación y mantenimiento)

La instrucción Get_IM_Data se utiliza para comprobar los datos de identificación y mantenimiento (I&M) del módulo o submódulo especificado.

Tabla 9-151 Instrucción Get_IM_Data

KOP / FUP	SCL	Descripción

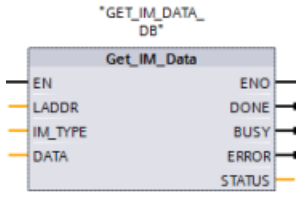
	`"GET_IM_DATA_DB"(LADDR:=16#0 ,`
	`IM_TYPE:=0,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DATA:=_variant_inout_);`
		Utilice la instrucción Get_IM_Data para comprobar los datos de identificación y mantenimiento (I&M) del módulo o submódulo especificado.

Tabla 9-152 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
LADDR	Input	HW_IO	Identificador del módulo
IM_TYPE	Input	UInt	Número de datos de identificación y mantenimiento (I&M): • 0: I&M0 (ref. (MLFB), número de serie, versión y otra información) • 1: I&M1 (identificadores) • 2: I&M2 (fecha de instalación) • 3: I&M3 (descripción) • 4: I&M4 (firma)
RET_VAL	Output	Int	Estado (código de condición)
DATA	InOut	Variante	Datos I&M (STRING o Array of BYTE); uso recomendado del SDT "IM0_Data" para IM_TYPE = 0.

Los datos de identificación y mantenimiento (I&M) pueden ayudarle a comprobar la configuración del sistema, detectar cambios en el hardware o visualizar datos de mantenimiento. Los datos de identificación del módulo (datos I) son de solo lectura. Los datos de mantenimiento del módulo (datos M) dependen de la información del sistema, como la fecha de instalación. Los datos M se crean durante la planificación del mantenimiento y se escriben en el módulo:

- Si el tipo de datos utilizado en el parámetro DATA es una cadena de caracteres, la longitud actual de la misma se ajusta en función de la longitud de los datos IM.
- Si el tipo de datos utilizado en el parámetro DATA es un array of Byte o un array of Char, los datos IM se copian como secuencia de bytes.
- Si el tipo de datos utilizado en el parámetro DATA es una estructura, los datos IM se copian como secuencia de bytes.
- Si el Array of Byte/Char que aparece en DATA es más largo que los datos IM solicitados, se agrega el valor 16#00.
- No se soportan otros tipos de datos, que devuelven el error 8093.

Tabla 9-153 Códigos de condición

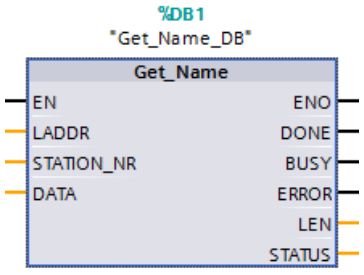
RET_VAL (W#16#...)	Descripción
0	No hay error
8091	LADDR no existe
8092	LADDR no direcciona un objeto HW que soporta datos IM
8093	El tipo de datos del parámetro DATA no se soporta
80B1	Instrucción DATA no soportada por la CPU para este LADDR
80B2	IM_TYPE no soportado por la CPU
8452	La información IM completa no cabe en la variable del parámetro DATA. Se devuelve un resultado parcial que abarca hasta la longitud de bytes de la variable.

9.7.5 Get_Name (leer el nombre de un dispositivo PROFINET IO)

La instrucción "Get_Name" lee el nombre de un dispositivo PROFINET IO, esclavo PROFIBUS o esclavo AS-i. El nombre se muestra en la vista de redes y en las propiedades del dispositivo IO.

Tabla 9-154 Instrucción Get_Name

KOP / FUP	SCL	Descripción

	`"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_);`
		Use la instrucción Get_Name para leer el nombre de un dispositivo PROFINET IO o esclavo PROFIBUS.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "Get_Name_DB" es el nombre del DB de instancia.

Seleccione el dispositivo PROFINET IO mediante el identificador de hardware del sistema de periferia descentralizada (en el parámetro LADDR) y el número del dispositivo PROFINET IO o la dirección PROFIBUS del esclavo PROFIBUS (en el parámetro STATION_NR).

Una vez se ha ejecutado la instrucción, el programa escribe el nombre del dispositivo IO en el área direccionada con el parámetro DATA.

El nombre que se lee depende del tipo de dispositivo IO:

- Esclavo DP o dispositivo IO: Nombre del módulo de cabecera
- Esclavo I o I-device: Nombre del módulo de interfaz
- Panel HMI: Nombre de la interfaz
- Estación PC: Nombre del módulo de interfaz
- Dispositivos GSD: Muestra el nombre del punto de acceso del dispositivo (DAP) (nombre de la interfaz o módulo de cabecera)

La instrucción escribe la longitud del nombre en el parámetroLEN. Si el nombre es más largo que el área especificada en el parámetro DATA, el programa solo escribe la sección que se corresponde con la longitud máxima del área direccionada.

La longitud máxima de un nombre es de 128 caracteres.

Nota

Nombre de la lectura de la CPU (versión 1.1)

Si se asigna "0" a cada uno de los parámetros LADDR y STATION_NR,, la instrucción escribe el nombre de la CPU.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción Get_Name:

Parámetro	Declaración	Tipo de datos	Descripción
LADDR	IN	HW_IOSYSTEM	Identificador de hardware (HW-IoSystem) del sistema de periferia descentralizada. El número se toma de las constantes del sistema o de las propiedades del sistema IO.
STATION_NR	IN	UInt	• Dispositivo PROFINET IO: El número de dispositivo se aplica en la vista de redes de las propiedades del dispositivo IO en "Direcciones Ethernet". • Esclavo PROFIBUS: La dirección PROFIBUS se aplica en la vista de redes de las propiedades del esclavo PROFIBUS en "Dirección PROFIBUS".
DATA	IN_OUT	Variant	Puntero al área en la que se ha escrito el nombre.
DONE	OUT	Bool	La instrucción se ejecuta correctamente. El nombre del módulo se transfiere al área en el parámetro DATA.
BUSY	OUT	Bool	Parámetro de estado: • 0: Ejecución de la instrucción terminada. • 1: Ejecución de la instrucción no terminada aún.
ERROR	OUT	Bool	Parámetro de estado: • 0: Sin errores • 1: Se ha producido un error al ejecutar la instrucción. El parámetro STATUS contiene información detallada.
LEN	OUT	DInt	Longitud del nombre del dispositivo IO (número de caracteres).
STATUS	OUT	Word	Parámetro de estado: El parámetro solo está activado mientras dura una llamada. Para mostrar el estado debe copiar STATUS en un área de datos libre.

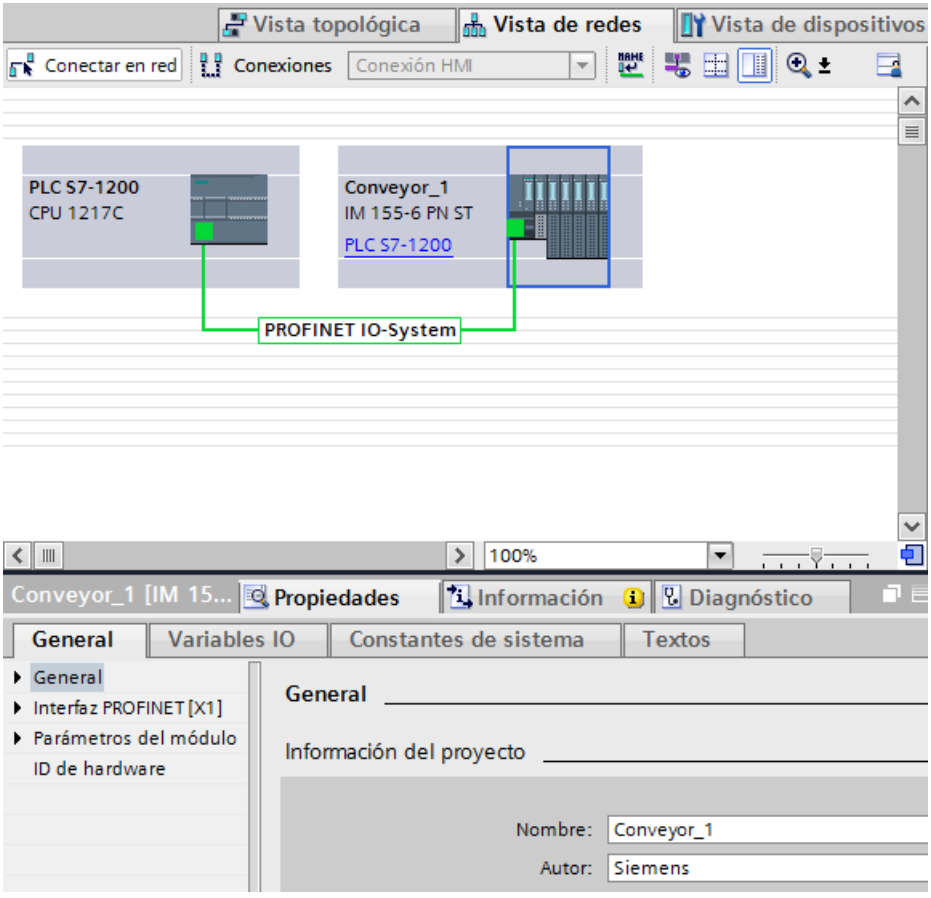
Parámetro STATUS

Código de error* (W#16#...)	Explicación
0	Sin errores
7000	Ninguna tarea en curso
7001	Primera llamada de la instrucción asíncrona Get_Name. Ejecución de la instrucción no terminada aún (BUSY = 1, DONE = 0).
7002	Llamada adicional de la instrucción asíncrona Get_Name. Ejecución de la instrucción no terminada aún (BUSY = 1, DONE = 0).
8090	El identificador de hardware especificado en el parámetroLADDR no existe en el proyecto.
8092	El valor del parámetro LADDR no direcciona un sistema PROFINET IO.
8093	La instrucción no soporta el tipo de datos del parámetro DATA.
8095	El número de dispositivo (parámetro STATION_NR) no existe en el sistema PROFINET IO seleccionado o no direcciona un dispositivo IO.
80B1	La CPU usada no soporta la instrucción.
80C3	Error de recurso temporal: La CPU procesa actualmente el número máximo posible de llamadas de bloque simultáneas. Get_Name no puede ejecutarse hasta que se haya finalizado como mínimo una de las llamadas de bloque.
8852	El área especificada en el parámetro DATA es demasiado corta para el nombre completo del dispositivo IO. El nombre puede escribirse hasta su longitud máxima posible. Para leer el nombre completo use un área de datos más larga en el parámetro DATA. El área debe tener como mínimo tantos caracteres como los que hay en el parámetro LEN.
* Los códigos de error en el editor de programas se pueden mostrar como valores enteros o hexadecimales.	

Ejemplo

El ejemplo siguiente muestra cómo leer el nombre de estación de un dispositivo PROFINET IO ET 200SP:

1. Configurar un ET 200SP:
   - Cree un ET 200SP con el nombre de estación "Conveyor_1" en la vista de redes y asígnelo al mismo sistema PROFINET IO que la CPU.
   - Asigne la CPU como controlador IO para el ET 200SP.
   - Utilice el número de dispositivo predeterminado "1" ubicado en las propiedades, bajo "Direcciones Ethernet".



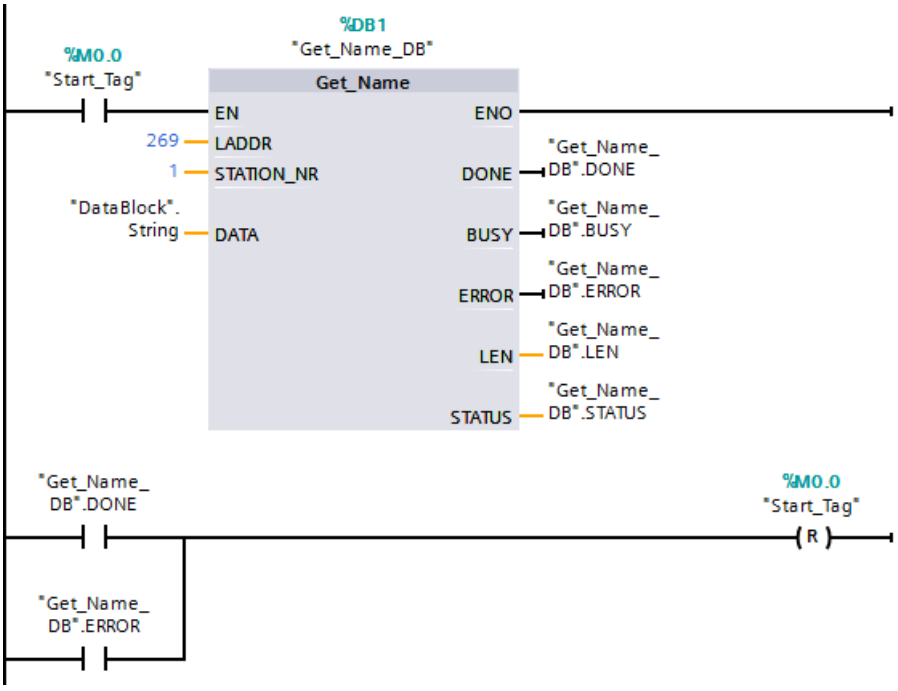
2. Asignar parámetros para la instrucción Get_Name:
   - Introduzca la ID de hardware del sistema IO en el parámetro LADDR. En este ejemplo, la ID de hardware es "269". Encontrará la ID de hardware en la ubicación siguiente: Variables PLC > Mostrar todas las variables> ficha Constantes de sistema > Sistema PROFINET IO local
   - Introduzca el número de dispositivo del ET 200SP en el parámetro STATION_NR. En este ejemplo, el número de dispositivo es "1".

– Conecte una variable con el tipo de datos STRING de un bloque de datos en el parámetro DATA.

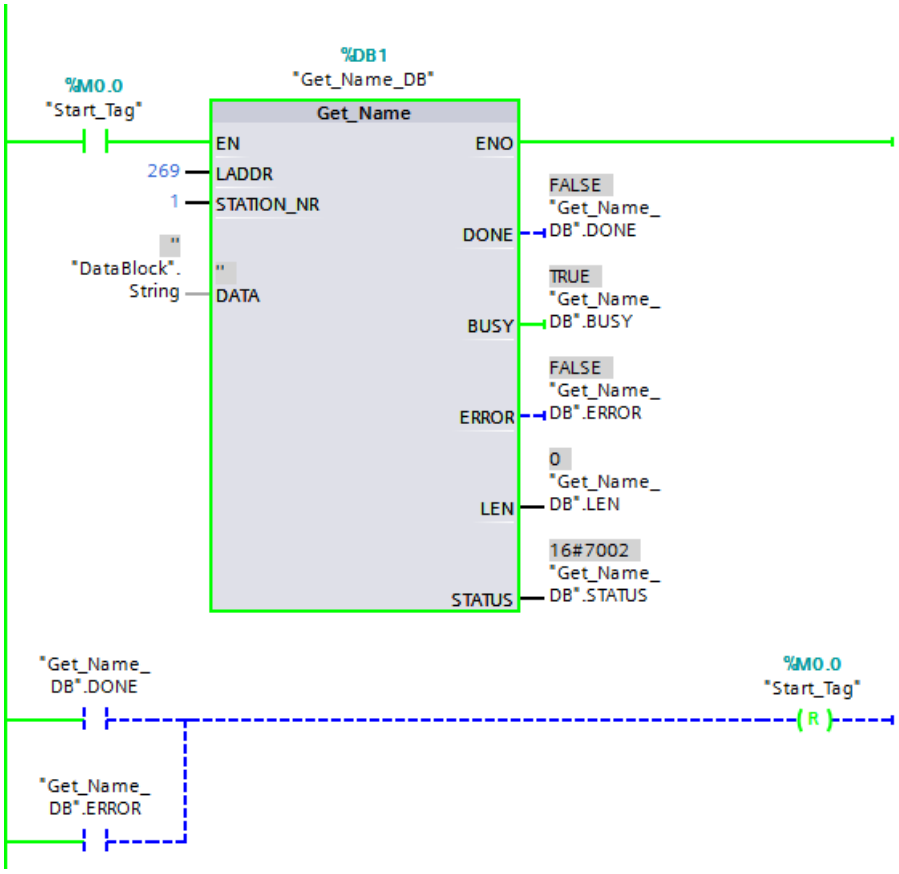
Nota

Cuando utilice la lista desplegable para configurar la variable en el parámetro DATA, seleccione el DB (en el ejemplo, "Bloque de datos") y la variable (en el ejemplo, "String[ ]"). Con el fin de leer todo el tipo de datos String, debe borrar los corchetes para que el resultado final sea: "Bloque de datos".String

– Definir variables PLC (área de memoria, marcas) para los parámetros de salida de la instrucción.

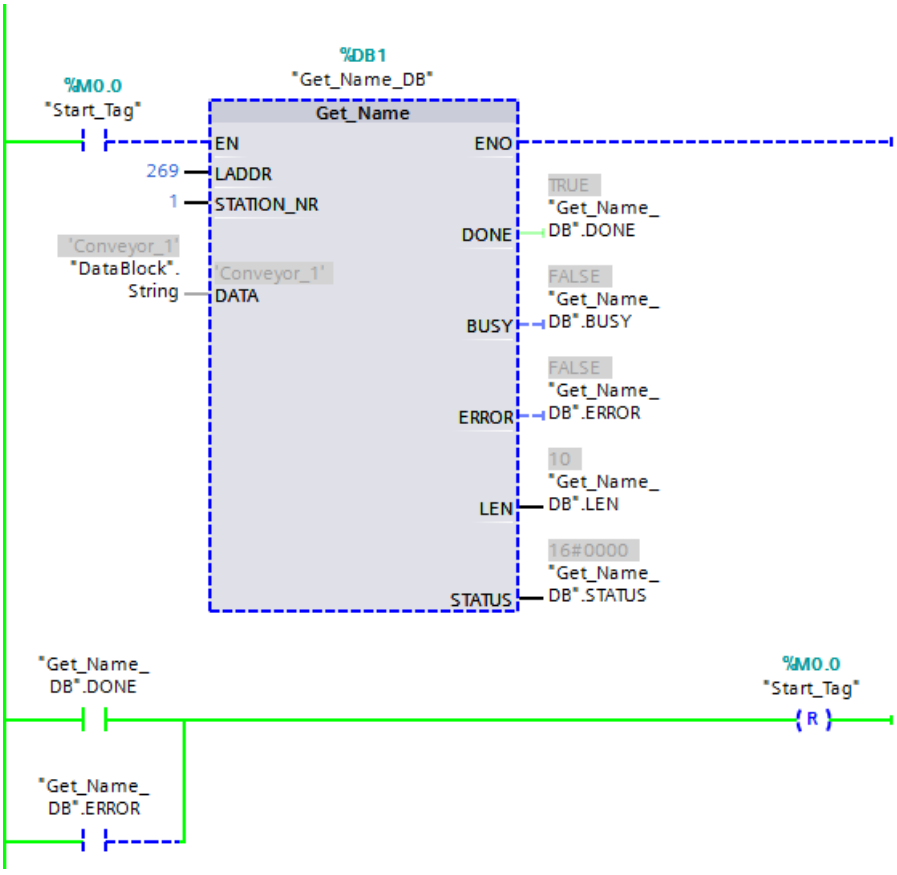


3. Ejecución de la instrucción Get_Name:
   - Mientras se ejecuta la instrucción, el parámetro de salida BUSY puede ponerse a "1" y el parámetro DONE se pone a "0".
   - La información del código de error se muestra en el parámetro de salida STATUS.



4. Completar la ejecución de la instrucción Get_Name:
   - Tras ejecutar la instrucción, el programa escribe "Conveyor_1", el nombre de la estación del ET 200SP, en el bloque de datos del parámetro DATA.

– El programa escribe "10", el número de caracteres que contiene el nombre de la estación, en el parámetro LEN.



9.7.6 GetStationInfo (leer la dirección IP o MAC de un dispositivo PROFINET IO)

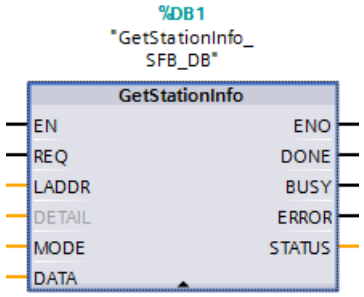
La instrucción "GetStationInfo" lee la dirección IP o MAC de un dispositivo PROFINET IO del sistema IO local o de un dispositivo PROFINET IO ubicado en un sistema IO de nivel inferior (conectado usando módulos CP/CM).

Nota

La instrucción GetStationInfo solo puede utilizarse con dispositivos PROFINET IO. No es posible emplear la instrucción con esclavos PROFIBUS DP.

Tabla 9-155 Instrucción GetStationInfo

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción GetStationInfo para leer la dirección IP o MAC de un dispositivo PROFINET IO. La instrucción también permite leer la dirección IP o MAC de un dispositivo IO ubicado en un sistema IO de nivel inferior (conectado usando módulos CP/CM).

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "GetStationInfo_SFB_DB" es el nombre del DB de instancia.

Direccione el dispositivo IO empleando el identificador de hardware de la estación en el parámetro LADDR. Encontrará la ID de hardware en la ubicación siguiente:
Variables PLC > Mostrar todas las variables> ficha Constantes de sistema. Busque "IODevice" en la columna de nombre y "Hw_Device" en la columna de tipo de datos.

Utilice el parámetro MODE para seleccionar la información que debe leerse.

En el parámetro DATA asigne el área de datos en la que la instrucción escribe los datos de dirección leídos. Para guardar la dirección IP, use la estructura "IF_CONF_v4". Para guardar la dirección MAC, use la estructura "IF_CONF_MAC".

Active la lectura de los datos de dirección mediante el parámetro de control REQ. Para ello, el dispositivo IO tiene que estar accesible.

La instrucción muestra el estado de ejecución de la tarea de lectura por medio de los parámetros de salida BUSY, DONE y ERROR y el parámetro de salida STATUS.

Nota

Direccione el dispositivo IO empleando solo el identificador de hardware de la estación.

La estación, el dispositivo IO y la interfaz PROFINET disponen de su propio identificador de hardware. Utilice únicamente el identificador de hardware de la estación para la instrucción GetStationInfo.

Si se direcciona una interfaz PROFINET mediante el parámetro LADDR, por ejemplo, los datos de dirección no se leen y la CPU genera un código de error "8092".

Para leer los datos de dirección de una interfaz PROFINET integrada o un módulo CM/CP en la configuración central, use la instrucción "RDREC".

Parámetros

La tabla siguiente muestra los parámetros de la instrucción GetStationInfo:

Parámetro	Declaración	Tipo de datos	Descripción
REQ	IN	Bool	Petición de parámetro de control Activa la lectura de la información con REQ = "1".
LADDR	IN	HW_DEVICE	El identificador de hardware de la estación del dispositivo IO El número se toma de las propiedades de la estación en la Vista de redes o de la ficha "Constantes de sistema" de la tabla de variables predeterminada.
DETAIL	IN	HW_SUBMODULE	El parámetro DETAIL no se utiliza. Deje el parámetro desconectado.
MODE	IN	UNIT	Selección de datos de dirección que deben leerse: • MODE = 1: Parámetro de dirección conforme a IPv4 (CPU S7-1200 a partir de la versión de firmware V4.2) • MODE = 2: Dirección MAC (CPU S7-1200 a partir de la versión de firmware V4.2)
DATA	IN_OUT	Variant	Puntero al área en la que el programa escribe los datos de dirección del dispositivo IO. Utilice la estructura "IF_CONF_v4" para MODE =1 y la estructura "IF_CONF_MAC" para MODE = 2.
DONE	OUT	Bool	El programa ha ejecutado la instrucción correctamente. El programa ha transferido los datos de dirección al parámetro DATA.
BUSY	OUT	Bool	Parámetro STATUS: • 0: Ejecución de la instrucción terminada. • 1: Ejecución de la instrucción no terminada aún.
ERROR	OUT	Bool	Parámetro STATUS: • 0: Sin errores. • 1: Se ha producido un error al ejecutar la instrucción. La información detallada se emite mediante el parámetro STATUS.
STATUS	OUT	Word	Parámetro STATUS: El parámetro solo está activado mientras dura una llamada. Para mostrar el estado debe copiar STATUS en un área de datos libre.

Parámetro DATA

- Utilice la estructura "IF_CONF_v4" en el parámetro DATA para guardar el parámetro de dirección conforme a IPv4:

Byte	Parámetro	Tipo de datos	Valor inicial	Descripción
0 ... 1	Id	UINT	30	ID de la estructura "IF_CONF_v4"
2 ... 3	Length	UNIT	18	Longitud de la lectura de datos en BYTE
4 ... 5	Mode	UNIT	0	Irrelevante para la instrucción "GetStationInfo" (dejar en "0")

Byte	Parámetro	Tipo de datos	Valor inicial	Descripción
6 ... 9	InterfaceAddress	ARRAY [1..4] of BYTE	-	Dirección IP del dispositivo IO en el formato IP_V4 (por ejemplo, 192.168.3.10): • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 10
10 ... 13	SubnetMask	ARRAY [1..4] of BYTE	-	Máscara de subred del dispositivo IO en el formato IP_V4 (por ejemplo, 255.255.255.0): • addr[1] = 255 • addr[2] = 255 • addr[3] = 255 • addr[4] = 0
14 ... 17	DefaultRouter	ARRAY [1..4] of BYTE	-	Dirección IP del router en el formato IP_V4 (por ejemplo, 192.168.3.1): • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 1

- Utilice la estructura "IF_CONF_MAC" en el parámetro DATA para guardar la dirección MAC:

Byte	Parámetro	Tipo de datos	Valor inicial	Descripción
0 ... 1	Id	UINT	3	ID de la estructura "IF_CONF_MAC"
2 ... 3	Length	UNIT	12	Longitud de la lectura de datos en BYTE
4 ... 5	Mode	UNIT	0	Irrelevante para la instrucción "GetStationInfo" (dejar en "0")
6 ... 11	MACAddress	ARRAY [1..6] of BYTE	-	Dirección MAC del dispositivo IO (por ejemplo, 08-00-06-12-34-56): • Mac[1] = 8 • Mac[2] = 0 • Mac[3] = 6 • Mac[4] = 12 • Mac[5] = 34 • Mac[6] = 56

Parámetro STATUS

Código de error* (W#16#...)	Explicación
0	Sin errores
7000	Ninguna tarea en curso
7001	Primera llamada de la instrucción asíncrona GetStationInfo. Ejecución de la instrucción no terminada aún (BUSY = 1, DONE = 0).

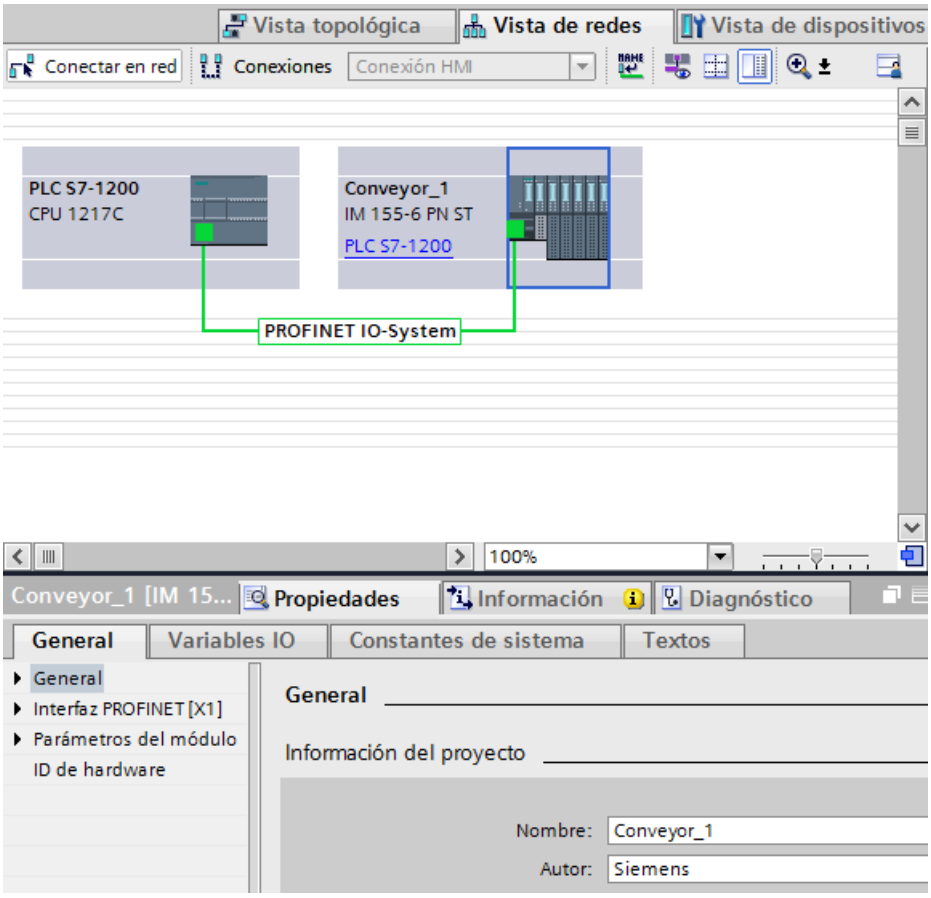
Código de error* (W#16#...)	**Explicación**
7002	Llamada adicional de la instrucción asíncrona GetStationInfo. Ejecución de la instrucción no terminada aún (BUSY = 1, DONE = 0).
8080	Valor no soportado en el parámetro MODE.
8090	El identificador de hardware especificado en el parámetroLADDR no está configurado.
8092	El parámetro LADDR no direcciona un dispositivo PROFINET IO.
8093	Tipo de datos no válido en el parámetro DATA.
80A0	La información solicitada no se ha leído.
80C0	El dispositivo IO direccionado no está accesible.
80C3	Se ha alcanzado el número máximo de llamadas simultáneas de la instrucción GetStationInfo (10 instancias).
* Los códigos de error en el editor de programas se muestran como valores enteros o hexadecimales.	

Ejemplo

En el ejemplo inferior se utiliza la instrucción GetStationInfo para leer los datos de la dirección IP de un dispositivo IO y escribir la información en un bloque de datos. Los datos de la dirección IP incluyen la dirección IP, la máscara de subred y los datos de dirección del router (si se utiliza).

El controlador IO ejecuta la instrucción GetStationInfo y la instrucción lee la información de la dirección IP de un dispositivo IO de nivel inferior (en este ejemplo, un ET200SP):

1. Configurar un ET 200SP:
   - Cree un ET 200SP con el nombre de estación "Conveyor_1" en la vista de redes y asígnelo al mismo sistema PROFINET IO que la CPU.
   - Asigne la CPU como controlador IO para el ET 200SP.



2. Asignar parámetros para la instrucción GetStationInfo :

– Cree cinco variables y una estructura con el tipo de datos IF_CONF_v4 en un bloque de datos global para guardar los datos de la dirección IP. Asigne un nombre cualquiera a la estructura. (En el ejemplo, el nombre de la estructura es "IP_Address".)

GetStationInfo_Global_DB

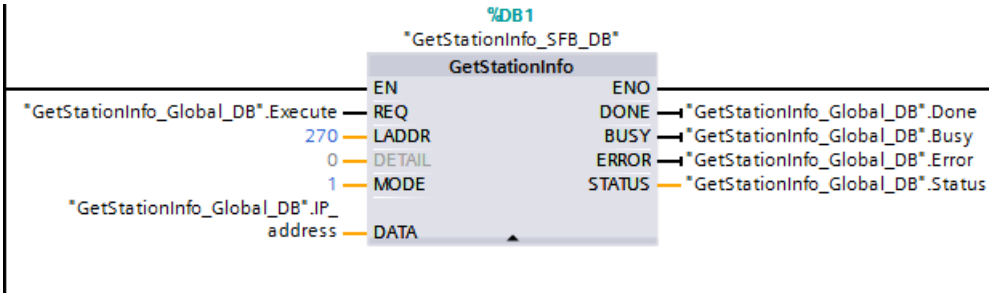
	Nombre	Tipo de datos	Valor de arranq...
1	Static		
2	Execute	Bool	false
3	IP_address	IF_CONF_v4	
4	Id	UInt	30
5	Length	UInt	18
6	Mode	UInt	0
7	InterfaceAddress	IP_V4	
8	ADDR	Array[1..4] of Byte	
9	ADDR[1]	Byte	16#0
10	ADDR[2]	Byte	16#0
11	ADDR[3]	Byte	16#0
12	ADDR[4]	Byte	16#0
13	SubnetMask	IP_V4	
14	DefaultRouter	IP_V4	
15	Done	Bool	false
16	Busy	Bool	false
17	Error	Bool	false
18	Status	Word	16#0

3. Asignar parámetros para la instrucción GetStationInfo:
   - Introduzca la ID de hardware del dispositivo IO en el parámetro LADDR. El identificador de hardware solo identifica el producto y, en este ejemplo, la ID de hardware es "270". Encontrará la ID de hardware en la ubicación siguiente: Variables PLC > Mostrar todas las variables> ficha Constantes de sistema
     Busque el dispositivo IO en la columna de nombre y "Hw_Device" en la columna de tipo de datos. El valor asociado es la ID de hardware que se ha introducido en el parámetro LADDR.
   - Seleccione "1" (leer parámetros de dirección conforme a IPv4) para el parámetro MODE.
   - Conecte la estructura IF_CONF_v4 en el parámetro DATA.

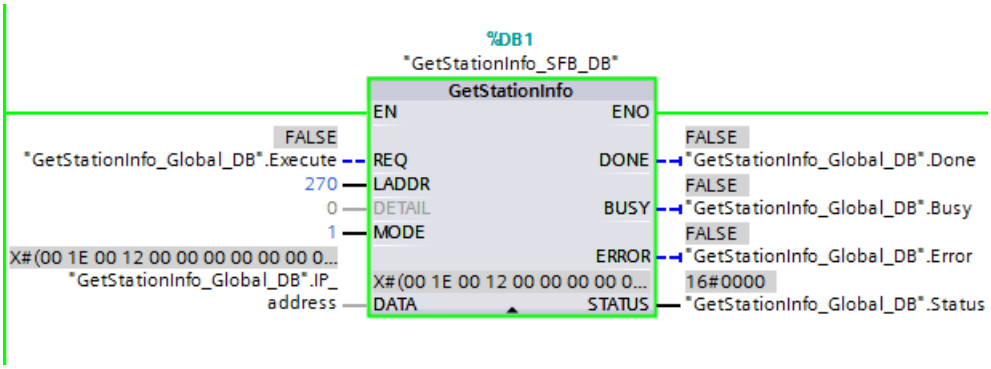
     Nota

     Cuando utilice la lista desplegable para configurar la variable en el parámetro DATA, seleccione el DB (en el ejemplo, "GetStationInfo_Global_DB") y la variable (en el ejemplo, "dirección IP"). Con el fin de leer todo el tipo de datos IF_CONF_v4, debe borrar el período que aparece después de "dirección IP" de forma que el resultado final sea: "GetStationInfo_Global_DB".Dirección IP

   - Definir variables PLC del DB global para los parámetros de salida de la instrucción.

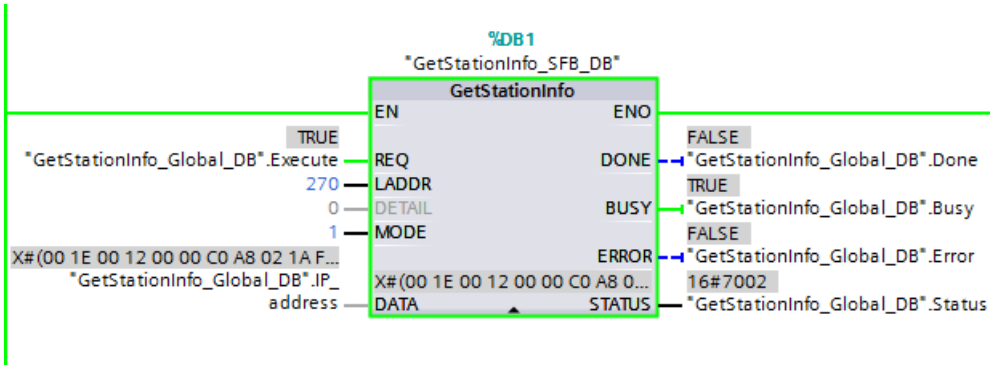


4. Ejecución de la instrucción GetStationInfo:
   – Cuando la entrada REQ = 1 (FALSE), la instrucción no muestra información de la dirección IP en el parámetro entrada/salida DATA ni información del código de error en el parámetro de salida STATUS.



5. Completar la ejecución de la instrucción GetStationInfo:

– Cuando la entrada REQ = 1 (TRUE), el programa ejecuta la instrucción y escribe la dirección IP en el bloque de datos. El programa escribe la dirección IP "C0 A8 02 1A" (valor decimal de "192.168.2.26") en el parámetro de entrada/salida DATA.



GetStationInfo_Global_DB

	Nombre	Tipo de datos	Valor de arranq...	Valor de observación
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 Instrucción DeviceStates

Puede utilizar la instrucción DeviceStates para devolver los estados de todos los dispositivos esclavos de E/S descentralizadas conectados a un maestro de E/S descentralizadas.

Tabla 9-156 Instrucción DeviceStates

KOP / FUP	SCL	Descripción

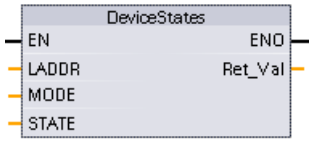
	`ret_val := DeviceStates(`
	`laddr:=_word_in_,`
	`mode:=_uint_in_,`
	`state:=_variant_inout_);`
		DeviceStates recupera los estados operativos del dispositivo de E/S de un subsistema de E/S. Tras la ejecución, el parámetro STATE contiene el estado de error de cada dispositivo de E/S en forma de lista de bits (para LADDR y MODE asignados). Esta información se corresponde con el estado del dispositivo que se indica en la vista de diagnóstico de STEP 7.
		La entrada LADDR de DeviceStates utiliza el identificador de hardware de una interfaz de E/S descentralizadas. En TIA Portal, los identificadores de hardware de un PLC pueden determinarse buscando los tipos de datos "HW_IOSYSTEM" en la pestaña de constantes del sistema en la tabla de variables PLC.

Tabla 9-157 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
LADDR	IN	HW_IOSYSTEM	Dirección lógica: (Identificador del sistema I/O)
MODE	IN	UInt	Admite cinco modos de funcionamiento. La entrada MODE determina los datos que se devolverán a la ubicación especificada para la información de STATE. Los modos son los siguientes: • 1: Configuración de dispositivo activa • 2: Dispositivo defectuoso • 3: Dispositivo deshabilitado • 4: El dispositivo existe • 5: Problema en el dispositivo

Parámetro y tipo		Tipo de datos	Descripción
RET_VAL	OUT	Int	Código de condición de ejecución
STATE[1]	InOut	Variant	Búfer que recibe el estado de error de cada dispositivo: El tipo de datos seleccionado para el parámetro STATE puede ser cualquier tipo de bit (Bool, Byte, Word o DWord) o una matriz del tipo bit. • El bit 0 del primer byte de los datos STATE devueltos es un bit de resumen. Cuando se establece en TRUE, indica que los otros datos están disponibles. • Los datos que devuelve el parámetro STATE muestran una correlación unívoca entre una ubicación de bit y una dirección de E/S descentralizadas. El direccionamiento de este dispositivo es TRUE para PROFIBUS y PROFINET. Por ejemplo, el bit 4 del primer byte se correlaciona con la dirección 4 de PROFIBUS o el número de dispositivo 4 de PROFINET.

[1] Para PROFIBUS-DP, la longitud de la información de estado es 128 bits. Para PROFINET IO, la longitud es de 1024 bits.

Tras la ejecución, el parámetro STATE contiene el estado de error de cada dispositivo E/S en forma de lista de bits (para LADDR y MODE asignados).

Tabla 9-158 Códigos de condición

RET_VAL (W#16#...)	Descripción
0	No hay error
8091	LADDR no existe.
8092	LADDR no direcciona un sistema I/O.
8093	Tipo de datos no válido asignado para el parámetro STATE: Los tipos de datos válidos son (Bool, Byte, Word o Dword) o una matriz de (Bool, Byte, Word o Dword).
80Bx	La CPU no soporta la instrucción DeviceStates para este LADDR.
8452	La longitud de todos los datos de estado es demasiado larga para el parámetro STATE asignado. El búfer de STATE contiene un resultado parcial.

9.7.7.1 Configuraciones de ejemplo de DeviceStates

Ejemplo de PROFIBUS

El ejemplo PROFIBUS comprende lo siguiente:

- 16 Los dispositivos PROFIBUS denominados de "DPSlave_10" a "DPSlave_25"
- Los 16 dispositivos PROFIBUS utilizan las direcciones PROFIBUS de 10 a 25, respectivamente.
- Cada esclavo se configura con varios módulos de E/S.
- Se muestran los primeros cuatro bytes de la información del parámetro STATE devuelto.

MODE	Ejemplo 1: Operación normal sin errores	Ejemplo 2: PROFIBUS Dispositivo esclavo DPSlave_12 con un único módulo extraído	Ejemplo 3: PROFIBUS Dispositivo esclavo DPSlave_12 desconectado
1: Configuración de dispositivo activa	0x01FC_FF03	0x01FC_FF03	0x01FC_FF03
2: Dispositivo defectuoso	0x0000_0000	0x0110_0000	0x0110_0000
3: Dispositivo deshabilitado	0x0000_0000	0x0000_0000	0x0000_0000
4: El dispositivo existe	0x01FC_FF03	0x01FC_FF03	0x01EC_FF03
5: Problema en el dispositivo	0x0000_0000	0x0110_0000	0x0110_0000

Las siguientes cuatro tablas muestran un desglose binario de los cuatro bytes de datos que se analizan:

Tabla 9-159 Ejemplo 1: Sin errores: se devuelve un valor de 0x01FC_FF03 para MODE 1 (configuración de dispositivo activa).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFC	Bit 15 1111-1100 Bit 8	
Byte 3 0xFF	Bit 23 1111-1111 Bit 16	
Byte 4 0x03	Bit 31 0000-0011 Bit 24	

Los dispositivos se configuran en las direcciones 10 (bit 10) a 25 (bit 25).
No hay dispositivos configurados en las direcciones 1 a 9.
MODE 4 (el dispositivo existe) datos coincidentes MODE 1 (configuración de dispositivo activa), de manera que los dispositivos configurados coinciden con los dispositivos existentes.

Tabla 9-160 Ejemplo 2: Se ha extraído un módulo del dispositivo esclavo PROFIBUS "DPSlave_12". Se devuelve un valor de 0x0110_0000 para MODE 2 (dispositivo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x10	Bit 15 0001-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

El dispositivo 12 (bit 12) está marcado como defectuoso.
MODE 5 (problema en el dispositivo) devuelve la misma información que MODE 2 (dispositivo defectuoso).

Tabla 9-161 Ejemplo 2 (continuación): Se ha extraído un módulo del dispositivo esclavo PROFIBUS "DPSlave_12". Se devuelve un valor de 0x01FC_FF03 para MODE 4 (el dispositivo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFC	Bit 15 1111-1100 Bit 8	

Byte con valor	Patrón de bit con valor	Notas
Byte 3 0xFF	Bit 23 1111-1111 Bit 16	
Byte 4 0x03	Bit 31 0000-0011 Bit 24	

Aunque el dispositivo 12 (bit 12) tiene un error, como se muestra en MODE 2 (dispositivo defectuoso) de arriba, el dispositivo todavía funciona en la red, lo que hace que MODE 4 (el dispositivo existe) muestre el dispositivo como "dispositivo existente".

Tabla 9-162 Ejemplo 3: El dispositivo esclavo PROFIBUS "DPSlave_12" está desconectado (cable desconectado o pérdida de corriente) de la red PROFIBUS. Todavía se detecta "DPSlave_12" como dispositivo defectuoso, así como un error en el dispositivo. La diferencia es que "DPSlave_12" ya no se detecta y que existe un dispositivo. Se devuelve un valor de 0x01EC_FF03 para MODE 4 (el dispositivo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xEC	Bit 15 111**0**-1100 Bit 8	
Byte 3 0xFF	Bit 23 1111-1111 Bit 16	
Byte 4 0x03	Bit 31 0000-0011 Bit 24	

El dispositivo 12 (bit 12) está marcado como no existente. Con esta excepción, los dispositivos de 10 a 25 todavía se notifican como existentes.

Ejemplo de PROFINET

El ejemplo PROFINET comprende lo siguiente:

- 16 Los dispositivos esclavos PROFINET denominados de "et200s_1" a "et200s_16"
- Los 16 dispositivos PROFINET utilizan los números de dispositivos PROFINET de 1 a 16, respectivamente.
- Cada esclavo se configura con varios módulos de E/S.
- Se muestran los primeros cuatro bytes de la información del parámetro STATE devuelto.

MODE	Ejemplo 1: Operación normal sin errores	Ejemplo 2: PROFINET Módulo esclavo et200s_1 extraído	Ejemplo 3: PROFINET Esclavo et200s_1 desconectado
1: Configuración de dispositivo activa	0xFFFF_0100	0xFFFF_0100	0xFFFF_0100
2 - Dispositivo defectuoso	0x0000_0000	0x0300_0000	0x0300_0000
3 - Dispositivo deshabilitado	0x0000_0000	0x0000_0000	0x0000_0000
4 - El dispositivo existe	0xFFFF_0100	0xFFFF_0100	0xFDFF_0100
5 - Problema en el dispositivo	0x0000_0000	0x0300_0000	0x0300_0000

Las siguientes cuatro tablas muestran un desglose binario de los cuatro bytes de datos que se analizan:

Tabla 9-163 Ejemplo 1: Sin errores: se devuelve un valor de 0xFFFF_0100 para MODE 1 (configuración de dispositivo activa).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFF	Bit 15 1111-1111 Bit 8	
Byte 3 0x01	Bit 23 0000-0001 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Los dispositivos se configuran en las direcciones 1 (bit 1) a 16 (bit 16).
No hay dispositivos configurados en las direcciones 1 a 9.
MODE 4 (el dispositivo existe) datos coincidentes MODE 1 (configuración de dispositivo activa), de manera que los dispositivos configurados coinciden con los dispositivos existentes.

Tabla 9-164 Ejemplo 2: Se ha desconectado un módulo del dispositivo esclavo PROFINET "et200s_1". Se devuelve un valor de 0x0300_0000 para MODE 2 (dispositivo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x03	Bit 7 0000-00**1**1 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

El dispositivo 1 (bit 1) está marcado como defectuoso. Puesto que el dispositivo todavía existe, MODE 4 (el dispositivo existe) muestra los mismos datos que durante la operación normal.
MODE 5 (problema en el dispositivo) devuelve la misma información que MODE 2 (dispositivo defectuoso).

Tabla 9-165 Ejemplo 2 (continuación): Se ha desconectado un módulo del dispositivo esclavo PROFIBUS "et200s_1". Se devuelve un valor de 0xFFFF_0100 para MODE 4 (el dispositivo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFF	Bit 7 1111-11**1**1 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFF	Bit 15 1111-1111 Bit 8	
Byte 3 0x01	Bit 23 0000-0001 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Aunque el dispositivo 1 (bit 1) tiene un error, como se muestra en MODE 2 (dispositivo defectuoso) de arriba, el dispositivo todavía funciona en la red, lo que hace que MODE 4 (el dispositivo existe) muestre el dispositivo como "dispositivo existente".

Tabla 9-166 Ejemplo 3: El dispositivo esclavo PROFINET "et200s_1" está desconectado (cable desconectado o pérdida de corriente) de la red PROFINET. Se devuelve un valor de 0xFDFF_0100 para MODE 4 (el dispositivo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFD	Bit 7 1111-11**0**1 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFF	Bit 15 1111-1111 Bit 8	
Byte 3 0x01	Bit 23 0000-0001 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

El dispositivo 1 (bit 1) no existe. Los dispositivos de 2 (bit 2) a 16 (bit 16) existen.

9.7.8 Instrucción ModuleStates

Puede utilizar la instrucción ModuleStates para devolver el estado de todos los módulos en una estación PROFIBUS o PROFINET.

Tabla 9-167 Instrucción ModuleStates

KOP / FUP	SCL	Descripción

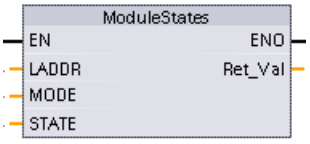
	`ret_val := ModuleStates(`
	`  laddr:=_word_in_,`
	`  mode:=_uint_in,`
	`  state:=_variant_inout);`
		ModuleStates determina los estados operativos de los módulos de E/S. Tras la ejecución, el parámetro STATE contiene el estado de error de cada módulo de E/S en forma de lista de bits (para LADDR y MODE asignados). Esta información se corresponde con el estado del módulo que se indica en la vista de diagnóstico de STEP 7.
		La entrada LADDR de ModuleStates utiliza un identificador de hardware de una estación de E/S descentralizadas y no del propio módulo de cabecera. El identificador de hardware puede encontrarse seleccionando toda la estación en la vista de red y buscando la sección del identificador de hardware debajo de las propiedades. También puede encontrarse buscando los tipos de datos "Hw_Device" y "Hw_DpSlave" en la pestaña de constantes del sistema en la tabla de variables PLC.

Tabla 9-168 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
LADDR	IN	HW_DEVICE	Dirección lógica (identificador de los módulos de E/S)
MODE	IN	UInt	Admite cinco modos de funcionamiento. La entrada MODE determina los datos que se devolverán a la ubicación especificada para la información de STATE. Los modos son los siguientes: • 1: Configuración de módulo activa • 2: Módulo defectuoso • 3: Módulo deshabilitado • 4: El módulo existe • 5: Problema en el módulo
RET_VAL	OUT	Int	Estado (código de condición)
STATE[1]	InOut	Variant	Búfer que recibe el estado de error de cada módulo: El tipo de datos utilizado para el parámetro STATE puede ser cualquier tipo de bit (Bool, Byte, Word o DWord) o una matriz del tipo bit. • El bit 0 del primer byte de los datos STATE devueltos es un bit de resumen. Cuando se establece en TRUE, indica que los otros datos están disponibles. • Los datos que devuelve el parámetro STATE muestran una correlación unívoca entre una ubicación de bit y una posición de módulo. El direccionamiento por slot es TRUE para PROFIBUS y PROFINET. Así, por ejemplo, para un ET 200SP con un módulo de cabecera, un módulo de potencia y dos módulos E/S, el bit 1 del primer byte equivale al módulo de cabecera, el bit 2 al módulo de potencia y los bits 3 y 4 a los módulos E/S, respectivamente.

[1] Pueden asignarse 128 bits como máximo. El número de bits necesarios depende del uso del módulo de E/S.

Tabla 9-169 Códigos de condición

RET_VAL (W#16#...)	Descripción
0	No hay error
8091	El módulo identificado por LADDR no existe.
8092	El módulo identificado por LADDR no direcciona un dispositivo de E/S.
8093	Tipo de datos no válido para el parámetro STATE: Los tipos de datos válidos son (Bool, Byte, Word o Dword) o una matriz de (Bool, Byte, Word o Dword).
80Bx	La instrucción ModuleStates no es soportada por la CPU para este LADDR.
8452	La longitud de todos los datos de estado es demasiado larga para el parámetro STATE asignado. El búfer de STATE contiene un resultado parcial.

9.7.8.1 Configuraciones de ejemplo de ModuleStates

Ejemplo de PROFIBUS

El ejemplo PROFIBUS comprende lo siguiente:

- 16 Los dispositivos PROFIBUS denominados de "DPSlave_10" a "DPSlave_25"
- Los 16 dispositivos PROFIBUS utilizan las direcciones PROFIBUS de 10 a 25, respectivamente.
- Cada esclavo se configura con varios módulos de E/S.
- El ejemplo utiliza el parámetro LADDR del esclavo PROFIBUS "DPSlave_12", que contiene un módulo de cabecera, un módulo de potencia y dos módulos de E/S.
- Se muestran los primeros cuatro bytes de la información del parámetro STATE devuelto.

MODE	Ejemplo 1: Operación normal sin errores	Ejemplo 2: PROFIBUS Módulo DPSlave_12 del dispositivo esclavo extraído	Ejemplo 3: PROFIBUS Dispositivo esclavo DPSlave_12 desconectado
1: Configuración de módulo activa	0x1F00_0000	0x1F00_0000	0x1F00_0000
2: Módulo defectuoso	0x0000_0000	0x0900_0000	0x1F00_0000
3: Módulo deshabilitado	0x0000_0000	0x0000_0000	0x0000_0000
4: El módulo existe	0x1F00_0000	0x1700_0000	0x0000_0000
5: Problema en el módulo	0x0000_0000	0x0900_0000	0x1F00_0000

Las siguientes cuatro tablas muestran un desglose binario de los cuatro bytes de datos que se analizan:

Tabla 9-170 Ejemplo 1: Sin errores: se devuelve un valor de 0x1F00_0000 para MODE 1 (configuración de módulo activa).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x1F	Bit 7 0001-1111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Los slots de 1 (bit 1) a 4 (bit 4) contienen módulos. Los slots 5 (bit 5) y en adelante no contienen módulos. MODE 4 (el módulo existe) datos coincidentes MODE 1 (configuración de módulo activa), de manera que los módulos configurados coinciden con los módulos existentes.

Tabla 9-171 Ejemplo 2: Se ha extraído un módulo del dispositivo esclavo PROFIBUS "DPSlave_12". Se devuelve un valor de 0x0900_0000 para MODE 2 (dispositivo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x09	Bit 7 0000-**1**001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	

Byte con valor	Patrón de bit con valor	Notas
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Solo el módulo 3 (bit 3) está marcado como defectuoso. Todos los demás módulos son funcionales.

Tabla 9-172 Ejemplo 2 (continuación): Se ha extraído un módulo del dispositivo esclavo PROFIBUS "DPSlave_12". Se devuelve un valor de 0x1700_0000 para MODE 4 (el módulo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x17	Bit 7 0001-**0**111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

El módulo 3 (bit 3) se muestra como ausente. Los módulos 1, 2 y 4 (bits 1, 2 y 4) se muestran como existentes.

Tabla 9-173 Ejemplo 3: El dispositivo esclavo PROFIBUS "DPSlave_12" está desconectado (cable desconectado o pérdida de corriente) de la red PROFIBUS. Se devuelve un valor de 0x1F00_0000 para MODE 2 (módulo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x1F	Bit 7 000**1**-1111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Los módulos de los slots de 1 a 4 (bits 1 a 4) se marcan como defectuosos, puesto que el dispositivo está ausente.
MODE 5 (problema en el dispositivo) muestra la misma información que MODE 2 (módulo defectuoso).

Ejemplo de PROFINET

El ejemplo PROFINET comprende lo siguiente:

- 16 Los dispositivos esclavos PROFINET denominados de "et200s_1" a "et200s_16"
- Los 16 dispositivos PROFINET utilizan los números de dispositivos PROFINET de 1 a 16, respectivamente.
- Cada esclavo se configura con varios módulos de E/S.
- El ejemplo utiliza el esclavo PROFINET "et200s_1", que contiene un módulo de cabecera, un módulo de potencia y 18 módulos de E/S.
- Se muestran los primeros cuatro bytes de la información del parámetro STATE devuelto.

MODE	Ejemplo 1: Operación normal sin errores	Ejemplo 2: PROFINET Módulo esclavo et200s_1 extraído	Ejemplo 3: PROFINET Esclavo et200s_1 desconectado
1: Configuración de módulo activa	0xFFFF_1F00	0xFFFF_1F00	0xFFFF_1F00
2: Módulo defectuoso	0x0000_0000	0x0180_0000	0xFFFF_1F00
3: Módulo deshabilitado	0x0000_0000	0x0000_0000	0x0000_0000
4: El módulo existe	0xFFFF_1F00	0xFF7F_1F00	0x0000_0000
5: Problema en el módulo	0x0000_0000	0x0180_0000	0xFFFF_1F00

Las siguientes cuatro tablas muestran un desglose binario de los cuatro bytes de datos que se analizan:

Tabla 9-174 Ejemplo 1: Sin errores: se devuelve un valor de 0xFFFF_1F00 para MODE 1 (configuración de módulo activa).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFF	Bit 15 1111-1111 Bit 8	
Byte 3 0x1F	Bit 23 0001-1111 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Los slots de 1 (bit 1) a 20 (bit 20) contienen módulos. Los slots 21 (bit 21) y en adelante no contienen módulos.
MODE 4 (el módulo existe) datos coincidentes MODE 1 (configuración de módulo activa), de manera que los módulos configurados coinciden con los módulos existentes.

Tabla 9-175 Ejemplo 2: Se ha desconectado un módulo del dispositivo esclavo PROFINET "et200s_1". Se devuelve un valor de 0x0180_0000 para MODE 2 (módulo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x80	Bit 15 **1**000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Solo el módulo 15 (bit 15) está marcado como defectuoso. Todos los demás módulos son funcionales.

Tabla 9-176 Ejemplo 2 (continuación): Se ha desconectado un módulo del dispositivo esclavo PROFIBUS "et200s_1". Se devuelve un valor de 0xFF7F_1F00 para MODE 4 (el módulo existe).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0x7F	Bit 15 **0**111-1111 Bit 8	

Byte con valor	Patrón de bit con valor	Notas
Byte 3 0x1F	Bit 23 0001-1111 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

El módulo 15 (bit 15) se muestra como ausente. Los módulos de 1 a 14 (bits de 1 a 14) y de 16 a 20 (bits de 16 a 20) se muestran como existentes.

Tabla 9-177 Ejemplo 3: El dispositivo esclavo PROFINET "et200s_1" está desconectado (cable desconectado o pérdida de corriente) de la red PROFINET. Se devuelve un valor de 0xFFFF_1F00 para MODE 2 (módulo defectuoso).

Byte con valor	Patrón de bit con valor	Notas
Byte 1 0xFF	Bit 7 **1111-1111** Bit 0	Bit 0 es true; datos disponibles.
Byte 2 0xFF	Bit 15 **1111-1111** Bit 8	
Byte 3 0x1F	Bit 23 000**1-1111** Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Los módulos de los slots de 1 a 20 (bits 1 a 20) se marcan como defectuosos, puesto que el dispositivo está ausente.
MODE 5 (problema en el dispositivo) muestra la misma información que MODE 2 (módulo defectuoso).

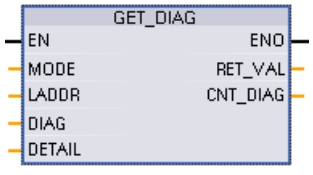
9.7.9 GET_DIAG (leer información de diagnóstico)

Descripción

Se puede usar la instrucción "GET_DIAG" para leer la información de diagnóstico de un dispositivo de hardware. El dispositivo de hardware se selecciona con el parámetro LADDR. Con el parámetro MODE se selecciona la información de diagnóstico que se va a leer.

Tabla 9-178 Instrucción GET_DIAG

KOP / FUP	SCL	Descripción

	`ret_val := GET_DIAG(`
	`  mode:=_uint_in_,`
	`  laddr:=_word_in_,`
	`  cnt_diag=>_uint_out_,`
	`  diag:=_variant_inout_,`
	`  detail:=_variant_inout_);`
		Lee la información de diagnóstico de un dispositivo de hardware asignado.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "GET_DIAG":

Tabla 9-179 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
MODE	IN	UInt	Utilice el parámetro MODE para seleccionar los datos de diagnóstico que se quieran obtener.
LADDR	IN	HW_ANY (Word)	ID de hardware del dispositivo
RET_VAL	OUT	Int	Estado de la instrucción
CNT_DIAG	OUT	UInt	Número de detalles de diagnóstico obtenidos
DIAG	InOut	Variant	Puntero a área de datos para almacenar la información de diagnóstico del modo seleccionado
DETAILS	InOut	Variant	Puntero a área de datos para almacenar los detalles de diagnóstico según el modo seleccionado

Parámetro MODE

Según el valor en el parámetro MODE, se obtienen datos diferentes de diagnóstico en los parámetros de salida DIAG, CNT_DIAG y DETAILS:

Tabla 9-180 Parámetro MODE

MODE	Descripción	DIAG	CNT_DIAG	DETAILS
0	Salida de toda la información de diagnóstico soportada para un módulo como DWord, donde Bit X=1 indica que se soporta el modo X.	Cadena de bits de los modos soportados como DWord, donde Bit X=1 indica que se soporta el modo X. Si la CPU S7-1200 ignora el parámetro LADDR cuando el parámetro MODE es 0.	0	-
1	Salida del estado inherente del objeto de hardware direccionado.	Estado de diagnóstico: Salida según la estructura DIS. (Nota: consulte la información de "Estructura DIS" a continuación, así como el ejemplo de la instrucción GET_DIAG al final del apartado.	0	-
2	Salida del estado de todos los módulos subordinados del objeto de hardware direccionado.	Salida de datos de diagnóstico según la estructura DNN. (Nota: consulte la información de "Estructura DNN" a continuación, así como el ejemplo de la instrucción GET_DIAG al final del apartado.	0	-

Estructura DIS

Con el parámetro MODE = 1, la información de diagnóstico se obtiene según la estructura DIS. En la tabla siguiente se explican los valores de cada parámetro:

Tabla 9-181 Estructura de la Diagnostic Information Source (DIS, fuente de información de diagnóstico)

Parámetro	Tipo de datos	Valor	Descripción
MaintenanceState	DWord	Enum	
		0	Mantenimiento no necesario
		1	El módulo o dispositivo está deshabilitado.
		2	-
		3	-
		4	-
		5	Mantenimiento necesario
		6	Mantenimiento solicitado
		7	Error
		8	Estado desconocido/error en módulo subordinado
		9	-
		10	Entradas/salidas no disponibles.
Componentstate Detail	DWord	Matriz de bits	Estado de los submódulos del módulo: • Bits 0 a 15: mensaje de estado del módulo • Bits 16 a 31: mensaje de estado de la CPU
		0 a 2 (enum)	Información adicional: • Bit 0: Sin información adicional • Bit 1: Transferencia no permitida
		3	Bit 3 = 1: Al menos un canal soporta calificadores de diagnóstico.
		4	Bit 4 = 1: Mantenimiento necesario para un canal o un componente como mínimo
		5	Bit 5 = 1: Mantenimiento solicitado para un canal o un componente como mínimo
		6	Bit 6 = 1: Error en un canal o componente como mínimo
		7 a 10	Reservados (siempre = 0)
		11 a 14	Bit 11 = 1: PNIO - submódulo correcto Bit 12 = 1: PNIO: módulo de repuesto Bit 13 = 1: PNIO: módulo incorrecto Bit 14 = 1: PNIO: módulo desconectado
		15	Reservados (siempre = 0)
		16 a 31	Información de estado de los módulos generada por la CPU: Bit 16 = 1: Módulo deshabilitado Bit 17 = 1: Modo CiR activo Bit 18 = 1: Entrada no disponible Bit 19 = 1: Salida no disponible Bit 20 = 1: Desbordamiento del búfer de diagnóstico Bit 21 = 1: Diagnóstico no disponible Bits 22 - 31: Reservados (siempre 0)

Parámetro	Tipo de datos	Valor	Descripción
OwnState	Uint16	Enum	El valor del parámetro OwnState describe el estado de mantenimiento del módulo.
		0	No hay fallos
		1	El módulo o dispositivo está deshabilitado.
		2	Mantenimiento necesario
		3	Mantenimiento solicitado
		4	Error
		5	La CPU no puede acceder al módulo o dispositivo (válido para módulos o dispositivos bajo una CPU).
		6	Entradas/salidas no disponibles.
		7	-
IO State	Uint16	Matriz de bits	Estado de E/S del módulo
		0	Bit 0 = 1: Mantenimiento no necesario
		1	Bit 1 = 1: El módulo o dispositivo está deshabilitado.
		2	Bit 2 = 1: Mantenimiento necesario
		3	Bit 3 = 1: Mantenimiento solicitado
		4	Bit 4 = 1: Error
		5	Bit 5 = 1: La CPU no puede acceder al módulo o dispositivo (válido para módulos o dispositivos bajo una CPU).
		6	Calificador; bit 7 = 1, si los bits 0, 2 o 3 están activados
		7	Entradas/salidas no disponibles.
		8 a 15	Reservados (siempre = 0)

Parámetro	Tipo de datos	Valor	Descripción
OperatingState	UInt16	Enum	
		0	-
		1	En STOP/actualización de firmware
		2	En STOP/borrado total
		3	En STOP/arranque automático
		4	En STOP
		5	Reset de memoria
		6	En ARRANQUE
		7	En RUN
		8	-
		9	En PARADA
		10	-
		11	-
		12	Módulo defectuoso
		13	-
		14	Sin alimentación
		15	CiR
		16	En STOP/sin DIS
		17	In
		18	
		19	
		20	

Estructura DNN

Con el parámetro MODE = 2, los detalles de información de diagnóstico se obtienen según la estructura DNN. En la tabla siguiente se explican los valores de cada parámetro:

Tabla 9-182 Estructura del Diagnostic Navigation Node (DNN, nodo de navegación de diagnóstico)

Parámetro	Tipo de datos	Valor	Descripción
SubordinateState	UINT	Enum	Estado del módulo subordinado (consultar parámetro OwnState de la estructura DIS).
SubordinateIOState	WORD	Bitarray	Estado de las entradas y salidas del módulo subordinado (consultar parámetro IO State de la estructura DIS).
DNNmode	WORD	Bitarray	• Bit 0 = 0: Diagnóstico habilitado • Bit 0 = 1: Diagnóstico deshabilitado • Bits 1 a 15: Reservado

Parámetro RET_VAL

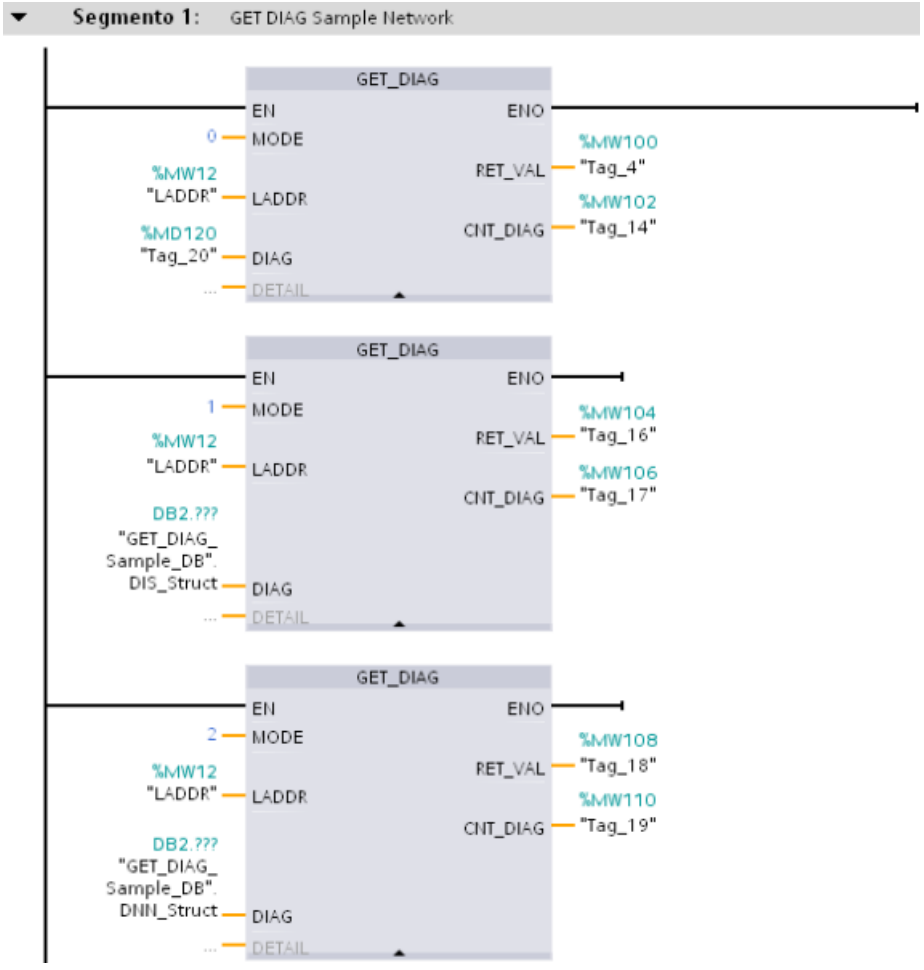
Tabla 9-183 Códigos de error del parámetro RET_VAL

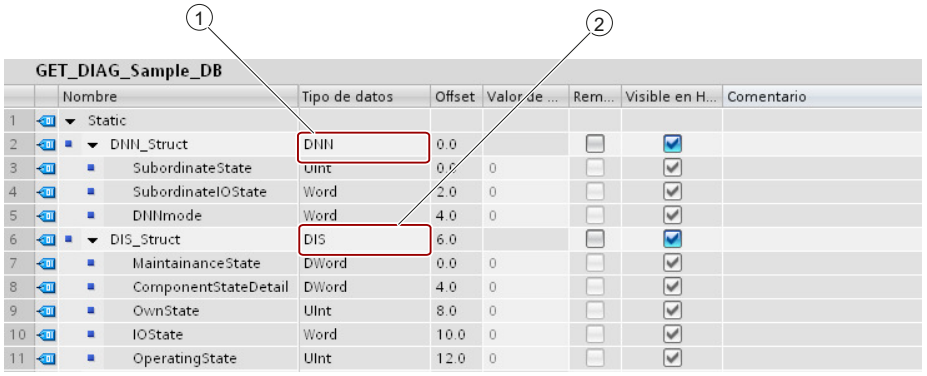
Código de error (W#16#...)	**Descripción**
0	No hay error
8080	Valor no soportado en el parámetro MODE.
8081	El tipo en el parámetro DIAG no está soportado con el modo seleccionado (parámetro MODE).
8082	El tipo en el parámetro DETAILS no está soportado con el modo seleccionado (parámetro MODE).
8090	LADDR no existe.
8091	El canal seleccionado en el parámetro CHANNEL no existe.
80C1	Falta de recursos para ejecución en paralelo

Ejemplo

En el siguiente esquema de lógica de escalera y DB se muestra cómo utilizar los tres modos con las tres estructuras:

- DIS
- DNN





① DNN

② DIS

Nota

En el DB, debe indicar manualmente el tipo de datos para acceder a cada una de las tres estructuras; no hay ninguna lista desplegable para efectuar la selección. Indique los tipos de datos exactamente como se muestra a continuación:

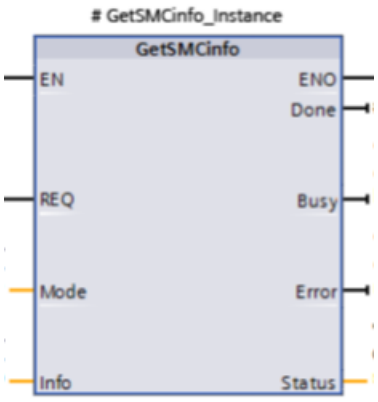
- DNN
- DIS

9.7.10 GetSMCInfo (leer información sobre la Memory Card)

La instrucción "GetSMCInfo" recupera información sobre la SIMATIC Memory Card insertada. La información que debe leerse se selecciona con el parámetro "Mode".

Tabla 9-184 Instrucción GetSMCinfo

KOP / FUP	SCL	Descripción

	`ret_val := GetSMCinfo(`
	`  Mode:=_uint_in_,`
	`  Info:=_variant_inout_);`
		Con la instrucción "GetSMCinfo" se lee la información relacionada con la SIMATIC Memory Card insertada. Si no hay insertada ninguna SIMATIC Memory Card, la instrucción devuelve el código de error W#16#8081. Utilice el parámetro "Mode" para seleccionar la información que debe leerse.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "GetSMCinfo":

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Solicitud de parámetro de control Activa la lectura de la información con REQ = "1".
Mode	IN	UInt	Utilice el parámetro Mode para seleccionar la información que debe leerse de la SIMATIC Memory Card: • 0: Tamaño de memoria en KiB (1 KiB = 1024 bytes) • 1: Memoria en uso en KiB • 2: Información de mantenimiento: Porcentaje de vida útil utilizado anteriormente Nota: – S7-1200 ahora proporciona la información de mantenimiento. Con S7-PLCSIM siempre se introduce 0x00 en Info (no hay información de mantenimiento disponible). – Si la SIMATIC Memory Card o el firmware de la CPU instalado actualmente no soportan la información, en Info se introduce 0xFF. • 3: Cuando se rebasa el umbral de porcentaje de vida útil establecido, el PLC genera una entrada en el búfer de diagnóstico y activa el LED de mantenimiento. Nota: –Si se ha desactivado la generación de alarmas de diagnóstico, en Info se introduce 0xFF. –Con S7-PLCSIM, siempre se introduce 0xFF en Info. • 10 o 20: equivale a Mode 0, pero está reservado para CPU S7-1500 R/H. • 11 o 21: equivale a Mode 1, pero está reservado para CPU S7-1500 R/H. • 12 o 22: equivale a Mode 2, pero está reservado para CPU S7-1500 R/H. • 13 o 23: equivale a Mode 3, pero está reservado para CPU S7-1500 R/H.
Done	OUT	Bool	1: la instrucción se ha ejecutado correctamente. La información leída se transfiere al parámetro Info.
Busy	OUT	Bool	Parámetro de estado • 0: ejecución de la instrucción terminada • 1: ejecución de la instrucción no terminada aún
Error	OUT	Bool	Parámetro de estado • 0: sin errores. • 1: se ha producido un error al ejecutar la instrucción Se proporciona información detallada a través del parámetro Status.
Status	OUT	Word	Código de error
Info	INOUT	UDInt	Búfer para la información leída.

Encontrará información adicional sobre los tipos de datos válidos en "AUTOHOTSPOT".

Nota

Determinar los datos I&M 0

Los datos I&M 0 de la SIMATIC Memory Card no pueden determinarse con la instrucción GetSMCinfo. Utilice para ello la instrucción "Get_IM_Data".

Estado del parámetro

Código de error* (W#16#...)	Explicación
0	Sin errores
7000	Ninguna tarea en curso.
7001	Primera llamada: activar tarea (Busy = 1, Done = 0).
7002	Llamada intermedia: tarea activa (Busy = 1, Done = 0).
8080	Valor no válido del parámetro "Mode"
8081	No hay ninguna SIMATIC Memory Card insertada
8092	No hay datos disponibles, por ejemplo, porque la CPU no soporta "GetSMCinfo"
80C3	Se ha alcanzado el número máximo de llamadas simultáneas de la instrucción "GetSMCinfo".

* Los códigos de error en el editor de programas se muestran como valores enteros o hexadecimales. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".

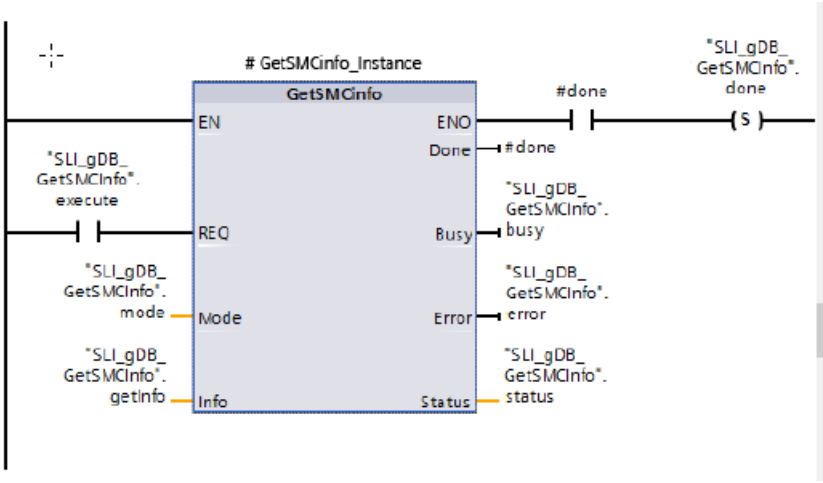
Ejemplo: Determinar la capacidad de la SIMATIC Memory Card utilizada

En el siguiente ejemplo se determina la capacidad de la SIMATIC Memory Card utilizada. Cree las siguientes variables en un bloque de datos global para guardar los datos:

SLI_gDB_GetSMCinfo

	Name	Data type	Start value
1	Static		
2	execute	Bool	false
3	mode	UInt	0
4	getinfo	UDInt	0
5	done	Bool	false
6	busy	Bool	false
7	error	Bool	false
8	status	Word	16#0

Cree un FB. En el área de estado del FB cree una variable local "#done" con el tipo de datos Bool. Interconecte los parámetros de la instrucción "GetSMCinfo" del siguiente modo.



La instrucción "GetSMCinfo" solo se ejecuta si el parámetro de entrada REQ ("execute") devuelve el estado lógico "TRUE". El modo que debe usarse para leer la SIMATIC Memory Card está guardado en el parámetro de entrada MODE ("mode"). En el ejemplo inferior, la capacidad de la SIMATIC Memory Card se lee si el valor "mode" está puesto a "0" y se muestra en KB en el parámetro INFO ("getInfo"). El estado correcto de GetSMCinfo se muestra en el parámetro de salida DONE ("#done") y se guarda en la variable "done".

Los parámetros de salida STATUS ("status") y ERROR ("error") indican que el procesamiento del ejemplo ha finalizado sin errores.

SLI_gDB_GetSMCinfo

	Name	Data type	Start value	Monitor value
1	Static			
2	execute	Bool	false	TRUE
3	mode	UInt	0	0
4	getInfo	UDInt	0	2025008
5	done	Bool	false	TRUE
6	busy	Bool	false	FALSE
7	error	Bool	false	FALSE
8	status	Word	16#0	16#0000

9.7.11 Eventos de diagnóstico de E/S descentralizadas

Nota

En un sistema PROFIBUS IO, tras una descarga o una desconexión y conexión, la CPU pasará a modo RUN a menos que se establezca la compatibilidad de hardware para permitir módulos sustitutos aceptables (Página 153) y que falten uno o más módulos o que no sea un sustituto aceptable para el módulo configurado.

Como muestra la tabla siguiente, la CPU soporta los diagnósticos que pueden configurarse para los componentes del sistema I/O descentralizado. Cada uno de estos errores genera una entrada en el búfer de diagnóstico.

Tabla 9-185 Tratamiento de eventos de diagnóstico de PROFINET y PROFIBUS

Tipo de error	¿Hay información de diagnóstico de la estación?	¿Hay una entrada en el búfer de diagnóstico?	Modo de operación de la CPU
Error de diagnóstico	Sí	Sí	Permanece en modo RUN
Fallo del rack o estación	Sí	Sí	Permanece en modo RUN
Error de acceso a E/S [1]	No	Sí	Permanece en modo RUN
Error de acceso a la periferia [2]	No	Sí	Permanece en modo RUN
Evento de presencia de módulo	Sí	Sí	Permanece en modo RUN

[1] Causa del ejemplo de error de acceso a E/S: se ha retirado un módulo.

[2] Causa del ejemplo de error de acceso a la periferia: comunicación acíclica con un submódulo que no está comunicando.

Utilice la instrucción GET_DIAG (Página 455) para obtener información de diagnóstico de cada estación. Esto le permitirá tratar desde el programa los errores encontrados en el dispositivo y, si es necesario, para la CPU a modo STOP. Este método hace necesario especificar el dispositivo de hardware desde el que leer la información de estado.

La instrucción GET_DIAG utiliza la "dirección L" (LADDR) de la estación para obtener el diagnóstico de toda la estación. Esta dirección L se encuentra en la vista de configuración de redes así como seleccionando todo el rack de la estación (todo el área gris), la dirección L se indica en la ficha Propiedades de la estación. La dirección LADDR de cada módulo se encuentra o bien en las propiedades del módulo (en la configuración del dispositivo) o en la tabla de variables predeterminada de la CPU.

9.8 Impulso

9.8.1 CTRL_PWM (modulación del ancho de impulso)

Tabla 9-186 Instrucción CTRL_PWM (modulación del ancho de impulsos)

KOP / FUP	SCL	Descripción

	`"CTRL_PWM_DB"(`
	`PWM:=_uint_in_,`
	`ENABLE:=_bool_in_,`
	`BUSY=>_bool_out_,`
	`STATUS=>_word_out_);`
		Ofrece una salida fija de tiempo de ciclo con un ciclo de trabajo variable. La salida PWM se ejecuta continuamente tras haberse iniciado a la frecuencia indicada (tiempo de ciclo). La duración de impulso varía según sea necesario para obtener el control deseado.

[1] Cuando se inserta la instrucción, STEP 7 muestra el cuadro de diálogo "Opciones de llamada" para crear el DB asociado.

[2] En el ejemplo SCL, "CTRL_PWM_DB" es el nombre del DB de instancia.

Tabla 9-187 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
PWM	IN	HW_PWM (Word)	Identificador PWM: Los nombres de los generadores de impulsos habilitados se convierten en variables en la tabla de variables "Constantes" y están disponibles para ser utilizados como parámetro PWM. (Valor predeterminado: 0)
ENABLE	IN	Bool	1= iniciar generador de impulsos 0 = parar generador de impulsos
BUSY	OUT	Bool	Función ocupada (valor predeterminado: 0)
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

La instrucción CTRL_PWM almacena la información de parámetros en el DB. El usuario no modifica por separado los parámetros del bloque de datos, sino que la instrucción CTRL_PWM los controla.

Determine el generador de impulsos habilitado que desea utilizar, utilizando su nombre de variable para el parámetro PWM.

Cuando la entrada EN es TRUE, la instrucción PWM_CTRL inicia o detiene el PWM identificado, según el valor de la entrada ENABLE. El valor de la dirección de salida de palabra Q asociada indica la duración de impulso.

Puesto que la CPU procesa la petición cuando se ejecuta la instrucción CTRL_PWM, el parámetro BUSY siempre notifica FALSE. Si se detecta un error, ENO se pone a FALSE y el parámetro STATUS contiene un código de condición.

La duración de impulso se pone al valor inicial ajustado en la configuración de dispositivos cuando la CPU pasa por primera vez al estado operativo RUN. Los valores se escriben en la ubicación de la palabra Q indicada en la configuración de dispositivos ("Direcciones de salida" / "Dirección inicial:") según sea necesario para modificar la duración de impulso. Para escribir la duración de impulso deseada en la palabra Q apropiada se utiliza una instrucción de desplazamiento, conversión, matemática o un cuadro PID. El valor de palabra Q debe estar comprendido en el rango válido (porcentaje, milésimos, diezmilésimos o formato analógico S7).

Nota

Las E/S digitales asignadas a PWM y PTO no se pueden forzar permanentemente

Las E/S digitales utilizadas por los dispositivos con modulación del ancho de impulsos (PWM) y tren de impulsos (PTO) se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a estos dispositivos, los valores de las direcciones de E/S asignadas no podrán ser modificados por la función de forzado permanente de la tabla de observación.

Tabla 9-188 Valor del parámetro STATUS

STATUS	Descripción
0	No hay error
80A1	El identificador PWM no direcciona un PWM válido.

9.8.2 CTRL_PTO (tren de impulsos)

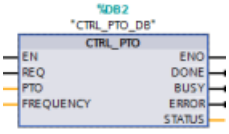
La instrucción PTO ofrece una onda cuadrada con un 50 % de salida de ciclo de carga a una frecuencia especificada. Se utiliza la instrucción CTRL_PTO para asignar la frecuencia sin un bloque de datos (DB) de eje para objetos tecnológicos (TO).

La instrucción requiere un generador de impulsos. Debe activar el generador de impulsos y seleccionar un tipo de señal en la configuración hardware. Consulte "Configurar un canal de impulsos para PWM o PTO" (Página 473) para obtener más información.

Se accede a la instrucción CTRL_PTO en las instrucciones avanzadas de las Task Cards.

Tabla 9-189 Instrucción CTRL_PTO (tren de impulsos)

KOP / FUP[1]	SCL[2]	Descripción

	`"CTRL_PTO_DB"(`
	`REQ:=_bool_in_,`
	`PTO:=_uint_in_,`
	`FREQUENCY:=_udint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		La instrucción PTO permite al usuario controlar la frecuencia para una salida de onda cuadrada (50% del ciclo de carga).

[1] Cuando se inserta la instrucción, STEP 7 muestra el cuadro de diálogo "Opciones de llamada" para crear el DB asociado.

[2] En el ejemplo SCL, "CTRL_PTO_DB" es el nombre del DB de instancia.

Tabla 9-190 Tipos de datos para los parámetros

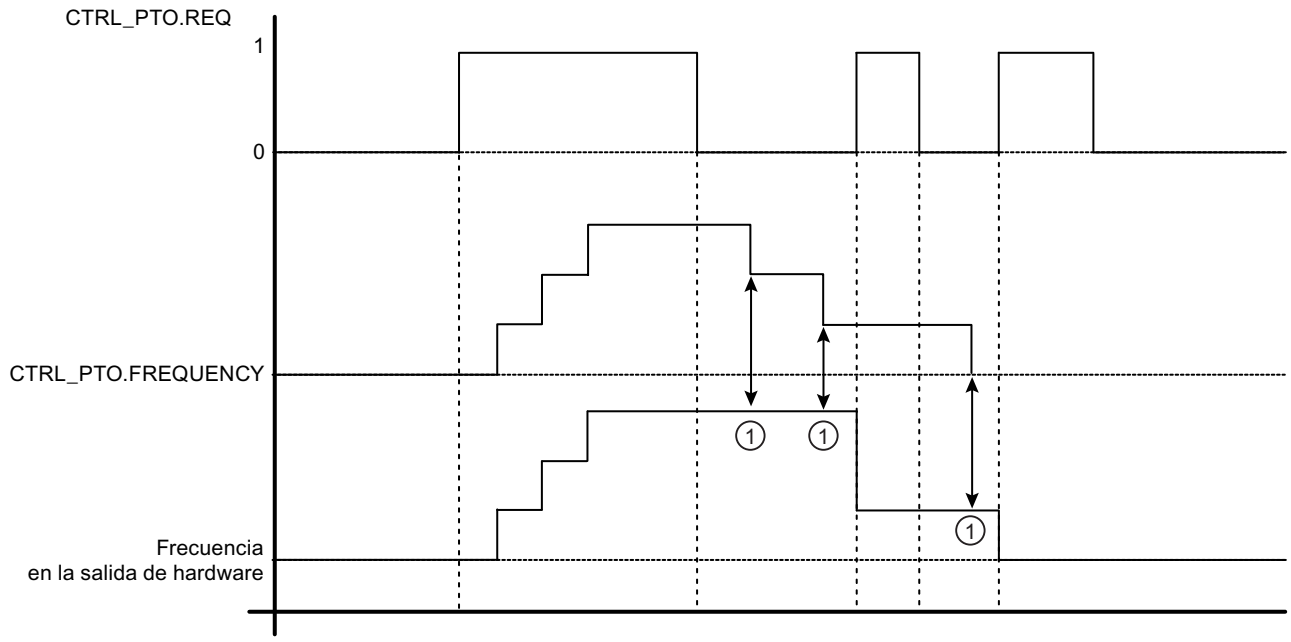
Parámetro y tipo		Tipo de datos	Descripción
EN	IN	Bool	1 = Instrucción activada 0 = Instrucción desactivada
REQ	IN	Bool	1 = Poner la frecuencia de salida de PTO al valor de la entrada FREQUENCY 0 = Sin cambios en PTO
PTO	IN	HW_PTO (Word)	Identificador PTO: ID de hardware del generador de impulsos: • Los nombres de los generadores de impulsos activados se convierten en variables en la tabla de variables "Constantes" y están disponibles para ser utilizados como parámetro PTO. (Valor predeterminado = 0) • Encontrará la ID de hardware en la vista de dispositivos de las propiedades del generador de impulsos. Las constantes de sistema también listan las ID de hardware de los generadores de impulsos. (Valor predeterminado = 0)
FREQUENCY	IN	UDInt	Frecuencia teórica (en Hz) de PTO. El valor solo se aplica si REQ = 1 (el valor predeterminado es 0 Hz)
DONE	OUT	Bool	Función finalizada sin errores (valor predeterminado: 0)
BUSY	OUT	Bool	Función ocupada (valor predeterminado: 0)
ERROR	OUT	Word	Se ha detectado un error (valor predeterminado: 0)
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

La instrucción CTRL_PTO almacena la información de parámetros en el DB. El usuario no modifica por separado los parámetros del bloque de datos, sino que la instrucción CTRL_PTO los controla.

Determine el generador de impulsos habilitado que desea utilizar, utilizando su nombre de variable o identificador de hardware para el parámetro PTO.

Cuando la entrada EN es TRUE, la instrucción CTRL_PTO inicia o detiene el PTO identificado. Cuando la entrada EN es FALSE, la instrucción CTRL_PTO no se ejecuta y el PTO mantiene su estado actual.

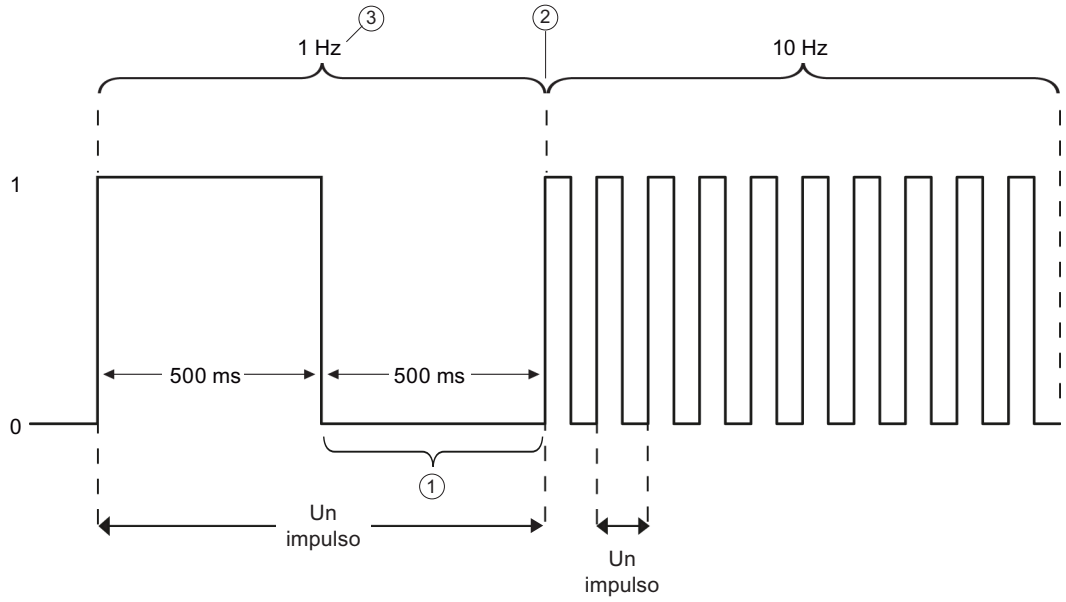
Cuando la entrada REQ se pone a TRUE, el valor de FREQUENCY se hace efectivo. Si REQ es FALSE, no es posible cambiar la frecuencia de salida del PTO, y el PTO continúa emitiendo impulsos.



(1) No hay cambios en la frecuencia de salida mientras REQ = 0

Puesto que la instrucción CTRL_PTO solo inicia el PTO, la instrucción CTRL_PTO termina inmediatamente. Como resultado, la salida BUSY nunca se activa. La salida DONE sigue activada mientras no se produce ningún error. Si se detecta un error, el parámetro ERROR se pone a TRUE y el parámetro STATUS contiene un código de condición.

Cuando el usuario activa la instrucción CTRL_PTO con una frecuencia determinada, el S7-1200 emite un tren de impulsos con dicha frecuencia. El usuario puede modificar la frecuencia teórica en cualquier momento. Si se modifica la frecuencia, el S7-1200 finaliza el impulso actual antes de cambiar la frecuencia a la nueva consigna. Así, por ejemplo, si la frecuencia teórica es de 1 Hz (que tarda 1000 ms en finalizar) y el usuario cambia la frecuencia a 10 Hz al cabo de 500 ms, la frecuencia no cambiará hasta que finalice el período de tiempo de 1000 ms.



① El usuario cambia la frecuencia a 10 Hz al cabo de 500 ms.

② El impulso de 1 Hz debe finalizar antes de que pueda cambiarse la frecuencia al nuevo valor de 10 Hz.

③ 1 Hz equivale a 1000 ms

El objeto de hardware Generador de impulsos tiene la restricción siguiente: Solo una instrucción puede utilizar el generador de impulsos como PTO y su uso es gestionado por el editor de la configuración hardware. Si otras instrucciones intentan acceder a ese PTO devuelven un error: "0x8090" (el generador de impulsos con la ID de hardware especificada está en uso).

Nota

Las E/S digitales asignadas a PWM y PTO no se pueden forzar permanentemente

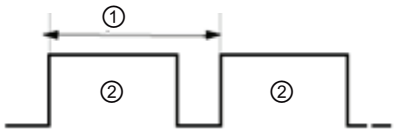
Las E/S digitales utilizadas por los dispositivos con modulación del ancho de impulsos (PWM) y tren de impulsos (PTO) se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a estos dispositivos, los valores de las direcciones de E/S asignadas no podrán ser modificados por la función de forzado permanente de la tabla de observación.

Tabla 9-191 Valor del código de error del parámetro STATUS

Código de error (W#16#...)	Descripción
0	Sin errores
0x8090	El generador de impulsos con la ID de hardware especificada está en uso.
0x8091	Frecuencia fuera de rango. La frecuencia teórica excede la frecuencia máxima de la salida de impulsos seleccionada.
0x80A1	El identificador PTO (ID de hardware) no direcciona un PTO válido.

Código de error (W#16#...)	Descripción
0x80D0	El generador de impulsos con la ID de hardware especificada no está activado. Active el generador de impulsos en "Generadores de impulsos (PTO/PWM)" de las propiedades de la CPU.
0x80D1	El generador de impulsos con la ID de hardware especificada no tiene ningún PTO seleccionado. Seleccione el PTO en la configuración hardware.

9.8.3 Manejo de las salidas de impulso



① Tiempo de ciclo
② Duración de impulso

El ancho de impulso puede expresarse en centésimas del tiempo de ciclo (0 a 100), milésimas (0 a 1000), diezmilésimas (0 a 10000) o en formato analógico S7.

La duración de impulso puede variar entre 0 (sin impulso, siempre off) y escala completa (sin impulso, siempre on).

Puesto que la salida PWM puede variar entre 0 y escala completa, ofrece una salida digital que, en numerosos aspectos, es similar a una salida analógica. Por ejemplo, la salida PWM puede utilizarse para controlar la velocidad de un motor (desde "paro" hasta "a toda velocidad") o la posición de una válvula (desde "cerrada" hasta "totalmente abierta").

La frecuencia se configura en la configuración hardware (Página 473). La duración de impulso se controla desde el programa de usuario.

Hay cuatro generadores de impulsos disponibles para controlar las funciones de salida de impulsos rápidos: PWM y PTO (tren de impulsos). PTO se usa en las instrucciones de Motion Control. Cada generador de impulsos puede asignarse a PWM o PTO, pero no a ambos simultáneamente.

Es posible utilizar las salidas integradas de la CPU o las salidas opcionales de la Signal Board. La tabla siguiente muestra los números de las salidas (suponiendo que se utiliza la configuración predeterminada de las salidas). Si se ha modificado la numeración de las salidas, sus números serán los que se hayan asignado. Tenga en cuenta que PWM solo requiere una salida, mientras que PTO puede utilizar opcionalmente dos salidas por canal. Si una salida no se requiere para una función de impulsos, estará disponible para otros usos. Consulte la tabla siguiente para la asignación de E/S.

En la tabla que se muestra más abajo se muestran las asignaciones de E/S predeterminadas; sin embargo, los cuatro generadores de impulsos se pueden configurar para cualquier salida digital

de CPU integrada o SB. Las diferentes salidas admiten tensiones y velocidades distintas, por lo que debe tenerlo en cuenta al asignar las ubicaciones PWM/PTO.

Nota

Los trenes de impulsos no pueden ser utilizados por otras instrucciones dentro del programa de usuario.

Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con la PWM o con instrucciones de PTO de Motion Control), las direcciones de salidas correspondientes se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa de usuario. Si el programa de usuario escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

Nota

Las salidas de sentido de PTO se pueden liberar para usarlas en cualquier otro lugar del programa.

Para cada PTO se necesita asignar dos salidas: una como salida de impulsos y otra como salida de sentido. Puede usar la salida de impulsos, pero no la salida de sentido. Puede liberar la salida de sentido para otros fines en el programa del usuario.

Tabla 9-192 Asignaciones de salidas predeterminadas para los generadores de impulsos[3]

Descripción	Impulso	Sentido
PTO1		
E/S incorporadas	Q0.0	Q0.1
E/S de la SB	Q4.0	Q4.1
PWM1		
Salidas incorporadas	Q0.0	-
Salidas de la SB	Q4.0	-
PTO2		
E/S incorporadas	Q0.2	Q0.3
E/S de la SB	Q4.2	Q4.3
PWM2		
Salidas incorporadas	Q0.2	-
Salidas de la SB	Q4.2	-
PTO3		
E/S incorporadas	Q0.4[1]	Q0.5[1]
E/S de la SB	Q4.0	Q4.1
PWM3		
Salidas incorporadas	Q0.4[1]	-
Salidas de la SB	Q4.1	-
PTO4		
E/S incorporadas	Q0.6[2]	Q0.7[2]
E/S de la SB	Q4.2	Q4.3
PWM4		

Descripción	Impulso	Sentido
Salidas incorporadas	Q0.6[2]	-
Salidas de la SB	Q4.3	-

[1] La CPU 1211C no tiene las salidas Q0.4, Q0.5, Q0.6 ni Q0.7. Por lo tanto, esas salidas no pueden usarse en la CPU 1211C.

[2] La CPU 1212C no tiene las salidas Q0.6 ni Q0.7. Por lo tanto, esas salidas no pueden usarse en la CPU 1212C.

[3] Esta tabla es válida para las funciones de PTO/PWM de la CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C y CPU 1217C.

9.8.4 Configurar un canal de impulso para PWM o PTO

Con el fin de prepararse para el funcionamiento de PWM o PTO, en primer lugar debe configurarse un canal de impulso en la configuración de dispositivos seleccionando la CPU, después el generador de impulsos (PTO/PWM) y eligiendo de PWM1/PTO1 a PWM4/PTO4. Habilite el generador de impulsos (casilla de verificación). Si está habilitado un generador de impulsos, se le asigna un nombre predeterminado y unívoco. El nombre puede cambiarse editándolo en el cuadro "Nombre:", pero debe ser un nombre unívoco. Los nombres de los generadores de impulsos habilitados se convierten en variables en la tabla de variables "Constantes" y están disponibles para ser utilizados del siguiente modo:

- Parámetro PWM de la instrucción CTRL_PWM
- Parámetro PTO de la instrucción CTRL_PTO

También es posible escribir un comentario relacionado con este generador de impulsos concreto en el cuadro de edición "Comentario:".

Tabla 9-193 Salida de la CPU: Frecuencia máxima (PTO) y tiempo de ciclo mínimo (PWM)

CPU	Canal de salida de la CPU	Frecuencia máxima PTO	Tiempo de ciclo mínimo PWM
1211C	De Qa.0 a Qa.3	100 kHz	10 µs
1212C	De Qa.0 a Qa.3	100 kHz	10 µs
	Qa.4, Qa.5	20 kHz	50 µs
1214C y 1215C	De Qa.0 a Qa.3	100 kHz	10 µs
	De Qa.4 a Qb.1	20 kHz	50 µs
1217C	De DQa.0 a DQa.3 (de .0+, .0- a .3+, .3-)	1 MHz	1 µs
	De DQa.4 a DQb.1	100 kHz	10 µs

Tabla 9-194 Salida de la Signal Board (SB): Frecuencia máxima (PTO) y tiempo de ciclo mínimo (PWM)

Signal Board (SB)	Canal de salida de la SB	Frecuencia máxima PTO	Tiempo de ciclo mínimo PWM
SB 1222, 200 kHz	De DQe.0 a DQe.3	200 kHz	5 µs
SB 1223, 200 kHz	DQe.0, DQe.1	200 kHz	5 µs
SB 1223	DQe.0, DQe.1	20 kHz	50 µs

Nota

El tiempo de ciclo mínimo de cada una de las salidas de la CPU y la Signal Board se indica en las tablas anteriores. Sin embargo, el TIA Portal no alerta cuando se configura un generador de impulsos PWM con un tiempo de ciclo inferior al tiempo de ciclo mínimo del hardware. Podría tener problemas en la aplicación. Por tanto, asegúrese de que el tiempo de ciclo esté siempre dentro de los límite del hardware.

Nota

Cuando se ajusta la duración de impulso de una señal PWM, el tiempo real de duración del impulso (tiempo durante el que el impulso está "high") debe ser mayor o igual a 1 milisegundo si la base de tiempo es "milisegundos". Si la base de tiempo es "microsegundos", el tiempo real de duración del impulso debe ser mayor o igual a 1 microsegundo. La salida se desactiva si el tiempo de duración del impulso es menor que 1 "base de tiempo".

Así, por ejemplo, un tiempo de ciclo de 10 microsegundos y una duración de impulso de 5 centésimas genera un tiempo de duración del impulso de 0,5 microsegundos. Puesto que es menor que 1 microsegundo, la señal PWM está desactivada.

Parametrización

La sección de parametrización permite al usuario configurar los parámetros del impulso de salida. Las funciones siguientes están disponibles en función de si se ha seleccionado PWM o PTO:

- Tipo de señal: Configure la salida de impulsos como PWM o PTO. Si se selecciona PTO, consulte "Sincronización" para más información:
  - PWM
  - PTO (impulso A y sentido B)
  - PTO (impulso ascendente A e impulso descendente B)
  - PTO (A/B con desplazamiento de fase)
  - PTO (A/B con desplazamiento de fase - cuádruple)
- Base de tiempo (solo aplicable a PWM): Seleccione las unidades de tiempo que se utilizarán:
  - milisegundos
  - microsegundos

- Formato de la duración de impulso (solo aplicable a PWM): Asigne la resolución de la duración de impulso (ancho):
  - Centésimas (0 a 100)
  - Milésimos (0 a 1000)
  - Diezmilésimos (0 a 10000)
  - Formato analógico S7 (0 a 27648)
- Tiempo de ciclo (solo aplicable a PWM): Asigne el tiempo que se tardará en finalizar un impulso (el tiempo de impulso "high" más el tiempo de impulso "low" es igual al tiempo de ciclo). El tiempo de ciclo se puede modificar en tiempo de ejecución seleccionando la casilla de verificación "Permitir modificar el tiempo de ciclo en tiempo de ejecución". Encontrará más información en el apartado inferior "Direcciones de periferia". El rango válido está comprendido entre 1 y 16.777.215 unidades de tiempo.
- Duración del impulso inicial (solo aplicable a PWM): Asigne la duración del primer impulso. Este valor puede modificarse en tiempo de ejecución por medio de la dirección de palabra Q configurada en las direcciones E/S. El rango se basa en el formato de la duración de impulso.
- Permitir modificar el tiempo de ciclo en tiempo de ejecución (solo aplicable a PWM): Si se selecciona esta opción, el programa puede modificar el tiempo de ciclo de la señal PWM mientras se está ejecutando. Encontrará más información en el apartado inferior "Direcciones de periferia".

Nota

Cuando se ajusta la duración de impulso de una señal PWM hay que tener en cuenta el retardo de conmutación del canal de salida tal y como se indica en el anexo A. La duración de impulso real medida en la salida puede ser mayor que la duración de impulso seleccionada. El incremento de la duración de impulso es más pronunciado para una duración de impulsos pequeña y frecuencias mayores. Asegúrese de verificar que la duración de impulso medida en la salida cumple sus requisitos.

Determinar el valor de duración de impulso

La "duración de impulso" se calcula multiplicando la "duración de impulso inicial" por el "tiempo de ciclo". Cuando se selecciona una "Base de tiempo", un "Formato de la duración de impulso", un "Tiempo de ciclo" y una "Duración de impulso inicial" hay que tener en cuenta que la "Duración de impulso" general no puede ser un valor fraccionario. Si la "Duración de impulso" resultante es un valor fraccionario, debe ajustarse la "Duración de impulso inicial" o cambiarse la "Base de tiempo" para generar un valor entero.

Aquí tiene dos ejemplos:

- Ejemplo 1: Si se seleccionan los valores siguientes:
  - Base de tiempo = milisegundos (ms)
  - Formato de duración de impulso = centésimas (0 a 100)
  - Tiempo de ciclo = 3 ms
  - Duración de impulso inicial = 75

  La "duración de impulso" resultante = 0,75 x 3 ms = 2,25 ms
  El valor de "Duración de impulso" es fraccionario y causa un error cuando se ejecuta la instrucción CTRL_PWM. La "Duración de impulso" debe ser un valor entero.
- Ejemplo 2: Si se seleccionan los valores siguientes:
  - Base de tiempo = microsegundos (µs)
  - Formato de duración de impulso = centésimas (0 a 100)
  - Tiempo de ciclo = 3000 µs
  - Duración de impulso inicial = 75

  La "duración de impulso" resultante = 0,75 x 3000 µs = 2250 µs
  Este valor de "Duración de impulso" es un valor entero y la instrucción CTRL_PWM funciona debidamente con este valor.

Salidas de hardware

En la sección de salidas de hardware, seleccione el canal de salida en el menú desplegable. Según sea la configuración, habrá una o dos salidas para seleccionar. Si se asigna un canal de salida a un generador de impulsos, dicho canal no podrá ser utilizado por ningún otro generador de impulsos, HSC ni por el registro de la memoria imagen de proceso.

Nota

Las salidas del generador de impulsos no pueden ser utilizadas por otras instrucciones del programa de usuario

Si las salidas de la CPU o Signal Board se configuran como generadores de impulsos (para su utilización con PWM, PTO o con instrucciones de Motion Control), las direcciones de salida correspondientes se eliminarán de la memoria Q y no podrán utilizarse para ningún otro fin en el programa. Si el programa escribe un valor en una salida utilizada como generador de impulsos, la CPU no escribirá ese valor en la salida física.

Direcciones de periferia

PWM tiene dos bytes de memoria Q para "Duración de impulso". Mientras PWM se está ejecutando, es posible modificar el valor en la memoria Q asignada y cambiar la duración de impulso.

En la sección Dirección de periferia, introduzca la dirección de palabra Q en la que desee almacenar el valor de la duración de impulso.

Las direcciones predeterminadas para los valores de la duración de impulso PWM son las siguientes:

- PWM1: QW1000
- PWM2: QW1002
- PWM3: QW1004
- PWM4: QW1006

Para PWM, el valor en este punto controla la duración de impulso y se inicializa al valor de "Duración de impulso inicial:" (asignado anteriormente) cada vez que la CPU pasa de STOP a RUN. El valor de esta palabra Q se cambia en tiempo de ejecución con el fin de modificar la duración de impulso. El rango de este valor depende del formato configurado para la duración de impulso en la parametrización.

También es posible asignar cuatro bytes adicionales de memoria Q al "Tiempo de ciclo" de la señal PWM. Encontrará un diagrama de la señal PWM en Funcionamiento de las salidas pulsadas (Página 471). Al seleccionar la casilla de verificación "Permitir modificar el tiempo de ciclo en tiempo de ejecución", los dos primeros bytes retienen el valor de duración de impulso y los cuatro últimos bytes retienen el valor de tiempo de ciclo.

Mientras PWM se está ejecutando, es posible modificar el valor de la palabra doble al final de la memoria Q asignada a PWM. Con ello cambia el tiempo de ciclo de la señal PWM. Active esta opción, por ejemplo, para que la CPU asigne seis bytes a PWM1; seguidamente, decide utilizar de QB1008 a QB1013. Una vez descargado el programa e iniciada PWM, es posible modificar la duración de impulso utilizando QW1008 y el tiempo de ciclo utilizando QD1010.

Cada vez que la CPU pasa de STOP a RUN, inicializa el valor del tiempo de ciclo en la memoria Q al valor "Tiempo de ciclo" asignado anteriormente en la sección "Parametrización". Las unidades y el rango de valores del valor del tiempo de ciclo en la memoria Q son los mismos que los configurados en la sección "Parametrización".

Al seleccionar la casilla de verificación "Permitir modificar el tiempo de ciclo en tiempo de ejecución", el TIA Portal selecciona automáticamente una nueva dirección para la salida. La nueva dirección de salida no puede ser la misma que la dirección predeterminada para este generador de impulsos. El TIA Portal utiliza el primer bloque disponible de seis bytes consecutivos. Si no encuentra ningún bloque disponible de la memoria Q antes de llegar al final de la misma, vuelve a iniciar la búsqueda en la dirección "0" de la memoria Q y sigue buscando un bloque disponible.

Un generador de impulsos configurado para PTO no utiliza la dirección de palabra Q.

9.9 Recetas y registros de datos

9.9.1 Recetas

9.9.1.1 Vista general de la receta

Almacenamiento de datos de la receta

- El bloque de datos de receta que cree en su proyecto debe almacenarse en la memoria de **carga** de la CPU. Puede utilizarse la memoria interna de la CPU o una tarjeta de "programa" como memoria externa.
- Otro DB que debe crear es el bloque de datos de receta activa. Este DB debe estar en la memoria de **trabajo**, donde un registro de receta activa se lee o graba con su lógica de programa.

Gestión de datos de receta

El DB de receta utiliza una matriz de registros de receta de producto. Cada elemento de la matriz de receta representa un sabor de receta diferente que se basa en un conjunto de componentes común.

- Debe crear un tipo de datos o estructura PLC que defina todos los componentes de un registro de receta. Esta plantilla de tipo de datos se reutiliza para todos los registros de receta. Las recetas del producto varían en función de los valores de arranque que se asignan a los componentes de la receta.
- Puede transferirse una de las recetas en cualquier momento del DB de receta (todas las recetas de la memoria de carga) al DB de receta activa (una receta en la memoria de trabajo) utilizando la instrucción READ_DBL. Después de mover un registro de receta a la memoria de trabajo, la lógica de programa puede leer los valores de los componentes e iniciar una ejecución de producción. Esta transferencia minimiza la cantidad de memoria de trabajo de CPU necesaria para los datos de receta.
- Si los valores de los componentes de la receta activa se ajustan mediante un dispositivo HMI durante una ejecución de producción, puede volver a escribir los valores modificados en el DB de receta utilizando la instrucción WRIT_DBL.

Exportación de recetas (del DB de receta al archivo CSV)

El conjunto completo de registros de receta puede generarse en forma de archivo CSV utilizando la instrucción RecipeExport. Los registros de receta no utilizados también se exportan.

Importación de recetas (del archivo CSV al DB de receta)

Una vez completada la operación de exportación, puede utilizar el archivo CSV generado como plantilla de estructura de datos.

1. Utilice la página File Browser en el servidor web de la CPU para cargar un archivo CSV de receta existente de la CPU en un PC.
2. Modifique el CSV de la receta con un editor de texto ASCII. Puede modificar los valores de arranque asignados a los componentes, pero no los tipos de datos o la estructura de datos.
3. Vuelva a cargar a la CPU el archivo CSV modificado del PC. Sin embargo, el archivo CSV anterior en la memoria de carga de la CPU (con el mismo nombre) debe suprimirse o renombrarse para que el servidor web de la CPU permita la operación de carga.
4. Después de cargar el archivo CSV modificado en la CPU, puede utilizar la instrucción RecipeImport para transferir los nuevos valores de arranque del archivo CSV modificado (en la memoria de carga de la CPU) en el DB de la receta (en la memoria de carga de la CPU).

9.9.1.2 Ejemplo de receta

Recetas de ejemplo

En la tabla siguiente se muestra cómo preparar información de recetas para su uso en un DB de receta. El DB de receta de ejemplo consta de cinco registros, de los cuales se utilizan tres. El cuatro y quinto registro se dejan libres para ampliaciones posteriores. Cada fila de la tabla representa un registro que almacena el nombre de receta, los tipos de datos de componentes y los valores de los componentes.

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

Creación de un bloque de datos de receta

Nota

Reglas para los bloques datos de receta

- El DB de receta debe contener una matriz de una sola dimensión de un tipo de datos de PLC o una estructura. El ejemplo de receta muestra cómo crear un DB de receta con un tipo de datos de PLC.
- En el ejemplo, el tipo de datos de los ingredientes de los componentes es para todos el tipo de datos UINT. Los tipos de datos de componentes también pueden ser una combinación de cualquier tipo de datos, excepto para estructuras. En un elemento de matriz de DB de receta, no se permite una estructura en un tipo de datos de PLC o una estructura.

En primer lugar, cree un nuevo tipo de datos de PLC

Añada un nuevo tipo de datos PLC cuyo nombre sea el tipo de receta. En la imagen siguiente, "Beer_Recipe" es el nuevo tipo de datos de PLC complejo que almacena una secuencia de tipos de datos simples. El tipo de datos de PLC "Beer_Recipe" es una plantilla de datos que se reutiliza en cada registro de DB de receta y también en el DB de la receta activa. Introduzca los nombres de componentes y los tipos de datos comunes a todas las recetas de ejemplo. Los valores individuales de los componentes se añaden posteriormente al DB de receta.

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

En segundo lugar, cree un bloque de datos de receta

- Cree su DB de receta como un bloque de datos global con la propiedad de DB "Depositar solo en la memoria de carga" habilitada.
- El nombre de un bloque de datos de la receta se utiliza como nombre de archivo del archivo CSV correspondiente. Los caracteres del nombre del DB que asigne deben cumplir las restricciones referentes a los nombres del sistema de archivos de Windows. Los caracteres \ / : * ? " < > | y el espacio no están permitidos.
- La asignación de la matriz de receta es "Products" como Array [1.. 5] of "Beer_Recipe". El tamaño de matriz 5 es el número máximo posible de sabores de la receta.
- Los valores de los componentes de la receta se añaden como valores de arranque del DB.

En la imagen siguiente, la receta "BlackBeer" se amplía para mostrar todos los componentes de un registro de receta.

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"	...	

Exportación de recetas (del DB de receta al archivo CSV)

La ejecución de "RecipeExport (Página 483)" transfiere los datos del DB de receta a un archivo CSV, como se muestra en el siguiente archivo de texto.

```
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
```

Importación de recetas (del archivo CSV al DB de receta)

1. Utilice la página File Browser (Página 889) en el servidor web para descargar en un PC un archivo CSV de receta existente de la memoria de carga de la CPU.
2. Modifique el CSV de la receta con un editor de texto ASCII. Puede modificar los valores de arranque asignados a los componentes, pero no los tipos de datos o la estructura de datos.
3. Vuelva a cargar en la CPU el archivo CSV modificado del PC. Sin embargo, es necesario borrar o cambiar el nombre del archivo CSV en la memoria de carga de la CPU (con el mismo nombre) para que el servidor web permita la operación de carga.
4. Una vez que el archivo CSV modificado se ha cargado en la CPU, puede utilizar la instrucción RecipeImport para transferir los nuevos valores de arranque del archivo CSV modificado (en la memoria de carga de la CPU) al DB de la receta (en la memoria de carga de la CPU).

Los archivos CSV deben coincidir exactamente con la estructura del DB de receta correspondiente

- Los valores del archivo CSV pueden cambiarse, pero no se permite cambiar la estructura. La instrucción RecipeImport requiere que el número exacto de registros y componentes coincida con la estructura del DB de la receta de destino. De lo contrario, la ejecución de RecipeImport fallará. Por ejemplo, si se definen 10 recetas en el DB de la receta pero solo 6 están en uso, las líneas 7 a 10 del archivo CSV también se transferirán al DB. Debe coordinar si estos datos serán válidos o no. Por ejemplo, puede asignar una variable "Not_used" para el nombre del producto en registros de receta no utilizados.
- Si añade registros de datos al archivo de texto e importa el archivo modificado, asegúrese de que el límite de la matriz de DB de la receta que asigne tenga suficientes elementos para todos los registros de la receta.
- Automáticamente se genera un número de índice durante la exportación al archivo CSV. Si crea registros de datos adicionales, añada los números de índice consecutivos en correspondencia.
- La ejecución de RecipeImport comprueba en los datos del archivo CSV que la estructura sea correcta y que los valores se ajusten a los tipos de datos asignados en el DB de receta asociado. Por ejemplo, un tipo de datos Bool no puede almacenar un valor entero, por lo que la ejecución de RecipeImport fallará.

Mostrar datos de receta CSV en Excel

Para facilitar la lectura y edición, el archivo CSV se puede abrir en Excel. Si las comas no se reconocen como separadores decimales, utilice la función de importación de Excel para visualizar los datos de forma estructurada.

	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

Nota

Comas en el campo de nombre del elemento del tipo de datos PLC

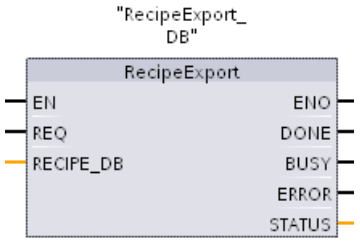
No ponga comas en el campo de nombre del o de los elemento(s) de tipo de datos PLC utilizado(s) en una receta. Si se utilizan comas en el campo de nombre, Excel insertará columnas adicionales en el archivo .csv visualizado. Estas columnas adicionales pueden provocar errores a la hora de editar los valores iniciales del archivo de registro de receta.

9.9.1.3 Instrucciones del programa que transfieren datos de receta

RecipeExport (exportación de receta)

Tabla 9-195 Instrucción RecipeExport

KOP / FUP	SCL	Descripción

	`"RecipeExport_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	
	`Recipe_DB:=_variant_inout_);`
		La instrucción "RecipeExport" exporta todos los registros de receta de un bloque de datos de receta al formato de archivo CSV. El archivo CSV contiene nombres del producto, nombres de componente y valores de arranque. El archivo se almacena en la memoria de carga interna o la memoria de carga externa, si se ha instalado una Memory Card de "programa" externa opcional.
		El parámetro "REQ" dispara la operación de exportación. El parámetro BUSY se pone a 1 durante el proceso de exportación. Después de que se detenga la ejecución de RecipeExport, BUSY se pone a cero y la finalización de la operación se indica con un "1" en el parámetro DONE. Si se produce un error durante la ejecución, los parámetros ERROR y STATUS indican el resultado.

Debe crearse un DB de receta antes de que se pueda realizar la exportación de la receta. El nombre de un bloque de datos de la receta se utiliza como nombre del nuevo archivo CSV. Si existe un archivo CSV de nombre idéntico, se sobrescribirá durante la operación de exportación.

Puede utilizar la página File Browser (Página 889) del servidor web incorporado de la CPU para acceder al archivo CSV de la receta. El archivo se coloca en la carpeta de la receta, en el directorio raíz de la memoria de carga de la CPU.

Tabla 9-196 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Parámetro de control REQUEST: Activa la exportación cuando hay un flanco ascendente.
RECIPE_DB	In/Out	Variant	Puntero al bloque de datos de receta. Para obtener más información, véase "Ejemplo de DB de receta" (Página 479). Los caracteres del nombre de DB deben cumplir las restricciones referentes a los nombres del sistema de archivos de Windows. Los caracteres \ / : * ? " < > | y el espacio no están permitidos.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores. (Valor predeterminado: False)
BUSY	OUT	Bool	Ejecución de RecipeExport • 0: Ninguna operación en curso • 1: Operación en curso

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE. • 0: Ningún aviso o error • 1: Se ha producido un error. El parámetro STATUS ofrece información detallada sobre el tipo de error.
STATUS	OUT	Word	Código de condición de ejecución

Tabla 9-197 Valores de ERROR y STATUS

ERROR	STATUS (W#16#....)	Descripción
0	0000	No hay error
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia.
1	8090	El nombre de archivo contiene caracteres no válidos.
1	8091	La estructura de datos a la que hace referencia RECIPE_DB no puede procesarse.
1	8092	La estructura de datos especificada en RECIPE_DB supera los 5000 bytes
1	80B3	No hay suficiente espacio en el MC o en la memoria de carga interna.
1	80B4	El MC está protegido contra escritura
1	80B6	El atributo de DB de receta "Depositar solo en la memoria de carga" no está habilitado.
1	80C0	El archivo CSV está temporalmente bloqueado.
1	80C1	El DB está temporalmente bloqueado.

RecipeImport (importación de receta)

Tabla 9-198 Instrucción RecipeImport

KOP / FUP	SCL	Descripción

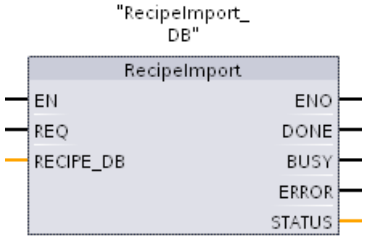
	`"RecipeImport_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`Recipe_DB:=_variant_inout_`
	`);`
		La instrucción "RecipeImport" importa los datos de receta de un archivo CSV de la memoria de carga de la CPU en un bloque de datos referenciado por el parámetro RECIPE_DB. El proceso de importación sobrescribe los valores de arranque del bloque de datos de la receta. El parámetro "REQ" dispara la operación de importación. El parámetro BUSY se pone a 1 durante el proceso de importación. Después de que se detenga la ejecución de RecipeImport, BUSY se pone a cero y la finalización de la operación se indica con un "1" en el parámetro DONE. Si se produce un error durante la ejecución, los parámetros ERROR y STATUS indican el resultado.

Tabla 9-199 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Parámetro de control REQUEST: Activa la importación cuando hay un flanco ascendente.
RECIPE_DB	In/Out	Variant	Puntero al bloque de datos de receta. Para obtener más información, véase "Ejemplo de DB de receta" (Página 479). Los caracteres del nombre de DB deben cumplir las restricciones referentes a los nombres del sistema de archivos de Windows. Los caracteres \ / : * ? " < > | y el espacio no están permitidos.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores. (Valor predeterminado: False)
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

Para que una operación de importación de receta sea posible, debe existir un DB de receta que contenga una estructura coherente con la estructura de datos del archivo CSV.

Reglas de archivo CSV:

- El archivo CSV debe encontrarse en la carpeta "Recipes" del directorio raíz de la memoria de carga interna o la memoria de carga externa, si se ha instalado una Memory Card de "programa" externa opcional.
- El nombre del archivo CSV debe coincidir con el nombre del bloque de datos en el parámetro RECIPE_DB.
- La primera línea (encabezado) del archivo CSV contiene el nombre de los componentes de receta. La primera línea se ignora durante la importación. Los nombres de los componentes de receta del archivo CSV y el bloque de datos no se concilian durante el proceso de importación.
- En cada caso el primer valor de cada línea del archivo CSV es el número de índice de la receta. Las recetas individuales se importan según el orden del índice. Para ello, el índice del archivo CSV debe presentar un orden ascendente y no debe haber espacios en blanco (si este no es el caso, se obtiene el mensaje de error 80B0 en el parámetro STATUS).
- El archivo CSV no puede contener más datos de receta de los que se ha suministrado en el bloque de datos de receta. El número máximo de registros se indica mediante los límites de matriz en el bloque de datos.

Tabla 9-200 Valores de ERROR y STATUS

ERROR	STATUS (W#16#....)	Descripción
0	0000	No hay error
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia.
1	8090	El nombre de archivo contiene caracteres no válidos.
1	8092	No se ha encontrado ningún archivo CSV coincidente para la importación. Causa posible: El nombre del archivo CSV no coincide con el nombre del DB de receta.
1	80C0	El archivo CSV está temporalmente bloqueado.
1	80C1	El bloque de datos está temporalmente bloqueado.
1	80B0	La numeración del índice del archivo CSV no es continua, no es ascendente o excede el número máximo (límite de matriz) del bloque de datos.
1	80B1	La estructura del bloque de datos de receta y el archivo CSV no coinciden: El archivo CSV contiene demasiados campos.
1	80B2	La estructura del bloque de datos de receta y el archivo CSV no coinciden: El archivo CSV contiene no contiene suficientes campos.
1	80B6	El atributo de DB de receta "Depositar solo en la memoria de carga" no está habilitado.
1	80D0 +n	La estructura del bloque de datos de receta y el archivo CSV no coinciden: El tipo de datos en el campo n no coincide (n<=46).
1	80FF	La estructura del bloque de datos de receta y el archivo CSV no coinciden: El tipo de datos en el campo n no coincide (n>46).

9.9.1.4 Programa de ejemplo de receta

Requisitos para el programa de ejemplo de receta

Los requisitos del programa de ejemplo de receta son los siguientes:

- Un DB de receta que almacena todos los registros de receta. El DB de receta se almacena en la memoria de carga.
- Un DB de receta activa que almacena una copia de una receta en la memoria de trabajo.

Véase "Ejemplo de DB de receta (Página 479)" para obtener más información sobre el DB de receta y el archivo CSV correspondiente.

Creación del DB de receta activa

En la ventana "Agregar nuevo bloque":

- Seleccione el botón "Bloque de datos" en la ventana "Agregar nuevo bloque".
- En el menú desplegable "Tipo", seleccione el tipo de datos de PLC "Beer_recipe" que ha creado anteriormente.

Los valores de arranque no son necesarios. Los valores de datos de DB se establecen cuando se transfiere una receta del DB de receta al DB de receta activa. En el ejemplo, el DB de receta activa es el destino para los datos READ_DBL y proporciona los datos de origen para WRITE_DBL. En la imagen siguiente se muestra el DB Active_Recipe.

Active_Recipe

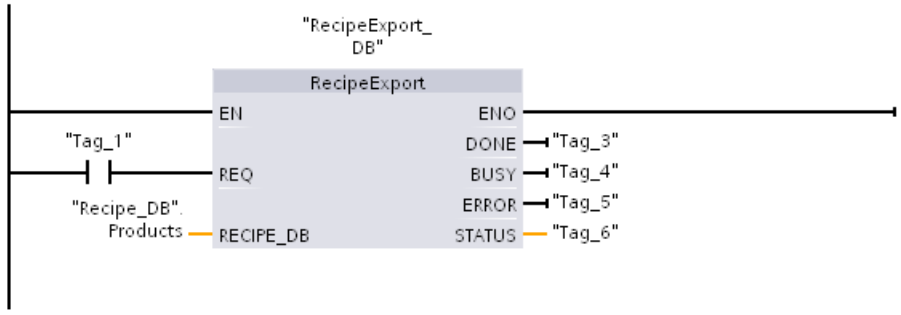
	Name	Data type	Start value
1	Static		
2	productname	String[20]	'Beer_Recipe'
3	water	UInt	0
4	barley	UInt	0
5	wheat	UInt	0
6	hops	UInt	0
7	yeast	UInt	0
8	waterTmp	UInt	0
9	mashTmp	UInt	0
10	mashTime	UInt	0
11	QTest	UInt	0

DB de instancia

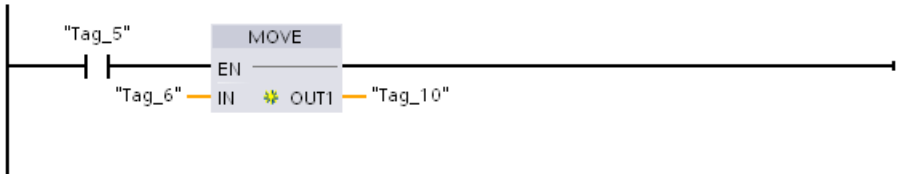
Los DB de instancia que utilizan las instrucciones RecipeExport ("RecipeExport_DB") y RecipeImport ("RecipeImport_DB") se crean automáticamente al colocar las instrucciones en el programa. Los DB de instancia se utilizan para controlar la ejecución de las instrucciones y no están referenciados en la lógica del programa.

Programa de ejemplo de receta

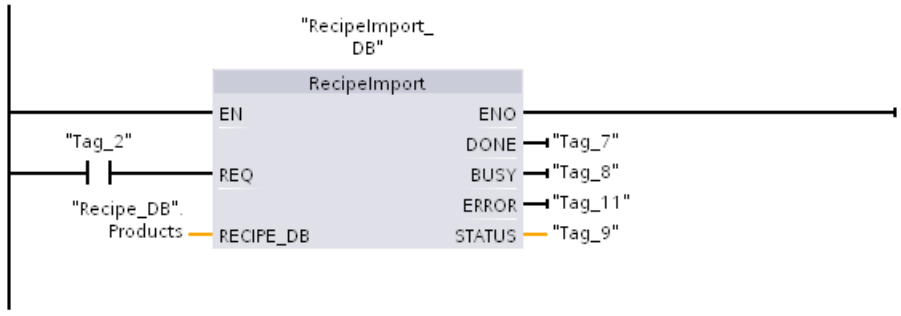
Segmento 1 Un flanco ascendente en REQ inicia el proceso de exportación. Se genera un archivo CSV a partir de los datos del DB de receta y se colocan en la carpeta de recetas de memoria de la CPU.



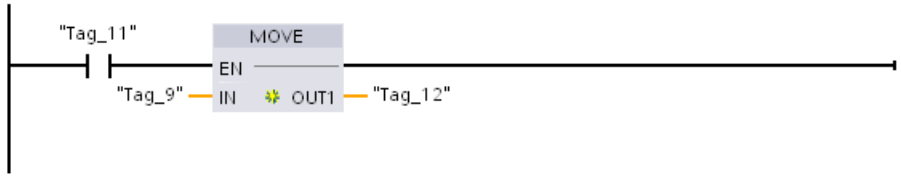
Segmento 2 Capturar la salida STATUS de la ejecución de RecipeExport porque solo es válida durante un ciclo.



Segmento 3 Un flanco ascendente en REQ inicia el proceso de importación. El DB de la receta existente se carga con todos los datos de la receta del archivo CSV correspondiente ubicado en la carpeta de recetas de la memoria de la CPU.

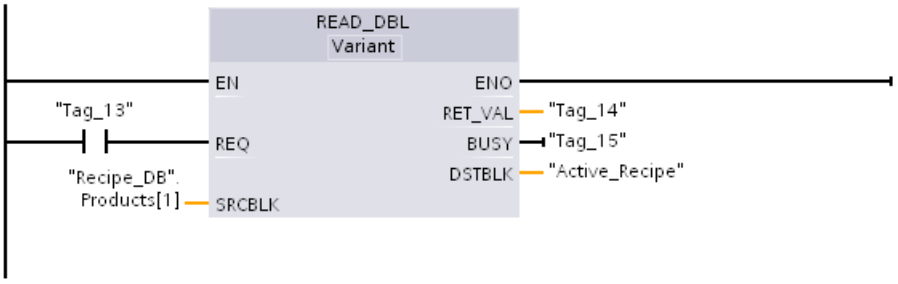


Segmento 4 Capturar la salida STATUS de la ejecución de RecipeExport porque solo es válida durante un ciclo.

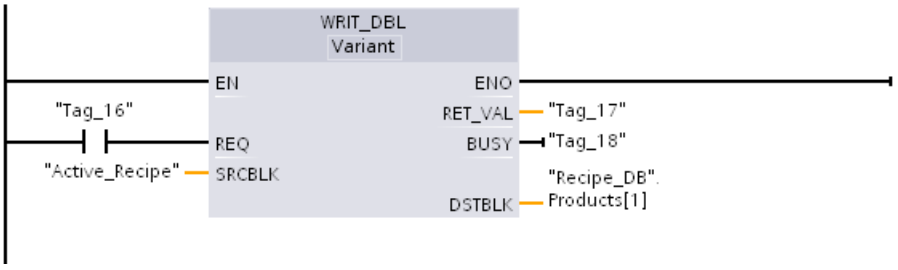


Segmento 5 READ_DBL copia los valores de arranque de una receta "Recipe_DB". Productos[1] (en la memoria de carga de la CPU) en los valores actuales del DB Active_Recipe (en la memoria de trabajo de la CPU). Después de la ejecución de READ_DBL, la lógica del programa puede

acceder a los valores de los componentes de la receta direccionando las ubicaciones del DB Active_Recipe. Por ejemplo, las direcciones simbólicas ("Active_Recipe".productname) y ("Active_Recipe.water) proporcionan a su lógica de programa el nombre de la receta actual y la cantidad de agua.



Segmento 6 Durante el tiempo de ejecución, un dispositivo HMI puede modificar un valor de componente almacenado en DB Active_Recipe. Los datos mejorados de la receta pueden almacenarse ejecutando WRIT_DBL. En el ejemplo, todos los valores de arranque de Recipe_DB para la receta individual "Recipe_DB". Los productos[1] se sobrescriben con los valores actuales del DB "Active_Recipe".



9.9.2 Registros

El programa de control puede usar las instrucciones Data log para almacenar valores de datos de runtime en archivos de registro permanentes. La CPU guarda los archivos de registro en la memoria flash (CPU o Memory Card) en formato estándar CSV (Comma Separated Value). La CPU organiza los registros como un archivo de registro circular con un tamaño predeterminado.

Las instrucciones Data log del programa se utilizan para crear, abrir, escribir un registro y cerrar los archivos de registro. Para decidir qué valores de programa deben registrarse se crea un búfer de datos que define un solo registro. La CPU utiliza el búfer de datos como almacenamiento temporal para un nuevo registro. El programa de control mueve los valores actuales nuevos al búfer durante la ejecución. Una vez el programa ha actualizado todos los valores de datos actuales, puede ejecutar la instrucción DataLogWrite para transferir datos del búfer a un registro de datos.

Los archivos de registro se pueden abrir, editar, renombrar y borrar desde la página del explorador de archivos del servidor web. Se deben tener derechos de lectura para ver el explorador de archivos y derechos de modificación para editar, borrar o cambiar el nombre de los archivos de registro.

9.9.2.1 Estructura del registro

Los parámetros DATA y HEADER de la instrucción DataLogCreate asignan el tipo de datos y la descripción de encabezado de columna para todos los datos de un registro.

Parámetro DATA para la instrucción DataLogCreate

El parámetro DATA apunta a la memoria utilizada como búfer temporal para un nuevo registro y debe asignarse a una ubicación M o DB.

Es posible asignar un DB completo (derivado de un tipo de datos PLC que se asigna al crear el DB) o parte de un DB (el elemento especificado del DB puede ser cualquier tipo de datos, estructura de tipo de datos, tipo de datos PLC o matriz de datos).

La instrucción DataLogCreate limita los tipos de datos de la estructura a un solo nivel de anidamiento. En este contexto, un ARRAY of STRINGs no se considera un nivel de anidamiento. Actualmente, la instrucción DataLogCreate no devuelve un error. La instrucción solo procesa la primera cadena del ARRAY of STRINGs. El número total de elementos de datos declarados debe coincidir con el número de columnas especificado en el parámetro de encabezado. El número máximo de elementos de datos que pueden asignarse es de 253 (con sello de tiempo) o de 255 (sin sello de tiempo). Esta restricción mantiene el registro dentro del límite de 256 columnas de una hoja de Excel.

El parámetro DATA puede asignar elementos de datos remanentes o no remanentes en un tipo de DB "estándar" (compatible con S7-300/400) u "optimizado".

Para escribir un registro, primero hay que cargar el registro DATA temporal con nuevos valores de proceso y, a continuación, ejecutar la instrucción DataLogWrite, que guarda nuevos valores de registro en el archivo de registro.

Parámetro HEADER de la instrucción DataLogCreate

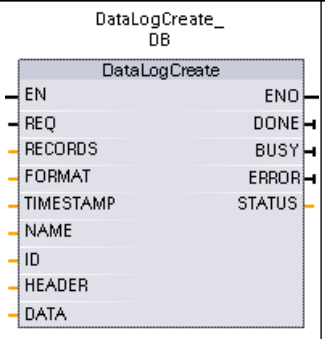
El parámetro HEADER apunta a nombres de encabezado de columna para la fila superior de la matriz de datos encriptada en el archivo CSV. Los datos HEADER deben estar ubicados en una memoria DB o M y los caracteres deben cumplir las reglas estándar del formato CSV, con comas como carácter de separación entre columnas. El tipo de datos puede ser una secuencia, una matriz de bytes o una matriz de caracteres. Las matrices de caracteres/bytes permiten un tamaño mayor, mientras que las secuencias están limitadas a un máximo de 255 bytes. El parámetro HEADER es opcional. Si HEADER no está asignado, no se crea ninguna fila de encabezado en el archivo de registro.

9.9.2.2 Instrucciones de programa que controlan registros

DataLogCreate (crear Data Log)

Tabla 9-201 Instrucción DataLogCreate

KOP / FUP	SCL	Descripción

	`"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_);`
		Crea e inicializa un archivo de registro. La CPU crea el archivo en la carpeta \DataLogs con el nombre definido en el parámetro NAME y abre implícitamente el archivo para operaciones de escritura. Las instrucciones Data log pueden utilizarse en el programa para almacenar en tiempo de ejecución datos de proceso en la memoria flash de la CPU o en la Memory Card.
		STEP 7 crea automáticamente el DB de instancia asociado al introducir la instrucción.

[1] En el ejemplo SCL, "DataLogCreate_DB" es el nombre del DB de instancia.

Tabla 9-202 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Valor predeterminado: False)
RECORDS	IN	UDint	Número máximo de registros que puede contener el registro circular antes de sobrescribir la entrada más antigua: No se incluye el registro de encabezado. Debe haber suficiente memoria de carga disponible en el PLC para poder crear el registro correctamente. (Valor predeterminado: 1)
FORMAT	IN	UInt	Formato del registro: • 0 - Formato interno (no soportado) • 1 - Valores separados por coma "csv-eng" (valor predeterminado)
TIMESTAMP	IN	UInt	Formato del sello horario: Los encabezados de columna de los campos de fecha y hora son opcionales. El sello de tiempo puede utilizar la hora del sistema (tiempo universal coordinado, UTC) o la hora local. • 0 - Sin sello de tiempo • 1 - hora del sistema mm/dd/aaaa,hh:mm:ss • 2 - hora local mm/dd/aaaa,hh:mm:ss • 3 - hora del sistema mm/dd/aaaa,hh:mm:ss • 4 - hora local aaaa-mm-dd,hh:mm:ss • 5 - hora del sistema aaaa-mm-dd,hh:mm:ss

Parámetro y tipo		Tipo de datos	Descripción
NAME	IN	Variant	Nombre del registro: El nombre lo especifica el usuario. Esta variante solo soporta un tipo de datos String y debe estar ubicada en un DB o una memoria local. (Valor predeterminado: ' ') La referencia del string es el nombre del archivo de registro. Utilice los caracteres del juego de caracteres ASCII, exceptuando los caracteres \ / : * ? " < > | y el espacio.
ID	In/Out	DWord	Identificador numérico del registro: Este valor generado se guarda para el uso en otras instrucciones de registros. El parámetro ID solo se utiliza como salida en la instrucción DataLogCreate. (Valor predeterminado: 0) El acceso a este parámetro por nombre simbólico no está permitido.
HEADER	In/Out	Variant	Puntero a nombres de encabezados de columnas de registros para la fila superior de la matriz de datos cifrada en el archivo CSV. (Valor predeterminado: cero). Los datos HEADER deben estar ubicados en una memoria DB o M. Los caracteres deben cumplir las reglas del formato CSV estándar, con comas como carácter de separación entre los nombres de las columnas. El tipo de datos puede ser una secuencia, una matriz de bytes o una matriz de caracteres. Las matrices de caracteres/bytes permiten un tamaño mayor, mientras que las secuencias están limitadas a un máximo de 255 bytes. El parámetro HEADER es opcional. Si HEADER no está parametrizado, no se crea ninguna fila de encabezado en el archivo de registro.
DATA	In/Out	Variant	Puntero a la estructura de registro, tipo definido por el usuario (UDT) o matriz. Los datos de registro deben estar ubicados en una memoria DB o M. El parámetro DATA especifica los elementos de datos individuales (columnas) de un registro, así como su tipo. La instrucción DataLogCreate limita los tipos de datos de la estructura a un solo nivel de anidamiento. En este contexto, un ARRAY of STRINGs no se considera un nivel de anidamiento. Actualmente, la instrucción DataLogCreate no devuelve un error. La instrucción solo procesa la primera cadena del ARRAY of STRINGs. El número de elementos de datos declarados debe coincidir con el número de columnas especificado en el parámetro de encabezado. El número máximo de elementos de datos que pueden asignarse es de 253 (con sello de tiempo) o de 255 (sin sello de tiempo). Esta restricción mantiene el registro dentro del límite de 256 columnas de una hoja de Excel.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última orden sin errores. (Valor predeterminado: False)
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última orden con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

La CPU crea un archivo de registro con un tamaño fijo predeterminado basado en los parámetros RECORDS y DATA y organiza los registros como un archivo de registro circular. La instrucción DataLogCreate asigna memoria remanente de la CPU para todo el registro de datos cuando la instrucción devuelve DONE = TRUE. La memoria requerida en la CPU es mayor que el tamaño del archivo debido a la administración del sistema de archivos y de los valores relacionados. La memoria permanente para el registro de datos permanece asignada hasta que la CPU anula la asignación de una de las siguientes maneras:

- El programa de usuario llama la instrucción DataLogDelete
- Un usuario del servidor web borra el registro de datos del servidor web
- Un usuario de la SIMATIC Automation Tool borra el registro de datos de la SIMATIC Automation Tool

Otra manera de borrar el archivo de registro, como p. ej. con un lector de tarjetas, no anula la asignación de la memoria permanente de la CPU para el registro.

La instrucción DataLogWrite agrega registros nuevos al archivo de registro hasta que este contenga el número máximo de registros especificado en el parámetro RECORDS. El siguiente registro que se escriba sobrescribirá el registro más antiguo. Una operación DataLogWrite adicional sobrescribirá el siguiente registro más antiguo, y así sucesivamente.

Uso de los recursos de memoria:

- Los registros solo consumen memoria de carga.
- El tamaño de todos los registros combinados está limitado por los recursos de memoria de carga disponibles. Solo se pueden abrir ocho registros a la vez. Los registros de datos se administran desde la página web estándar del File Browser (Página 889). En la descripción de esta página web estándar encontrará información acerca de cuántos registros de datos pueden administrarse al mismo tiempo.
- El número máximo posible para el parámetro RECORDS es el límite correspondiente a un número UDint (4.294.967.295). El límite actual del parámetro RECORD depende del tamaño de un solo registro, el tamaño de otros registros y los recursos de memoria de carga disponibles. Además, Excel limita el número de filas permitidas en una hoja.

Nota

La creación del registro debe haber finalizado antes de empezar una operación de escritura de registro.

- Las operaciones de creación de archivos de registro DataLogCreate y DataLogNewFile se extienden a lo largo de varios ciclos. El tiempo real requerido para la creación de un archivo de registro depende de la estructura y del número de registros. La lógica del programa debe observar y detectar la transición del bit DONE al estado TRUE, que indica que ha finalizado la creación del archivo de registro. Si el programa de usuario ejecuta una instrucción DataLogWrite antes de finalizar una operación de creación de registro, la operación de escritura no podrá escribir un registro nuevo tal como se ha previsto.
- En determinadas situaciones, cuando se está ejecutando un ciclo muy rápido del programa, la creación de un registro puede requerir más tiempo. Si el tiempo de creación es demasiado lento, hay que asegurarse de que la casilla de verificación Habilitar tiempo de ciclo mínimo para OBs cíclicos está activada, y de que el tiempo de ciclo mínimo se ha ajustado a un ms o más. Para más información, consulte Configuración del tiempo de ciclo y de la carga de comunicación (Página 91).

Nota

La instrucción DataLogNewFile copia una estructura de registro existente.

Para impedir la sobrescritura de registros antiguos se puede utilizar la instrucción DataLogNewFile para crear un registro nuevo basado en el actual, una vez que este ha guardado el número máximo de registros. Los nuevos registros se guardan en el nuevo archivo de registro. El archivo de registro antiguo y los datos de registro se almacenan en la memoria flash.

Tabla 9-203 Valores de ERROR y STATUS

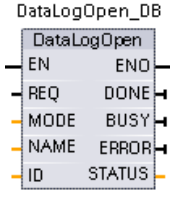
ERROR	STATUS (W#16#....)	Descripción
0	0000	No hay error
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia interna.
1	807F	Error interno
1	8090	Nombre de archivo no válido
1	8091	El parámetro de nombre no es una referencia de String.
1	8093	Ya existe un registro con ese nombre. Utilice otro nombre, asegúrese de que el archivo .csv existente para el registro no está abierto y utilice la página File Browser (Página 889) del servidor web para borrar el registro existente.
1	8097	La longitud de archivo solicitada supera el máximo del sistema de archivos.
1	80B2	ID fuera de recurso Nota: borre algunos registros o reduzca el número de columnas en la estructura de registros con el fin de evitar este error.
1	80B3	La memoria de carga disponible no es suficiente.
1	80B4	La MC (Memory Card) está protegida contra escritura.

ERROR	STATUS (W#16#....)	Descripción
1	80C0	El archivo comprimido está bloqueado
1	80C1	Demasiados archivos abiertos: no se permiten más de ocho archivos de registro abiertos.
1	8253	Contaje de registros no válido
1	8353	La selección de formato no es válida
1	8453	La selección de sello no es válida
1	8B24	Asignación no válida del área HEADER: por ejemplo, apuntar a la memoria local
1	8B51	El tipo de datos del parámetro HEADER no es válido
1	8B52	Hay demasiados elementos de datos en el parámetro HEADER
1	8C24	Asignación no válida del área DATA: por ejemplo, apuntar a la memoria local
1	8C51	El tipo de datos del parámetro DATA no es válido
1	8C52	Hay demasiados elementos de datos en el parámetro DATA

DataLogOpen (abrir Data Log)

Tabla 9-204 Instrucción DataLogOpen

KOP / FUP	SCL	Descripción

	`"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_);`
		Abre un archivo de registro ya existente. Es necesario abrir un registro antes de poder escribir (Página 497) registros nuevos en él. Los registros pueden abrirse y cerrarse individualmente. Es posible abrir simultáneamente un máximo de ocho registros de datos.
		STEP 7 crea automáticamente el DB de instancia asociado al introducir la instrucción.

[2] En el ejemplo SCL, "DataLogOpen_DB" es el nombre del DB de instancia.

Tabla 9-205 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Valor predeterminado: False)
MODE	IN	UInt	Modo de operación: • 0 - Agregar a los datos existentes (valor predeterminado) • 1 - Borrar todos los registros existentes
NAME	IN	Variant	Nombre de un registro existente: esta variante solo soporta un tipo de datos String y solo puede ubicarse en una memoria local, DB o M. (Valor predeterminado: ' ')
ID	In/Out	DWord	Identificador numérico de un registro. (Valor predeterminado: 0) **Nota:** el acceso a este parámetro por nombre simbólico no está permitido.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores. (Valor predeterminado: False)

Parámetro y tipo		Tipo de datos	Descripción
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

Se puede proporcionar el parámetro NAME o un parámetro ID (ID en forma de entrada) de un registro ya existente. Si se suministran ambos parámetros y una ID válida se corresponde con en registro NAME, se usa la ID y NAME se ignora.

NAME debe ser el nombre de un registro creado por la instrucción DataLogCreate. Si solo se proporciona NAME y NAME especifica un registro válido, se devolverá la ID correspondiente (parámetro ID como salida).

Nota

Uso general de los archivos de registro

- Los archivos de registro se abren automáticamente después de las operaciones DataLogCreate y DataLogNewFile.
- Los archivos de registro se cierran automáticamente después de una transición de RUN a STOP del PLC o de una desconexión y conexión del PLC.
- Debe haber un archivo de registro abierto antes de que sea posible otra operación DataLogWrite.
- Puede haber un máximo de ocho archivos de registro abiertos a la vez. Si bien pueden existir más de ocho archivos de registro, algunos de ellos deben estar cerrados para que no haya más de ocho abiertos.

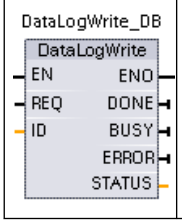
Tabla 9-206 Valores de ERROR y STATUS

ERROR	STATUS (W#16#)	Descripción
0	0000	No hay error
0	0002	Advertencia: el archivo de registro ya está abierto en este programa de aplicación
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia interna.
1	8090	La definición del registro no es coherente con el archivo de registro existente.
1	8091	El parámetro de nombre no es una referencia de String.
1	8092	El registro no existe.
1	80C0	El archivo de registro está bloqueado.
1	80C1	Demasiados archivos abiertos: no se permiten más de ocho archivos de registro abiertos.

DataLogWrite (escribir Data Log)

Tabla 9-207 Instrucción DataLogWrite

KOP / FUP	SCL	Descripción

	`"DataLogWrite_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`ID:=_dword_inout_);`
		Escribe un registro en el registro especificado. El registro de destino previo debe estar abierto (Página 495) antes de poder escribir en él con una instrucción DataLogWrite.
		STEP 7 crea automáticamente el DB de instancia asociado al introducir la instrucción.

[2] En el ejemplo SCL, "DataLogWrite_DB" es el nombre del DB de instancia.

Tabla 9-208 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Valor predeterminado: False)
ID	In/Out	DWord	Identificador numérico de registro. Solo se utiliza como entrada para la instrucción DataLogWrite. (Valor predeterminado: 0) **Nota:** el acceso a este parámetro por nombre simbólico no está permitido.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

El parámetro DATA de una instrucción DataLogCreate define la dirección de memoria y la estructura de datos del búfer de registro. El programa de control debe cargar el búfer de registro con valores de proceso actuales de runtime y, a continuación, ejecutar la instrucción DataLogWrite para desplazar los nuevos datos de registro del búfer al registro.

El parámetro ID identifica un registro y una configuración de registro. La instrucción DataLogCreate genera el número de ID.

Si hay registros vacíos en el archivo de registro circular, la instrucción DataLogWrite escribe el siguiente registro vacío disponible. Si todos los registros están llenos, la instrucción DataLogWrite sobrescribe el registro más antiguo.

ATENCIÓN
Las operaciones de creación de registro deben finalizarse antes empezar una operación de escritura de registro. Las operaciones de creación de archivos de registro DataLogCreate y DataLogNewFile se extienden a lo largo de varios ciclos. El tiempo real requerido para la creación de un archivo de registro depende de la estructura y del número de registros. La lógica de programa debe observar y capturar la transición del bit DONE al estado TRUE, que indica que ha finalizado la creación del archivo de registro. Si se ejecuta una instrucción DataLogWrite antes de finalizar una operación de creación de registro, la operación de escritura no podrá escribir un registro nuevo.

Nota

Efecto de los registros en la memoria interna de la CPU

Cada escritura en el registro consume como mínimo 2 KB de memoria. Si el programa escribe pequeñas cantidades de datos con frecuencia, consume al menos 2 KB de memoria en cada escritura. Sería mejor acumular los pequeños elementos de datos en un bloque de datos (DB) y escribir el bloque de datos en el registro en intervalos menos frecuentes.

Si el programa escribe muchas entradas en el registro con mucha frecuencia, es aconsejable utilizar una tarjeta de memoria SD extraíble.

ATENCIÓN
Potencial de pérdida de datos de registros durante un fallo de alimentación de la CPU Si se produce un fallo de alimentación durante una operación DataLogWrite incompleta, se podría perder el registro que se estuviera transfiriendo al registro.

Tabla 9-209 Valores de ERROR y STATUS

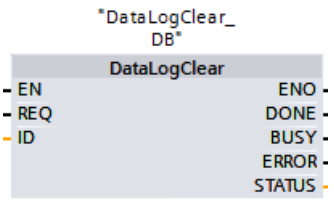
ERROR	STATUS (W#16#)	Descripción
0	0000	No hay error
0	0001	Indica que el registro está lleno: Cada registro se crea con un número máximo específico de registros. Se ha escrito el último registro del número máximo. La siguiente tarea de escritura sobrescribirá el registro más antiguo.
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia interna.
1	8092	El registro no existe.
1	80B0	El archivo de registro no está abierto (solo para modo de abertura explícita).

DataLogClear (vaciar Data Log)

Descripción

Tabla 9-210 Instrucción DataLogClear

KOP / FUP	SCL	Descripción

	`"DataLogClear_DB"(`
	`REQ:=_bool_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`ID:=_dword_inout_);`
		La instrucción "DataLogClear" borra todos los registros de un archivo de registro existente. La instrucción no borra el encabezado opcional del archivo CSV (consulte la descripción del parámetro HEADER de la instrucción "DataLogCreate (Página 491)").
		Utilice el parámetro ID para seleccionar el archivo cuyos registros deben borrarse.

"DataLogClear_DB" es el nombre del DB de instancia.

Requisitos

Antes de poder borrar registros, el archivo de registro debe estar abierto. Use la instrucción DataLogOpen (Página 495) para abrir un registro de datos.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "DataLogClear":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, L, D, T, C o constante (T y C solo están disponibles en KOP y FUP con S7-1500)	Ejecución de la instrucción con un flanco ascendente.
ID	InOut	DWORD	I, Q, M, D, L	Identificador numérico de registro
DONE	Output	BOOL	I, Q, M, D, L	La instrucción se ha ejecutado correctamente.
BUSY	Output	BOOL	I, Q, M, D, L	Ejecución de la instrucción no terminada aún.
ERROR	Output	BOOL	I, Q, M, D, L	• 0: Sin errores. • 1: Se ha producido un error al ejecutar la instrucción. La información detallada se genera en el parámetro STATUS.
STATUS	Output	WORD	I, Q, M, D, L	Parámetro de estado El parámetro solo está activado mientras dura una llamada. Para mostrar el estado debe copiar el parámetro STATUS en un área de datos libre.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

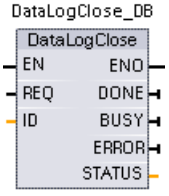
Parámetro STATUS

Código de error* (W#16#...)	Explicación
0000	Sin errores.
7000	No se está procesando ninguna tarea.
7001	Comienza el procesamiento de la tarea. Parámetro BUSY = 1, DONE = 0
7002	Llamada intermedia (REQ irrelevante): Instrucción ya activa; BUSY tiene el valor "1".
8080	El archivo de registro seleccionado con el parámetro ID no puede procesarse con la instrucción "DataLogClear".
8092	El registro no existe.
80A2	Error de escritura notificado por el sistema de archivos.
80B0	El archivo de registro no está abierto.
80B4	La Memory Card está protegida contra escritura.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".	

DataLogClose (cerrar Data Log)

Tabla 9-211 Instrucción DataLogClose

KOP / FUP	SCL	Descripción

	`"DataLogClose_DB"(`
	`req:=_bool_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`ID:=_dword_inout_);`
		Cierra un archivo de registro abierto. Las operaciones DataLogWrite sobre un registro cerrado producen un error. No se permite realizar operaciones de escritura en este registro hasta que no se lleve a cabo otra operación DataLogOpen.
		La transición al modo STOP cierra todos los archivos de registro abiertos.
		STEP 7 crea automáticamente el DB de instancia asociado al introducir la instrucción.

[2] En el ejemplo SCL, "DataLogClose_DB" es el nombre del DB de instancia.

Tabla 9-212 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Valor predeterminado: False)
ID	In/Out	DWord	Identificador numérico de un registro. Solo se utiliza como entrada para la instrucción DataLogClose. (Valor predeterminado: 0) **Nota:** el acceso a este parámetro por nombre simbólico no está permitido.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

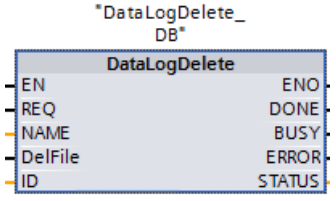
Tabla 9-213 Valores de ERROR y STATUS

ERROR	STATUS (W#16#)	Descripción
0	0000	No hay error
0	0001	El registro no se ha abierto
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8092	El registro no existe.

DataLogDelete (borrar Data Log)

Tabla 9-214 Instrucción DataLogDelete

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción "DataLogDelete" se utiliza para borrar un archivo de registro. El archivo y los registros que contiene solo pueden borrarse si se crearon con la instrucción "DataLogCreate" o "DataLogNewFile".

"DataLogDelete_DB" es el nombre del DB de instancia.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "DataLogDelete":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, L, D, T, C o constante (T y C solo están disponibles en KOP y FUP con S7-1500)	Ejecución de la instrucción con un flanco ascendente.
NAME	Input	VARIANT	L, D	Nombre del archivo de registro
DELFILE	Input	BOOL	I, Q, M, D, L o constante	• 0: El archivo de registro se conserva. • 1: El archivo de registro se borra.
ID	InOut	DWORD	I, Q, M, D, L	Identificador numérico de registro

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
DONE	Output	BOOL	I, Q, M, D, L	La instrucción se ha ejecutado correctamente.
BUSY	Output	BOOL	I, Q, M, D, L	El borrado del archivo de registro no ha finalizado aún.
ERROR	Output	BOOL	I, Q, M, D, L	• 0: Sin errores. • 1: Se ha producido un error al ejecutar la instrucción. La información detallada se genera en el parámetro STATUS.
STATUS	Output	WORD	I, Q, M, D, L	Parámetro de estado El parámetro solo está activado mientras dura una llamada. Para mostrar el estado debe copiar el parámetro STATUS en un área de datos libre.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Parámetros NAME e ID

Seleccione el registro que debe borrarse mediante los parámetros NAME e ID. El parámetro ID se evalúa en primer lugar. Si hay un registro con la ID relevante, el parámetro NAME no se evaluará. Si el valor "0" se utiliza en el parámetro ID, debe utilizarse un valor con el tipo de datos STRING en el parámetro NAME.

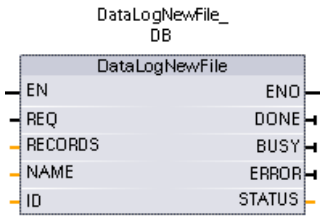
Parámetro RET_VAL

Código de error* (W#16#...)	Explicación
0	Sin errores.
7000	No se está procesando ninguna tarea.
7001	Comienza el procesamiento de la tarea. Parámetro BUSY = 1, DONE = 0
7002	Llamada intermedia (REQ irrelevante): Instrucción ya activa; BUSY tiene el valor "1".
8091	Se está utilizando un tipo de datos distinto de STRING en el parámetro NAME.
8092	El registro no existe.
80A2	Error de escritura notificado por el sistema de archivos.
80B4	La Memory Card está protegida contra escritura.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".	

DataLogNewFile (Data Log en archivo nuevo)

Tabla 9-215 Instrucción DataLogNewFile

KOP / FUP	SCL	Descripción

	`"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_);`
		Permite al programa crear un archivo de registro nuevo basándose en otro ya existente.
		STEP 7 crea automáticamente el DB de instancia asociado al introducir la instrucción.

[2] En el ejemplo SCL, "DataLogNewFile_DB" es el nombre del DB de instancia.

Tabla 9-216 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Valor predeterminado: False)
RECORDS	IN	UDInt	El número máximo de registros que puede contener el registro circular antes de sobrescribir la entrada más antigua. (Valor predeterminado: 1) No se incluye el registro de encabezado. Debe haber suficiente memoria de carga disponible en la CPU para poder crear el registro correctamente.
NAME	IN	Variant	Nombre del registro: El nombre lo especifica el usuario. Esta variante solo soporta un tipo de datos String y solo puede ubicarse en una memoria local, DB o M. (Valor predeterminado: ' ') La referencia del string también se utiliza como nombre del archivo de registro. Los caracteres del nombre deben cumplir las restricciones referentes a los nombres del sistema de archivos de Windows. Los caracteres \ / : * ? " < > | y el espacio no están permitidos.
ID	In/Out	DWord	Identificador numérico de registro (valor predeterminado: 0): • Durante la ejecución, la entrada ID identifica un registro válido. La nueva configuración de registro se copia desde este registro. • Tras la ejecución, el parámetro ID se convierte en una salida que devuelve la ID del archivo de registro recién creado. **Nota:** el acceso a este parámetro por nombre simbólico no está permitido.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1 - Operación en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

La instrucción DataLogNewFile se puede ejecutar cuando un registro se llena o se considera que está completo y no se desea perder ningún dato almacenado en él. Es posible crear un archivo

de registro vacío basándose en la estructura del archivo de registro lleno. El registro del encabezado se duplica a partir del registro original, junto con las propiedades de este último (búfer de registro DATA, formato de datos y configuración de sello de tiempo). El archivo de registro original se cierra implícitamente y el archivo de registro nuevo se abre también implícitamente.

Disparador del parámetro DataLogWrite: El programa de usuario debe controlar los parámetros ERROR y STATUS de cada operación DataLogWrite. Cuando se escribe el último registro y el archivo queda lleno, el bit ERROR de DataLogWrite = 1 y la palabra STATUS de DataLogWrite = 1. Los valores de ERROR y STATUS son válidos únicamente durante un ciclo, de modo que la lógica de vigilancia debe utilizar ERROR = 1 como puerta temporal para capturar el valor de STATUS y comprobar que STATUS = 1 (el archivo de registro está lleno).

Operación DataLogNewFile: Cuando la lógica del programa de usuario recibe la señal de que el registro está lleno, este estado se utiliza para activar una operación DataLogNewFile. DataLogNewFile debe ejecutarse con la ID de un registro ya existente (generalmente lleno) y abierto, pero con un solo parámetro NAME nuevo. Una vez que la operación DataLogNewFile se ha realizado, se devuelve un nuevo valor de ID de registro (como parámetro de salida), que corresponde al nombre del nuevo registro. El nuevo archivo de registro está implícitamente abierto, y está listo para almacenar nuevos registros. Las operaciones DataLogWrite nuevas directamente relacionadas con el nuevo archivo de registro deben utilizar el valor ID devuelto por la operación DataLogNewFile.

ATENCIÓN

Las operaciones de creación de registro deben finalizarse antes empezar una operación de escritura de registro.

Las operaciones de creación de archivos de registro DataLogCreate y DataLogNewFile se extienden a lo largo de varios ciclos. El tiempo real requerido para la creación de un archivo de registro depende de la estructura y del número de registros. La lógica de programa debe observar y capturar la transición del bit DONE al estado TRUE, que indica que ha finalizado la creación del archivo de registro. Si se ejecuta una instrucción DataLogWrite antes de finalizar una operación de creación de registro, la operación de escritura no podrá escribir un registro nuevo tal como se ha previsto.

Tabla 9-217 Valores de ERROR y STATUS

ERROR	STATUS (W#16#)	Descripción
0	0000	No hay error
0	7000	Llamada sin flanco REQ: BUSY = 0, DONE = 0
0	7001	Primera llamada con flanco de REQ (en curso): BUSY = 1, DONE = 0
0	7002	Llamada N° (en curso): BUSY = 1, DONE = 0
1	8070	Se está utilizando toda la memoria de instancia interna.
1	8090	Nombre de archivo no válido
1	8091	El parámetro de nombre no es una referencia de String.
1	8092	El registro no existe.
1	8093	El registro ya existe.
1	8097	La longitud de archivo solicitada supera el máximo del sistema de archivos.

ERROR	STATUS (W#16#)	Descripción
1	80B2	ID fuera de recurso Nota: borre algunos archivos de registro existentes para generar recursos para un archivo de registro nuevo.
1	80B3	La memoria de carga disponible no es suficiente.
1	80B4	El MC está protegido contra escritura.
1	80C1	Demasiados archivos abiertos.

9.9.2.3 Trabajar con registros de datos

Los archivos de registro se guardan en formato de valores separados por comas (*.csv) en una memoria flash remanente. Los archivos de registro pueden verse utilizando la función de servidor web del PLC o retirando la Memory Card del PLC e insertándola en un lector de tarjetas de PC convencional.

Leer registros con la función de servidor web del PLC

Si el puerto PROFINET del PLC y el equipo están conectados a una red, se puede emplear un navegador de PC (como Microsoft Internet Explorer o Mozilla Firefox) para acceder al servidor web integrado. El PLC puede estar en modo RUN o STOP cuando se utilice el servidor web. Si el PLC está en modo RUN, el programa de control sigue funcionando mientras el servidor web del PLC transfiere los datos de registro a través de la red.

Acceso al servidor web:

1. Habilite el servidor web en la configuración de dispositivos para la CPU de destino (Página 846).
2. Conecte el PC al PLC a través de la red PROFINET (Página 851).
3. Acceda a la CPU mediante un servidor web integrado (Página 857).
4. Cargue, edite y borre los archivos de registro con la página web "File Browser" (Página 889) estándar.
5. Abra el archivo .csv con una aplicación para hojas de cálculo como Microsoft Excel.

Nota

Administración de registros

No guarde más de 1000 registros en un sistema de archivos. Si se excede este número, existe la posibilidad de que el servidor web no tenga recursos suficientes en la CPU para mostrar los registros.

Si resulta que la página web del explorador de archivos no puede visualizar los registros, deberá poner la CPU en estado STOP para ver y borrar registros.

Administre los registros para garantizar que solo se guardan aquellos que deben mantenerse y no exceda el número de 1000 registros.

Ver registros de una Memory Card del PLC

Si la CPU S7-1200 tiene insertada una Memory Card "Programa" del tipo S7-1200, se puede extraer e insertar en una ranura SD (Secure Digital) o MMC (MultiMediaCard) estándar de un PC o una programadora. El PLC se encuentra en modo STOP cuando no tiene insertada la Memory Card, por lo que el programa de control no se ejecutará.

Utilice el explorador de Windows y vaya al directorio \DataLog de la Memory Card. Todos los archivos de registro *.csv se encuentran en ese directorio.

Haga una copia de los archivos de registro y guarde las copias en una unidad local del PC. A continuación se puede utilizar Excel para abrir una copia local de un archivo *.csv, pero no el archivo original, que permanece guardado en la Memory Card.

ATENCIÓN
Con ayuda de un lector de tarjetas para PC es posible copiar los archivos de registro de una Memory Card S7-1200, pero no pueden modificarse ni borrarse La página File Browser estándar del servidor web es la herramienta recomendada para visualizar, cargar (copiar) y borrar archivos de registro. En la navegación directa en el sistema de archivos de la Memory Card con Windows Explorer existe el riesgo de borrar o modificar accidentalmente registros u otros archivos de sistema, lo que podría estropear un archivo o hacer que la Memory Card sea inservible.

ATENCIÓN
Efecto de los registros en Memory Cards Para garantizar el rendimiento general y la robustez del sistema, limite la frecuencia de registro como máximo a 200 ms.

9.9.2.4 Límite del tamaño de los archivos de registro

Los archivos de registro comparten espacio de la memoria de carga del PLC con el programa, los datos del programa, los datos de configuración, las páginas web definidas por el usuario y los datos del sistema PLC. Un programa grande que utiliza la memoria de carga interna requiere una gran cantidad de memoria de carga. Podría suceder que no hubiera espacio libre suficiente para archivos de registro. En tal caso, se puede utilizar una "Tarjeta de programa" (Página 125) para aumentar el tamaño de la memoria de carga. Las CPU S7-1200 pueden utilizar memoria de carga interna o externa, pero no ambas a la vez.

Regla de tamaño máximo para archivos de registro de datos (Data Log)

El tamaño máximo de un archivo de registro no puede superar el tamaño de la memoria de carga libre o 500 MB, lo que sea más pequeño. En este caso, el tamaño de 500 MB se refiere a la definición decimal de megabytes, de modo que el tamaño máximo de un archivo de registro es de 500.000.000 bytes o 500 x 1000^2 bytes.

Tabla 9-218 Tamaño de la memoria de carga

Área de datos	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C, CPU 1217C	Almacenamiento de datos
Memoria de carga interna memoria flash	1 MB	2 MB	4 MB	4 MB	Programa de usuario y datos de programa, datos de configuración, registros de datos, páginas web definidas por el usuario, y datos del sistema PLC
Memoria de carga externa Memoria flash opcional como "Tarjeta de programa"	4 MB, 12 MB, 24 MB, 256 MB, 2 GB o 32 GB, según el tamaño de la tarjeta SD				

Cálculo del espacio libre en la memoria de carga

La cantidad de memoria de carga libre varía durante las operaciones normales, ya que el sistema operativo va utilizando y liberando memoria. Siga los pasos siguientes para ver el tamaño de la memoria de carga.

1. Establecer una conexión online entre STEP 7 y el PLC S7-1200 de destino.
2. Descargue el programa que controla las operaciones de registro de datos.
3. Cree las páginas web personalizadas que se requieran. Las páginas web estándar que permiten acceder a registros de datos se guardan en el firmware del PLC y no utilizan memoria de carga.
4. Utilice las herramientas online y de diagnóstico (Página 1197) o la página de diagnóstico del servidor web (Página 864) para ver el tamaño total de la memoria de carga y el espacio libre.

Cálculo del tamaño de un archivo de registro (todos los registros)

Cuando se crea el archivo de registro, la CPU asigna el tamaño de memoria máximo. Además del tamaño necesario para todos los registros, debe incluir espacio de almacenamiento para un encabezado de registro (si se utiliza), un encabezado de sello de tiempo (si se utiliza), un encabezado de índice de registro y el tamaño del bloque mínimo para la asignación de memoria.

Utilice la fórmula siguiente para determinar el tamaño de los archivos de registro de datos y asegúrese de no infringir la regla de tamaño máximo.

Bytes de datos de registro = ((bytes de datos en un registro + bytes de sello de tiempo + 12 bytes) * número de registros)

Encabezado

Bytes de encabezado de registro = bytes de caracteres de encabezado + 2 bytes

Bytes de caracteres de encabezado

- Sin encabezado de datos ni sellos de tiempo = 7 bytes
- Sin encabezado de datos ni sellos de tiempo (tiene un encabezado de sello de tiempo) = 21 bytes
- Encabezados de datos sin sellos de tiempo = Número de bytes de caracteres en todo el texto de encabezado de la columna, incluidas comas separadoras
- Encabezados de datos y sellos de tiempo (tiene un encabezado de sello de tiempo) = Número de bytes de caracteres en todo el texto de encabezado de la columna, incluidas comas separadoras + 21 bytes

Datos

Bytes de datos de registro = ((bytes de datos en un registro + bytes de sello de tiempo + 12 bytes) * número de registros)

Bytes de datos en un registro

El parámetro DATA de DataLogCreate apunta a una estructura que asigna el número de campos de datos y tipo de datos de cada campo para un registro.

Multiplique el número de apariciones del tipo de datos indicado por el nombre de bytes requerido. Repita el proceso por cada tipo de datos en un registro y sume todos los bytes de datos para obtener el total de elementos de datos en un registro.

Tamaño de los elementos de datos individuales

El registro se guarda en forma de bytes de caracteres en el formato de archivo CSV (valores separados por coma). La tabla siguiente muestra el número de bytes necesarios para guardar cada elemento de datos.

Tipo de datos	Número de bytes (incluye datos más un byte de coma)
Bool	2
Byte	5
Word	7
DWord	12
Char	4
String	**Ejemplo 1**: MyString String[10] El tamaño máximo de la cadena es de 10 caracteres. Caracteres de texto + relleno automático con caracteres vacíos = 10 bytes Comillas de apertura y cierre + caracteres coma = 3 bytes 10 + 3 = 13 bytes en total **Ejemplo 2**: Mystring2 String Si se omiten los corchetes al indicar el tamaño, se asignan de forma predeterminada 254 bytes. Caracteres de texto + relleno automático con caracteres vacíos = 254 bytes Comillas de apertura y cierre + caracteres coma = 3 bytes 254 + 3 = 257 bytes en total
USInt	5
UInt	7

UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Tiempo	15
DTL	24

Número de registros en un archivo de registro

El parámetro RECORDS de la instrucción DataLogCreate especifica el número máximo de registros de un archivo de registro.

Bytes de sello de tiempo en un registro

- Sin sello de tiempo = 0 bytes
- Sello de tiempo = 20 bytes

9.9.2.5 Programa de ejemplo de registros de datos

Este programa de ejemplo de registro no muestra toda la lógica del programa necesaria para obtener valores de muestreo de un proceso dinámico, pero muestra las operaciones clave de las instrucciones de registro. La estructura y el número de archivos de registro utilizados dependen de las necesidades de control del proceso.

Nota

Uso general de los archivos de registro

- Los archivos de registro se abren automáticamente después de las operaciones DataLogCreate y DataLogNewFile.
- Los archivos de registro se cierran automáticamente después de una transición de RUN a STOP del PLC o de una desconexión y conexión del PLC.
- Debe haber un archivo de registro abierto antes de que sea posible una operación DataLogWrite.
- Puede haber un máximo de ocho archivos de registro abiertos a la vez. Si bien pueden existir más de ocho archivos de registro, algunos de ellos deben estar cerrados para que no haya más de ocho abiertos.

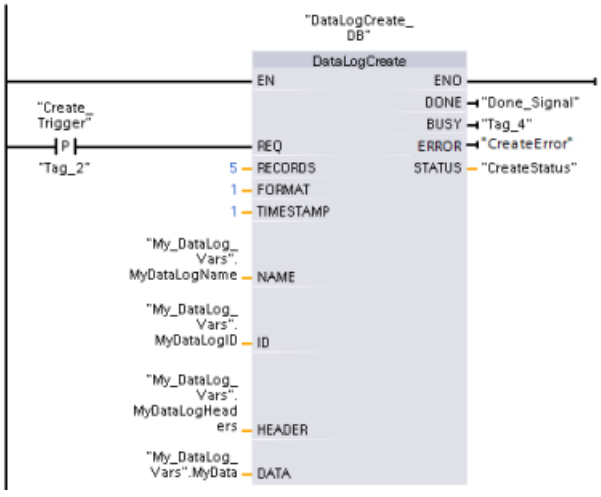
Ejemplo de programa de registro

Los nombres del registro, el texto del encabezado y la estructura MyData del ejemplo se crean en un bloque de datos. Las tres variables MyData guardan temporalmente valores de muestreo nuevos. Los valores de muestreo del proceso en las ubicaciones de estos DBs se transfieren a un archivo de registro ejecutando la instrucción DataLogWrite.

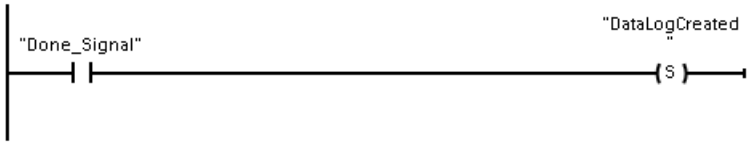
My_Datalog_Vars

	Nombre	Tipo de datos	Valor de arranque
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

Segmento 1 El flanco ascendente REQ inicia el proceso de creación de registros de datos.



Segmento 2 Capturar la salida DONE de DataLogCreate porque solo es válida durante un ciclo.



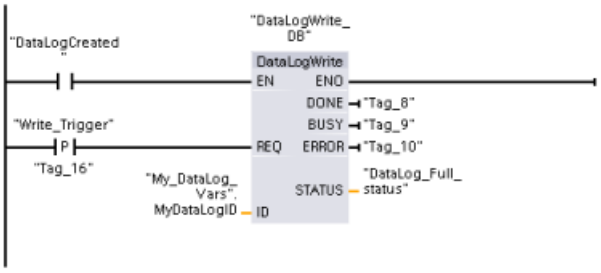
Segmento 3 Guardar la salida de estado si hay un error



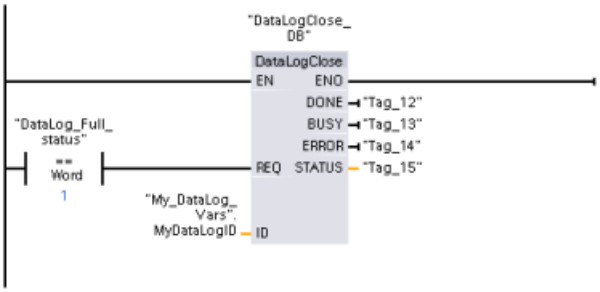
Segmento 4 Una señal de flanco ascendente se dispara cuando deben guardarse valores de proceso nuevos en la estructura MyData.



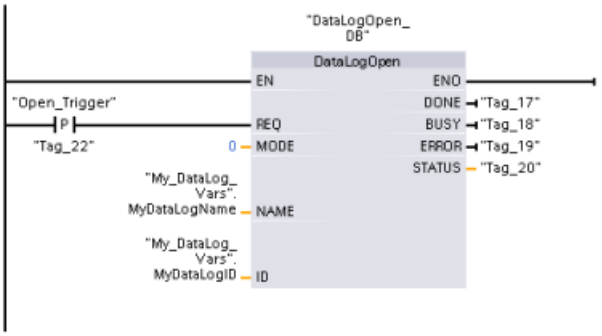
Segmento 5 El estado de la entrada EN se toma como base cuando la operación DataLogCreate ha finalizado. Una operación de creación dura muchos ciclos y debe completarse antes de ejecutar una operación de escritura. La señal de flanco ascendente en la entrada REQ es el evento que dispara una operación de escritura habilitada.



Segmento 6 Cierre el registro una vez se haya escrito el último registro. Después de ejecutar la operación DataLogWrite que escribe el último registro, el estado "lleno" del archivo de registro se indica cuando la salida de ESTADO de DataLogWrite = 1.

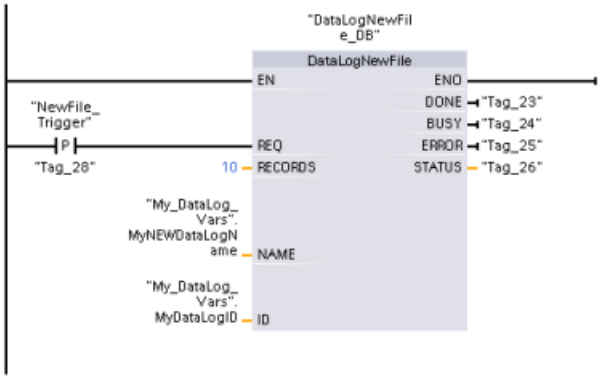


Segmento 7 Una entrada REQ DataLogOpen para la señal de flanco ascendente simula la pulsación de un botón por parte del usuario en un HMI que abre un archivo de registro. Si se abre un archivo de registro cuyos registros contienen datos de proceso, la próxima operación DataLogWrite sobrescribirá el registro más antiguo. Quizás prefiera conservar el registro antiguo y crear un registro nuevo, tal como se muestra en el segmento 7.



Segmento 8 El parámetro ID es del tipo IN/OUT. En primer lugar debe proporcionarse el valor ID del registro existente cuya estructura desee copiar. Una vez ha finalizado la operación

DataLogNewFile, se escribe un valor ID unívoco para el nuevo registro en la ubicación de referencia de la ID. No se muestra la obtención necesaria del bit DONE = TRUE; consulte los segmentos 1, 2 y 4 para ver un ejemplo de la lógica del bit DONE.



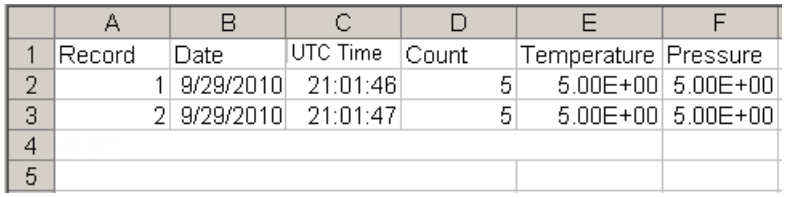
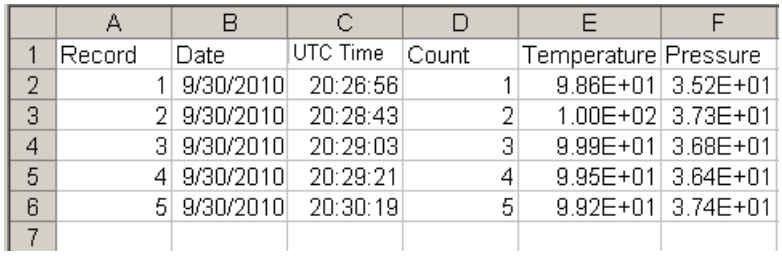
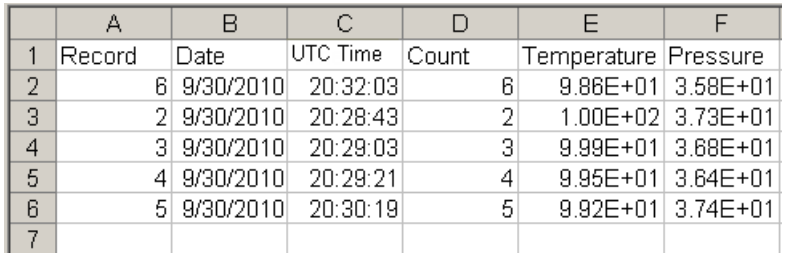
Archivos de registro creados por el programa de ejemplo vistos con el servidor web de la CPU S7-1200



① La opción "Borrar" solo está disponible si el usuario ha iniciado sesión con privilegios de modificación.

② La opción "Cambiar nombre" solo está disponible si el usuario ha iniciado sesión con privilegios de modificación.

Tabla 9-219 Ejemplos de archivos .csv descargados vistos con Excel

Dos registros escritos en un archivo de cinco registros como máximo	
Cinco registros en un archivo de registro con un máximo de cinco registros	
Después de escribir un registro adicional en el archivo anterior, que ya está lleno, la sexta operación de escritura sobrescribe el registro más antiguo con el sexto registro. Otra operación de escritura sobrescribirá el segundo registro con el séptimo, etc.	

Nota

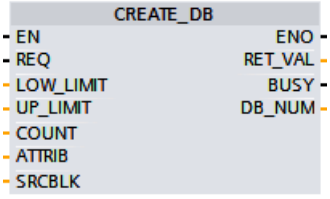
Los registros ya no usan una marca //END para señalizar el final del archivo de registro que no está lleno. En las versiones de la CPU S7-1200 anteriores a V4.1, los registros que no estaban llenos incluían una marca //END.

9.10 Control de bloques de datos

9.10.1 CREATE_DB (crear bloque de datos)

Tabla 9-220 Instrucción CREATE_DB

KOP / FUP	SCL	Descripción

	`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_);`
		Utilice la instrucción "CREATE_DB" para crear un bloque de datos nuevo en la memoria de carga y/o de trabajo.
		La instrucción "CREATE_DB" no cambia la suma de verificación del programa de usuario.
		Un bloque de datos creado únicamente en la memoria de trabajo tiene las propiedades siguientes:
		• Después de un borrado total o de una desconexión y conexión (POWER OFF/ON), el bloque deja de existir.
		• Durante la carga o cuando hay una transición de STOP a RUN, su contenido no cambia.

Número del bloque de datos

El bloque de datos creado se asigna a un número comprendido dentro del rango definido en los parámetros LOW_LIMIT (límite inferior) y UP_LIMIT (límite superior). "CREATE_DB" asigna al DB el número más bajo posible del rango especificado. No es posible asignar los números de los DB que ya están incluidos en el programa de usuario.

Para crear un DB con un número concreto, introduzca el mismo número en los límites superior e inferior del rango que debe especificarse. Si ya existe un DB con el mismo número en la memoria de trabajo y/o de carga, o bien si el DB existe como versión copiada, la instrucción finalizará y se generará un mensaje de error en el parámetro RET_VAL.

Valores iniciales del bloque de datos

El parámetro SRCBLK se utiliza para definir valores iniciales en el DB que se creará. El parámetro SRCBLK es un puntero a un DB o área de DB desde el que se aplican los valores iniciales. El DB direccionado en el parámetro SRCBLK debe haberse generado con un acceso estándar (atributo "Acceso optimizado al bloque" desactivado).

- Si el área especificada en el parámetro SRCBLK es más grande que el DB generado, se aplicarán como valores iniciales los que vayan hasta la longitud del DB generado.
- Si el área especificada en el parámetro SRCBLK es más pequeña que el DB generado, los demás valores se rellenarán con "0".

Para garantizar la coherencia de datos, no hay que cambiar esta área de datos mientras se ejecute "CREATE_DB" (es decir, mientras el parámetro BUSY tenga el valor TRUE).

Descripción del funcionamiento

La instrucción "CREATE_DB" funciona de forma asíncrona. El procesamiento se extiende a lo largo de varias llamadas. Inicie la tarea llamando "CREATE_DB" con REQ = 1.

Los parámetros de salida RET_VAL y BUSY indican el estado de la tarea.

Consulte también: DELETE_DB (borrar bloque de datos) (Página 523)

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "CREATE_DB":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L o constante	Parámetro de control disparado por nivel "request to activate" REQ = 1: Petición para generar el bloque de datos
LOW_LIMIT	Input	UINT	I, Q, M, D, L o constante	Límite inferior del rango para asignar un número de DB. El número más bajo posible es 60000.
UP_LIMIT	Input	UINT	I, Q, M, D, L o constante	Límite superior del área utilizada por "CREATE_DB" para asignar un número al DB (número más alto posible: 60999)
COUNT	Input	UDINT	I, Q, M, D, L o constante	El valor de contaje especifica el número de bytes que deben reservarse para el DB generado. El número de bytes debe ser un número par. La longitud máxima es de 65534 bytes.

<table>
<tr><th>Parámetro</th><th>Declaración</th><th>Tipo de datos</th><th>Área de memoria</th><th>Descripción</th></tr>
<tr><td rowspan="7">ATTRIB</td><td rowspan="7">Input</td><td rowspan="7">BYTE</td><td rowspan="7">I, Q, M, D, L o constante</td><td>Los 4 primeros bits del byte en el parámetro ATTRIB se utilizan para definir las propiedades del bloque de datos *:</td></tr>
<tr><td>• Bit 0 = 0: El atributo "Depositar solo en la memoria de carga" no está activado.
• Bit 0 = 1: El atributo "Depositar solo en la memoria de carga" está activado. Con este ajuste, el DB no ocupa espacio en la memoria de trabajo y no está incluido en el programa. No es posible acceder al DB con comandos de bits. Si el bit 0 = 1, la selección del bit 2 es irrelevante.
Para garantizar la compatibilidad con STEP 7 V5.x, los bits 0 y 3 deben considerarse juntos (véase más abajo).</td></tr>
<tr><td>• Bit 1 = 0: El atributo "Bloque de datos protegido contra escritura en el dispositivo" no está activado.
• Bit 1 = 1: El atributo "Bloque de datos protegido contra escritura en el dispositivo" está activado.</td></tr>
<tr><td>• Bit 2 = 0: El DB es remanente (solo para DB generados en la memoria de carga y de trabajo). El DB se considera remanente si al menos uno de los valores se ha puesto a remanente.
• Bit 2 = 1: El DB no es remanente.
La remanencia no se soporta en DB que solo están almacenados en la memoria de carga o solo en la memoria de trabajo. Si se llama la instrucción "CREATE_DB" con una de las dos combinaciones "remanente y solo memoria de carga" o "remanente y solo memoria de trabajo", el DB que debe generarse no se marcará como remanente.</td></tr>
<tr><td>• Bit 3 = 0: Creación del DB en la memoria de carga o en la memoria de trabajo (selección con el bit 0, véase más arriba)
• Bit 3 = 1: Creación del DB tanto en la memoria de carga como en la memoria de trabajo (bit 0 irrelevante)
Para garantizar la compatibilidad con STEP 7 V5.x, los bits 0 y 3 deben utilizarse combinados. Si el bit 3 = 1, el bit 0 es irrelevante.
<table>
<tr><th>Bit 0</th><th>Bit 3</th><th>Generación del DB</th></tr>
<tr><td>0</td><td>0</td><td>Solo en la memoria de trabajo</td></tr>
<tr><td>1</td><td>0</td><td>Solo en la memoria de carga</td></tr>
<tr><td>Irrelevante</td><td>1</td><td>Memoria de trabajo y de carga</td></tr>
</table></td></tr>
<tr><td>• Bit 4 = 0 - No se han especificado valores iniciales (los valores de entrada en el parámetro SRCBLK se ignorarán).
• Bit 4 = 1 - Especificar valores iniciales (los valores se corresponden con el DB direccionado por el parámetro SRCBLK).</td></tr>
<tr><td>SRCBLK</td><td>Input</td><td>VARIANT</td><td>D</td><td>Puntero al bloque de datos cuyos valores se utilizarán para inicializar el bloque de datos que se generará.</td></tr>
<tr><td>RET_VAL</td><td>Return</td><td>INT</td><td>I, Q, M, D, L</td><td>Información de error</td></tr>
<tr><td>BUSY</td><td>Output</td><td>BOOL</td><td>I, Q, M, D, L</td><td>BUSY = 1: El proceso no ha finalizado aún.</td></tr>
</table>

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
DB_NUM	Output	DB_DYN (UINT)	I, Q, M, D, L	Número del DB creado.
* Las propiedades seleccionadas aquí equivalen a los atributos de las propiedades de un bloque de datos.				

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Parámetro RET_VAL

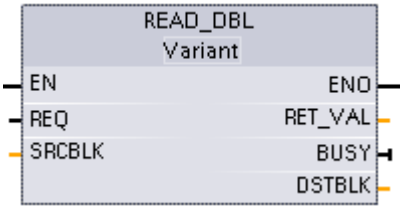
Código de error* (W#16#...)	Descripción
0000	Sin errores
0081	El área de destino es mayor que la de origen. El área de origen se escribe por completo en el área de destino. Los bytes restantes del área de destino no cambian.
7000	Primera llamada con REQ = 0: No hay ninguna transferencia de datos activa; BUSY tiene el valor "0".
7001	Primera llamada con REQ = 1: Transferencia de datos disparada; BUSY tiene el valor "1".
7002	Llamada intermedia (REQ irrelevante): Transferencia de datos ya activa; BUSY tiene el valor "1".
8081	El área de origen es mayor que la de destino. Se escribe el área de destino completa. Los bytes restantes del área de origen se ignoran.
8092	La función "Crear bloque de datos" no está disponible actualmente porque • La función "Comprimir memoria de usuario" está activa. • Se ha alcanzado el número máximo de bloques en la CPU.
8093	Para el parámetro SRCBLK no se ha especificado ningún bloque de datos o bien el bloque de datos especificado no está en la memoria de trabajo.
8094	Se ha especificado un valor no válido en el parámetro ATTRIB.
80A1	Error en el número del DB: • El número es "0" • El número excede el límite superior de la CPU para números de DB • Límite inferior > límite superior
80A2	Error en la longitud del DB: • La longitud es "0" • La longitud es un número impar • La longitud es mayor que la permitida por la CPU
80A3	El bloque de datos del parámetro SRCBLK no se ha creado con acceso estándar.
80B1	No hay ningún número de DB libre.
80B2	No hay suficiente memoria de trabajo.
80B4	La Memory Card está protegida contra escritura.
80BB	No hay suficiente memoria de carga.
80C3	Se ha alcanzado el número máximo de instrucciones "CREATE_DB" activas al mismo tiempo.

Código de error* (W#16#...)	Descripción
Información general del error	Consulte también: Códigos de error comunes para las instrucciones avanzadas (Página 531)
* El código de error puede visualizarse en el editor de programas como valor entero o hexadecimal.	

9.10.2 Instrucciones READ_DBL y WRIT_DBL (Leer de/escribir en un bloque de datos de la memoria de carga)

Tabla 9-221 Instrucciones READ_DBL y WRIT_DBL

KOP / FUP	SCL	Descripción

	`READ_DBL(`
	`req:=_bool_in_,`
	`srcblk:=_variant_in_,`
	`busy=>_bool_out_,`
	`dstblk=>_variant_out_);`
		Copia valores iniciales del DB o partes de ellos de la memoria de carga a un DB de destino en la memoria de trabajo.
		El contenido de la memoria de carga no cambia durante el proceso de copia.

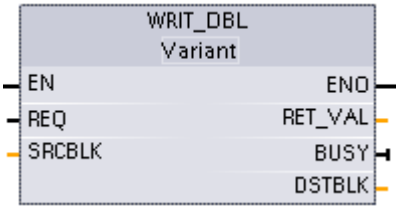
	`WRIT_DBL(`
	`req:=_bool_in_,`
	`srcblk:=_variant_in_,`
	`busy=>_bool_out_,`
	`dstblk=>_variant_out_);`
		Copia valores de DB actuales o partes de ellos de la memoria de trabajo a un DB de destino en la memoria de carga.
		El contenido de la memoria de trabajo no cambia durante el proceso de copia.

Tabla 9-222 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	BOOL	Una señal high inicia la operación si BUSY = 0.
SRCBLK	IN	VARIANT	READ_DBL: Puntero al bloque de datos fuente en la memoria de carga WRIT_DBL: Puntero al bloque de datos fuente en la memoria de trabajo
RET_VAL	OUT	INT	Código de condición de ejecución
BUSY	OUT	BOOL	BUSY = 1 indica que el proceso de lectura/escritura no se ha completado.
DSTBLK	OUT	VARIANT	READ_DBL: Puntero al bloque de datos de destino en la memoria de trabajo WRIT_DBL: Puntero al bloque de datos de destino en la memoria de carga

Típicamente, un DB se almacena tanto en memoria de carga (flash) como en memoria de trabajo (RAM). Los valores de arranque (valores iniciales) siempre se almacenan en memoria de carga, y los valores actuales siempre se almacenan en memoria de trabajo. READ_DBL se puede utilizar para copiar un conjunto de valores de arranque de la memoria de carga a los valores actuales de un DB en memoria de trabajo que se referencia mediante el programa. WRIT_DBL se puede

utilizar para actualizar los valores de arranque guardados en la memoria de carga interna o en una Memory Card a partir de los valores actuales de la memoria de trabajo.

Nota

Efecto de las instrucciones WRIT_DBL y READ_DBL en la memoria flash

La instrucción WRIT_DBL realiza operaciones de escritura en una memoria flash (memoria de carga interna o Memory Card). Para evitar que se reduzca la vida útil de la memoria flash, utilice la instrucción WRIT_DBL para actualizaciones poco frecuentes, como grabar los cambios en un proceso de producción. Por el mismo motivo, evite llamar con frecuencia READ_DBL para operaciones de lectura.

Los bloques de datos para READ_DBL y WRIT_DBL deben crearse antes de llamar estas instrucciones en el programa STEP 7. Si el DB de origen se creó como tipo "estándar", el DB de destino también debe tener el tipo "estándar". Si el bloque de datos de origen se creó como tipo "optimizado", el bloque de datos de destino también debe tener el tipo "optimizado".

Si los DB son estándar, se puede especificar un nombre de etiqueta o un valor de P#. El valor de P# permite especificar y copiar cualquier número de elementos del tamaño especificado (Byte, Word o DWord). De este modo, se puede copiar todo un DB o parte de él. Si los DB están optimizados, solamente se puede especificar un nombre de variable; no se puede utilizar el operador P#. Si se especifica un nombre de variable para los DBs estándar u optimizados (o para otros tipos de memoria de trabajo), la instrucción copia los datos a los que hace referencia este nombre de variable. Puede ser un tipo definido por el usuario, una matriz o un elemento básico. El tipo Struct solo puede utilizarse con estas instrucciones si el DB es estándar y no optimizado. Se debe utilizar un tipo definido por el usuario (UDT) si es una estructura en memoria optimizada. Solo un tipo definido por el usuario garantiza que los "tipos de datos" sean exactamente los mismos para las estructuras de origen y destino.

Nota

Uso de estructuras (tipo de datos Struct) en un DB "optimizado"

Al usar un tipo de datos Struct con DB "optimizados", primero debe crearse un tipo de datos definido por el usuario (UDT) para la Struct. Después, se configuran los DB de origen y de destino en el UDT. El UDT asegura que los tipos de datos dentro de la Struct sean coherentes para ambos DB.

Para DB "estándar", se usa la Struct sin crear un UDT.

READ_DBL y WRIT_DBL se ejecutan de forma asíncrona respecto al ciclo del programa. El procesamiento se extiende en varias llamadas de READ_DBL y WRIT_DBL. La tarea de transferencia de DB se inicia llamando con REQ = 1 y a continuación controlado las salidas BUSY

y RET_VAL para determinar cuándo se ha completado la transferencia de datos y si lo ha hecho correctamente.

Nota

Efecto de las instrucciones WRIT_DBL y READ_DBL en la carga de comunicación

Cuando la instrucción WRIT_DBL o READ_DBL está activa de forma continua, puede consumir recursos de comunicación hasta el punto que STEP 7 pierda la comunicación con la CPU. Por este motivo, utilice una entrada de flanco ascendente (Página 214) para el parámetro REQ y no una entrada normalmente abierta o cerrada (Página 209) que se mantenga activada (nivel de señal "high") durante varios ciclos.

Para garantizar la coherencia de datos, no modifique el área de destino durante el procesamiento de READ_DBL o el área de origen durante el procesamiento de WRIT_DBL (es decir, mientras el parámetro BUSY sea TRUE).

Restricciones de los parámetros SRCBLK y DSTBLK:

- Un bloque de datos debe haber sido creado antes de poder ser referenciado.
- La longitud de un puntero VARIANT del tipo BOOL debe ser divisible entre 8.
- La longitud de un puntero VARIANT del tipo STRING debe ser el mismo en los punteros fuente y de destino.

Información de recetas y de configuración de la máquina

Se pueden utilizar las instrucciones READ_DBL y WRIT_DBL para gestionar información de recetas o de configuración de la máquina. En esencia, esto constituye otro método de obtener datos remanentes para los valores que no cambian con frecuencia, aunque se desee limitar el número de escrituras para impedir que la memoria flash se desgaste prematuramente. Ello permite aumentar de forma eficaz la cantidad de memoria remanente más allá de la soportada para los datos normales remanentes tras el apagado, al menos para los valores que no cambian con frecuencia. La información de recetas o la información de configuración de la máquina de la memoria de trabajo se puede guardar en la memoria de carga mediante la instrucción WRIT_DBL, y dicha información se puede recuperar de la memoria de carga a la memoria de trabajo mediante la instrucción READ_DBL.

Tabla 9-223 Códigos de condición

RET_VAL (W#16#...)	Descripción
0000	No hay error
0081	Advertencia: Que el área de origen sea menor que la de destino. Los datos fuente se copian al completo con los bytes adicionales en el área de destino sin modificación.
7000	Llamada con REQ = 0: BUSY = 0
7001	Primera llamada con REQ = 1 (en curso): BUSY = 1
7002	Llamada N° (en curso): BUSY = 1
8051	Error de tipo de bloque de datos
8081	El área de origen es mayor que la de destino. El área de destino está llena por completo y los bytes restantes del origen se ignoran.

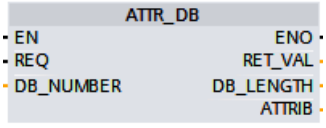
RET_VAL (W#16#...)	Descripción
8251	Error de tipo de bloque de datos de origen
82B1	El bloque de datos de origen falta
82C0	El DB de origen está siendo editado por otra instrucción o función de comunicación.
8551	Error de tipo de bloque de datos de destino
85B1	El bloque de datos de destino falta
85C0	El DB de destino está siendo editado por otra instrucción o función de comunicación.
80C3	Actualmente hay más de 50 instrucciones READ_DBL o 50 WRIT_DBL en cola de ejecución.

Consulte también Recetas (Página 478)

9.10.3 ATTR_DB (leer atributos de un bloque de datos)

Tabla 9-224 Instrucción ATTR_DB

KOP / FUP	SCL	Descripción

	`ret_val := ATTR_DB(`
	`REQ:=_bool_in_,`
	`DB_NUMBER:=_uint_in_,`
	`DB_LENGTH=>_udint_out_,`
	`ATTRIB=>_byte_out_);`
		La instrucción "ATTR_DB" se utiliza para obtener información acerca del bloque de datos (DB) ubicado en la memoria de trabajo de la CPU. La instrucción determina los atributos activados en el parámetro ATTRIB para el DB seleccionado.
		No es posible leer la longitud en bloques de datos con acceso optimizado ni en bloques de datos que solo están en la memoria de carga. En estos casos, el parámetro DB_LENGH tiene el valor "0".
		No utilice ATTR_DB para bloques de datos con acceso optimizado y reserva de memoria activada.
		No lea los bloques de datos de Motion Control con la instrucción "ATTR_DB". En este caso se emite el código de error 80B2.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "ATTR_DB":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L o constante	REQ = 1: Orden para leer atributos del bloque
DB_NUMBER	Input	DB_ANY	I, Q, M, D, L o constante	Número del DB que debe comprobarse
RET_VAL	Output	INT	I, Q, M, D, L	Información de error
DB_LENGTH	Output	UDINT	I, Q, M, D, L	• Número de bytes de datos que contiene el DB seleccionado. • "0" para bloques de datos con acceso optimizado y bloques de datos que solo están en la memoria de carga.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
ATTRIB	Output	BYTE	I, Q, M, D, L	Propiedades del DB:
				• Bit 0*= 0: El atributo "Depositar solo en la memoria de carga" no está activado. • Bit 0*= 1: El atributo "Depositar solo en la memoria de carga" está activado.
				• Bit 1 = 0: El atributo "Bloque de datos protegido contra escritura en el dispositivo" no está activado. • Bit 1 = 1: El atributo "Bloque de datos protegido contra escritura en el dispositivo" está activado.
				Si el bit 0 = 1, el bit 2 es irrelevante y se pone a 1. • Bit 2 = 0: Remanente - El DB se considera remanente si al menos uno de los valores se ha puesto a remanente. • Bit 2 = 1: No remanente - El DB en su conjunto no es remanente.
				• Bit 3*= 0: El DB está en la memoria de carga (bit 0 = 1) o en la memoria de trabajo (bit 0 = 0). • Bit 3*= 1: El DB se genera en la memoria de carga y de trabajo.
* La relación entre el bit 0 y el bit 3 se explica en los parámetros de la instrucción "CREATE_DB (crear bloque de datos) (Página 514)".				

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

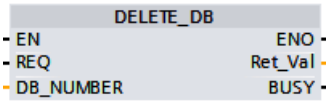
Parámetro RET_VAL

Código de error* (W#16#...)	Explicación
0000	No ha ocurrido ningún error.
80A1	Error en el parámetro de entrada DB_NUMBER: el parámetro actual seleccionado • Es "0" • Es mayor que el número de DB máximo permitido para la CPU utilizada.
80B1	El DB con el número especificado no existe en la CPU.
80B2	No es posible leer bloques de datos de objetos tecnológicos de Motion Control con la instrucción "ATTR_DB".
Información general del error	Consulte también: Códigos de error comunes para las instrucciones avanzadas (Página 531)
* El código de error puede visualizarse en el editor de programas como valor entero o hexadecimal.	

9.10.4 DELETE_DB (borrar bloque de datos)

Tabla 9-225 Instrucción DELETE_DB

KOP / FUP	SCL	Descripción

	`ret_val := DELETE_DB(`
	`  REQ := _bool_in_,`
	`  DB_NUMBER := _uint_in_,`
	`  BUSY => _bool_out_);`
		La instrucción "DELETE_DB" se utiliza para borrar un bloque de datos (DB) que el programa de usuario había creado llamando la instrucción "CREATE_DB (Página 514)".
		Si el bloque de datos no se ha creado con "CREATE_DB", DELETE_DB devuelve el código de error W#16#80B5 en el parámetro RET_VAL.
		La llamada de DELETE_DB no borra el bloque de datos seleccionado inmediatamente, sino en el punto de control del ciclo posterior a la ejecución del OB de ciclo.

Descripción del funcionamiento

La instrucción "DELETE_DB" funciona de forma asíncrona, es decir, su ejecución abarca varias llamadas. Inicie la transferencia de interrupción llamando la instrucción con REQ = 1.

El parámetro de salida BUSY y los bytes 2 y 3 del parámetro de salida RET_VAL indican el estado de la tarea.

La eliminación del bloque de datos se ha completado cuando el parámetro de salida BUSY tiene el valor FALSE.

Parámetros

La tabla siguiente muestra los parámetros de la instrucción "DELETE_DB":

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L o constante	REQ = 1: Petición para borrar el DB con el número del parámetro DB_NUMBER
DB_NUMBER	Input	UINT	I, Q, M, D, L o constante	Número del DB que debe borrarse
RET_VAL	Output	INT	I, Q, M, D, L	Información de error (consulte el parámetro "RET_VAL")
BUSY	Output	BOOL	I, Q, M, D, L	BUSY= 1: El proceso no ha finalizado aún.

Encontrará información adicional sobre los tipos de datos válidos en "Tipos de datos (Página 106)".

Parámetro RET_VAL

Código de error* (W#16#...)	Explicación
0000	No ha ocurrido ningún error.
7000	Primera llamada con REQ = 0: No hay ninguna transferencia de datos activa; BUSY tiene el valor "0".
7001	Primera llamada con REQ = 1: Transferencia de datos disparada; BUSY tiene el valor "1".
7002	Llamada intermedia (REQ irrelevante): Transferencia de datos ya activa; BUSY tiene el valor "1".
80A1	Error en el parámetro de entrada DB_NUMBER: • El valor del parámetro es "0". • El valor del parámetro es mayor que el número de DB máximo permitido para la CPU utilizada.
80B1	El DB con el número especificado no existe en la CPU.
80B4	El DB no puede borrarse porque la Memory Card de la CPU está protegida contra escritura.
80B5	El DB no se ha creado mediante "CREATE_DB".
80BB	No hay suficiente memoria de carga.
80C3	La función "Borrar un DB" no puede ejecutarse en este momento debido a una escasez temporal de recursos.
Información general del error	Consulte también: Códigos de error comunes para las instrucciones avanzadas (Página 531)
* El código de error puede visualizarse en el editor de programas como valor entero o hexadecimal.	

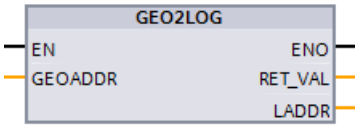
9.11 Procesamiento de direcciones

9.11.1 GEO2LOG (determinar la ID de hardware a partir del slot)

La instrucción GEO2LOG se utiliza para determinar el identificador de hardware en base a una información de slot.

Tabla 9-226 Instrucción GEO2LOG

KOP / FUP	SCL	Descripción

	`ret_val := GEO2LOG(`
	`  GEOADDR:=_variant_in_out_,`
	`  laddr:=_word_out_);`
		La instrucción GEO2LOG se utiliza para determinar el identificador de hardware en base a una información de slot.

La instrucción GEO2LOG determina el identificador de hardware sobre la base de la información de slot que se define usando el tipo de datos de sistema GEOADDR:

Dependiendo del tipo de hardware definido en el parámetro HWTYPE, se evalúa la información siguiente de los otros parámetros GEOADDR:

- Con HWTYPE = 1 (sistema PROFINET IO):
  - Solo se evalúa IOSYSTEM. El resto de parámetros de GEOADDR no se toman en consideración.
  - Se emite el identificador de hardware del sistema PROFINET IO.
- Con HWTYPE = 2 (dispositivo PROFINET IO):
  - Se evalúan IOSYSTEM y STATION. El resto de parámetros de GEOADDR no se toman en consideración.
  - Se emite el identificador de hardware del dispositivo PROFINET IO.
- Con HWTYPE = 3 (rack):
  - Solo se evalúan IOSYSTEM y STATION. El resto de parámetros de GEOADDR no se toman en consideración.
  - Se emite el identificador de hardware del rack.
- Con HWTYPE = 4 (módulo):
  - Se evalúan IOSYSTEM, STATION, y SLOT. El parámetro SUBSLOT de GEOADDR no se toma en consideración.
  - Se emite el identificador de hardware del módulo.
- Con HWTYPE = 5 (submódulo):
  - Se evalúan todos los parámetros de GEOADDR.
  - Se emite el identificador de hardware del submódulo.

El parámetro AREA del tipo de datos del sistema GEOADDR no se evalúa.

Tabla 9-227 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
GEOADDR	IN/OUT or IN ?	Variant	Puntero a la estructura de tipo de datos del sistema GEOADDR. El tipo de datos del sistema GEOADDR contiene la información de slot a partir de la que se determina la ID de hardware. Consulte el "Tipo de datos del sistema GEOADDR (Página 530)" para obtener más información.
RET_VAL	OUT or RETURN ?	Int	Salida de la información de error.
LADDR	OUT	HW_ANY	Identificador de hardware del conjunto o del módulo. El número se asigna automáticamente y se almacena en las propiedades de la configuración hardware.

Para obtener más información sobre los tipos de datos válidos, consulte la "Vista general de los tipos de datos válidos" en la Ayuda en pantalla de STEP 7.

Tabla 9-228 Códigos de condición

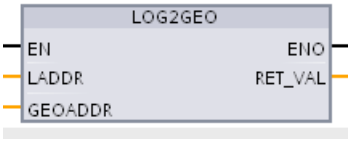
RET_VAL* (W#16#...)	Explicación
0	No ha ocurrido ningún error.
8091	Valor no válido en GEOADDR para HWTYPE.
8094	Valor no válido en GEOADDR para IOSYSTEM.
8095	Valor no válido en GEOADDR para STATION.
8096	Valor no válido en GEOADDR para SLOT.
8097	Valor no válido en GEOADDR para SUBSLOT.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas.	

9.11.2 LOG2GEO (determinar el slot a partir de la ID de hardware)

Utilice la instrucción LOG2GEO para determinar la dirección geográfica (slot de módulo) a partir de la dirección lógica perteneciente a un identificador de hardware.

Tabla 9-229 Instrucción LOG2GEO

KOP / FUP	SCL	Descripción

	`ret_val := LOG2GEO(`
	`  laddr:=_word_in_,`
	`  GEOADDR:=_variant_in_out_)`
	`;`
		Utilice la instrucción LOG2GEO para determinar el slot de módulo perteneciente a un identificador de hardware.

La instrucción LOG2GEO determina la dirección geográfica de una dirección lógica basada en el identificador de hardware:

- Use el parámetro LADDR para seleccionar la dirección lógica en función del identificador de hardware.
- GEOADDR contiene la dirección geográfica de la dirección lógica proporcionada en la entrada LADDR.

Nota

En los casos en los que el tipo HW no admita un componente, se devuelve un número de subslot para un módulo 0.

Se proporciona un error si la entrada LADDR no direcciona un objeto HW.

Tabla 9-230 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
LADDR	IN	HW_ANY	Identificador de hardware del sistema IO o el módulo. El número se asigna automáticamente y se almacena en las propiedades de la CPU o la interfaz de la configuración del hardware.
RET_VAL	OUT	Int	Código de error de la instrucción
GEOADDR	IN_OUT	Variant	Puntero al tipo de datos del sistema GEOADDR. El tipo de datos del sistema GEOADDR contiene la información de slot. Consulte el "Tipo de datos del sistema GEOADDR (Página 530)" para obtener más información.

Para obtener más información sobre los tipos de datos válidos, consulte la "Vista general de los tipos de datos válidos" en la Ayuda en pantalla de STEP 7.

Tabla 9-231 Códigos de condición

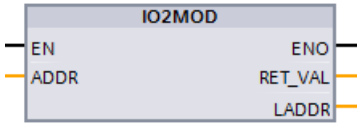
RET_VAL (W#16#...)	Descripción
0000	No hay error
8090	La dirección especificada en el parámetro LADDR no es válida.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas.	

9.11.3 IO2MOD (determinar la ID de hardware a partir de una dirección ES)

La instrucción IO2MOD se utiliza para determinar el identificador de hardware del módulo a partir de una dirección E/S de un (sub)módulo.

Tabla 9-232 Instrucción IO2MOD

KOP / FUP	SCL	Descripción

	`ret_val := IO2MOD(`
	`  ADDR:=_word_in_,`
	`  LADDR:=_word_out_);`
		Utilice la instrucción IO2MOD para determinar el slot de módulo perteneciente a un identificador de hardware.

La instrucción IO2MOD determina el identificador de hardware del módulo a partir de una dirección E/S (I, Q, PI, PQ) de un (sub)módulo.

Introduzca la dirección E/S en el parámetro ADDR. Si en este parámetro se usa una serie de direcciones E/S, solo se evalúa la primera dirección para determinar el identificador de hardware. Si la primera dirección se ha especificado de forma correcta, la longitud para la especificación de la dirección en ADDR no es relevante. Si se usa un área de direcciones que abarca varios módulos o direcciones no empleadas, también puede determinarse el identificador de hardware del primer módulo.

Si no se especifica ninguna dirección E/S de un (sub)módulo en el parámetro ADDR, se emite el código de error "8090" en el parámetro RET_VAL.

Nota

Entrada de una dirección E/S en SCL

No es posible programar usando la ID de acceso E/S "%QWx:P" en SCL. En este caso, use el nombre simbólico de la variable o la dirección absoluta en la memoria imagen de proceso.

Tabla 9-233 Tipos de datos para los parámetros

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
ADDR	IN or IN/OUT ?	Variant	I, Q, M, D, L	Dirección E/S (I, Q, PI, PQ) dentro de un (sub)módulo. Asegúrese de que el acceso al segmento no se usa para el parámetro ADDR. Si fuera así, en el parámetro LADDR se emiten valores incorrectos.
RET_VAL	OUT or RETURN ?	Int	I, Q, M, D, L	Código de error de la instrucción.
LADDR	OUT	HW_IO	I, Q, M, D, L	Identificador de hardware determinado (dirección lógica) del (sub)módulo E/S.

Para obtener más información sobre los tipos de datos válidos, consulte la "Vista general de los tipos de datos válidos" en la Ayuda en pantalla de STEP 7.

Tabla 9-234 Códigos de condición

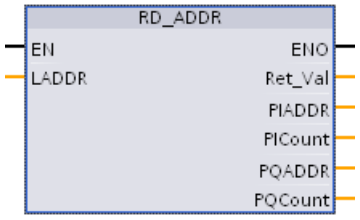
RET_VAL* (W#16#...)	Explicación
0	No ha ocurrido ningún error.
8090	La dirección E/S especificada en el parámetro ADDR no se utiliza en ningún componente de hardware.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas.	

9.11.4 RD_ADDR (determinar las direcciones ES a partir de la ID de hardware)

Use la instrucción RD_ADDR para obtener las direcciones de E/S de un submódulo.

Tabla 9-235 Instrucción RD_ADDR

KOP / FUP	SCL	Descripción

	`ret_val := RD_ADDR(`
	`laddr:=_word_in_,`
	`PIADDR=>_udint_out_,`
	`PICount=>_uint_out_,`
	`PQADDR=>_udint_out_,`
	`PQCount=>_uint_out_,);`
		Use la instrucción RD_ADDR para obtener las direcciones de E/S de un submódulo.

La instrucción RD_ADDR determina la longitud y la dirección inicial de las entradas o salidas en función del identificador de hardware de un submódulo:

- Use el parámetro LADDR para seleccionar el módulo de entradas o salidas en función del identificador de hardware.
- Los siguientes parámetros de salida se usan en función de si se trata de un módulo de entradas o salidas:
  - En el caso de un módulo de entrada, los valores determinados se obtienen en los parámetros PIADDR y PICOUNT.
  - En el caso de un módulo de salida, los valores determinados se obtienen en los parámetros PQADDR y PQCOUNT.
- Los parámetros PIADDR y PQADDR contienen respectivamente la dirección inicial de las direcciones de E/S del módulo.
- Los parámetros PICOUNT y PQCOUNT contienen respectivamente el número de bytes de las entradas o salidas (1 byte para 8 entradas/salidas, 2 bytes para 16 entradas/salidas).

Tabla 9-236 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
LADDR	IN	HW_IO	Identificador de hardware del (sub)módulo
RET_VAL	OUT	Int	Código de error de la instrucción
PIADDR	OUT	UDInt	Dirección inicial del módulo de entrada
PICOUNT	OUT	UInt	Número de bytes de las entradas
PQADDR	OUT	UDInt	Dirección inicial del módulo de salida
PQCOUNT	OUT	UInt	Número de bytes de las salidas

Para obtener más información sobre los tipos de datos válidos, consulte la "Vista general de los tipos de datos válidos" en la Ayuda en pantalla de STEP 7.

Tabla 9-237 Códigos de condición

RET_VAL (W#16#...)	**Descripción**
0000	No hay error
8090	El identificador de hardware del módulo en el parámetro LADDR no es válido.
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas.	

9.11.5 Tipo de datos del sistema GEOADDR

Dirección geográfica

El tipo de datos del sistema GEOADDR incluye la dirección geográfica de un módulo (o la información de slot).

- Dirección geográfica para PROFINET IO:
  Para PROFINET IO, la dirección geográfica está compuesta por la ID del sistema PROFINET IO, el número de dispositivo, el número de slot y el submódulo (si se usa un submódulo).
- Dirección geográfica para PROFINET DP:
  Para PROFIBUS DP, la dirección geográfica consiste en la ID del sistema maestro DP, el número de estación y el número de slot.

La información de slot de los módulos está depositada en la configuración hardware de cada módulo.

Estructura del tipo de datos del sistema GEOADDR

La estructura GEOADDR se crea automáticamente si introduce "GEOADDR" como tipo de datos en un bloque de datos.

Nombre del parámetro	Tipo de datos	Descripción
GEOADDR	STRUCT	
HWTYPE	UINT	Tipo de hardware: • 1: sistema IO (PROFINET/PROFIBUS) • 2: dispositivo IO/esclavo DP • 3: rack • 4: módulo • 5: submódulo Si la instrucción no admite el tipo de hardware, se emite HWTYPE "0".
AREA	UINT	ID de área: • 0 = CPU • 1 = PROFINET IO • 2 = PROFIBUS DP • 3 = AS-i
IOSYSTEM	UINT	Sistema PROFINET IO (0=unidad central en el rack)
STATION	UINT	• Número del rack si el identificador de área AREA = 0 (módulo central). • Número de estación si el identificador de área AREA > 0.
SLOT	UINT	Número de slot
SUBSLOT	UINT	Número de submódulo. Este parámetro tiene el valor "0" si no hay ningún submódulo o no puede conectarse.

9.12 Códigos de error comunes para las instrucciones avanzadas

Tabla 9-238 Códigos de condición comunes para las instrucciones avanzadas

Código de condición (W#16#....)[1]	Descripción
8x22[2]	Área demasiado pequeña para la entrada
8x23	Área demasiado pequeña para la salida
8x24	Área de entrada no válida
8x25	Área de salida no válida
8x28	Asignación del bit de entrada no válida
8x29	Asignación del bit de salida no válida
8x30	El área de salida es un DB de solo lectura.
8x3A	El DB no existe.

[1] Si uno de estos errores se produce cuando se ejecuta un bloque lógico, la CPU se mantendrá en RUN (valor predeterminado) o se puede configurar para que pase a STOP. Opcionalmente, puede usar las instrucciones GetError o GetErorID de ese bloque lógico para procesar el error localmente (la CPU se mantiene en RUN) y crear una reacción programada ante el error.

[2] La "x" representa el número del parámetro con error. Los números de parámetro comienzan con 1.

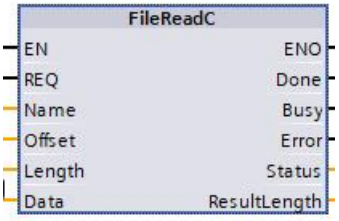
9.13 Tratamiento de archivos

9.13.1 FileReadC: leer archivo de la SIMATIC Memory Card

La instrucción "FileReadC" se utiliza para leer datos del archivo ubicado en la SIMATIC Memory Card y escribirlos en un área de destino de la CPU. El archivo se especifica introduciendo el nombre y la ruta completa.

Tabla 9-239 Instrucción FileReadC

KOP/FUP	SCL	Descripción

	`"FileReadC_SFB_DB_2"(`
	`REQ:=_bool_in_`
	`Name:=_string_in_`
	`Offset:=_udint_in_,`
	`Length:=_udint_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`Error=>_bool_out_,`
	`Status=>_word_out_,`
	`ResultLength=>_udint_out_,`
	`Data:=_variant_inout_);`
		La instrucción "FileReadC" se utiliza para leer datos del archivo ubicado en la SIMATIC Memory Card y escribirlos en un área de destino de la CPU.

Active la lectura del archivo mediante el parámetro de control REQ. Los parámetros de salida Done, Busy, Error y Status indican el estado de la tarea.

Para la especificación del nombre del archivo y de la ruta deben observarse las siguientes normas:

- El nombre del archivo debe tener más de 55 caracteres.
- Los siguientes caracteres se admiten para el nombre de la carpeta y del archivo: "0" a "9", "a" a "z", "A" a "Z", "-", "_"
- El nombre del archivo no puede contener más de un punto ("."), que separa el nombre de la extensión. El nombre de archivo debe contener como mínimo un carácter. No es necesaria una extensión.
- El nombre de la ruta no puede comenzar por "/", "\" o ".".
- El nombre de la ruta no puede contener "..".
- El nombre de la ruta no puede contener una subcarpeta en la carpeta UserFiles.

Ejemplos de rutas y nombres de archivo válidos: "UserFiles/Lift16_DataBase.txt", "UserFiles/2017-04-13_ErrorLog.bin"

Se pueden leer hasta 16 MB (16.777.216 bytes) ejecutando "FileReadC"; los datos se leen segmento a segmento. El tamaño del segmento es específico de cada dispositivo. Una CPU S7-1500, por ejemplo, utiliza bloques de 32 KB (32.768 bytes). Si el número de elementos de datos que deben leerse es mayor que el tamaño del segmento, será necesario llamar la instrucción varias veces en el programa. Por razones de coherencia, no conviene acceder a los datos leídos hasta llamar la instrucción por última vez.

Los parámetros "Offset" y "Length" especifican la longitud de los elementos de los datos que deben leerse. El recurso ocupado se libera de nuevo una vez completado el proceso de lectura.

La instrucción "FileReadC" funciona de forma asíncrona. El procesamiento se extiende a lo largo de varias llamadas y comienza con un flanco ascendente en el parámetro "REQ".

Los parámetros "Busy" y "Done" indican el estado de la tarea.

Si se produce un error durante la ejecución, los parámetros "Error" y "Status" lo indican.

Tabla 9-240 Tipos de datos de los parámetros

Parámetros y tipo		Tipo de datos	Descripción
REQ	Input	BOOL	Petición de parámetro de control Permite leer un archivo de la SIMATIC Memory Card con un flanco ascendente.
Name	Input	STRING	Nombre del archivo que debe leerse, incluida la ruta completa
Offset	Input	UDINT	Offset de byte después del cual hay que leer el archivo
Length	Input	UDINT	Longitud del área que debe leerse en bytes Length = 0 significa que se ha leído el número máximo de elementos de datos posible por llamada (para una CPU S7-1200, 8 KB o el tamaño del parámetro "Data")
Done	Output	BOOL	1: la instrucción se ha ejecutado correctamente. La información leída se ha transferido al parámetro "Data".
Busy	Output	BOOL	Parámetro de estado • 0: ejecución de la instrucción finalizada o no iniciada aún. • 1: ejecución de la instrucción no finalizada aún.
Error	Output	BOOL	Parámetro de estado • 0: sin errores. • 1: se ha producido un error al ejecutar la instrucción. Se proporciona información detallada a través del parámetro "Status".
Status	Output	WORD	Código de error
ResultLength	Output	UDINT	Longitud de los datos leídos en bytes
Data	InOut	VARIANT	Área de destino de los datos leídos Tipos de datos permitidos: BYTE y ARRAY of BYTE

Tabla 9-241 Códigos de condición

Código de error* (W#16#...)	Explicación
0	Sin errores
7000	No se está procesando ninguna tarea
7001	Inicio del procesamiento de la tarea. Parámetro Busy = 1, Done = 0.
7002	Llamada intermedia (REQ irrelevante): instrucción ya activa; Busy tiene el valor "1".
0081	Advertencia: Offset + Length excede la longitud del archivo. Se leen datos de "Offset" al final del archivo. "Data" contiene menos elementos de datos que los solicitados; "ResultLength" es más corto que "Length"; los datos que superan "ResultLength" en "Data" no cambian.
8091	La ruta no existe o no es válida.
8092	El parámetro "Name" no es del tipo de datos "STRING", es demasiado largo o contiene caracteres no válidos.

Código de error* (W#16#...)	Explicación
8093	El parámetro "Offset" apunta más allá del final del archivo que debe leerse.
8094	El parámetro "Length" es mayor que el valor máximo permitido. Para una CPU S7-1500 o S7-1200, por ejemplo, el valor máximo permitido es 16 MB (16.777.216 bytes).
80A1	Error de lectura; el área de destino especificada por el parámetro "Data" puede sobrescribirse parcialmente.
80B1	El área de destino especificada por el parámetro "Data" no alcanza la longitud requerida en el parámetro "Length".
80C0	No es posible acceder al archivo (está protegido contra escritura o bloqueado por otro proceso).
80C3	Se ha alcanzado el número máximo de instrucciones "FileReadC" activas al mismo tiempo.
8A30	La zona de destino está protegida contra escritura, por ejemplo un DB con protección contra escritura.
8A3A	"Data" apunta a un área no permitida, por ejemplo, a la memoria de carga o a los datos locales.
8A51	Tipo de datos no válido del parámetro "Data".
8A52	La variable del parámetro Data no es suficiente. Puede haberse escrito una parte de los datos del área de origen.
* Los códigos de error del editor de programas se muestran como valores enteros o hexadecimales. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".	

Consulte también

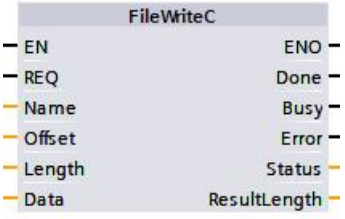
Nuevas funciones (Página 35)

9.13.2 FileWriteC: escribir archivo en la Memory Card

La instrucción "FileWriteC" se utiliza para escribir datos ubicados en un área de origen de la CPU en un archivo ubicado en la carpeta "UserFiles" de la SIMATIC Memory Card.

Tabla 9-242 Instrucción FileWriteC

KOP/FUP	SCL	Descripción

	`"FileWriteC_SFB_DB_1"(`
	`REQ:=_bool_in_,`
	`Name:=_string_in_,`
	`Offset:=_udint_in_,`
	`Length:=_udint_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`Error=>_bool_out_,`
	`Status=>_word_out_,`
	`ResultLength=>_udint_out_,`
	`Data:=_variant_inout_);`
		La instrucción "FileWriteC" se utiliza para escribir datos ubicados en un área de origen de la CPU en un archivo ubicado en la carpeta "UserFiles" de la SIMATIC Memory Card. El archivo se especifica introduciendo el nombre y la ruta completa. Si el archivo no existe, la CPU lo crea en la carpeta "UserFiles". Si la carpeta "UserFiles" tampoco existe, la CPU también la crea. No se crean subcarpetas; en este caso W#16#8091 se deposita en el parámetro "Status".

Active la escritura en el archivo mediante el parámetro de control "REQ". Los parámetros de salida "Done", "Busy", "Error" y "Status" indican el estado de la tarea.

Para la especificación del nombre del archivo y de la ruta deben observarse las siguientes normas:

- El nombre del archivo debe tener más de 55 caracteres.
- Los siguientes caracteres se admiten para el nombre de la carpeta y del archivo: "0" a "9", "a" a "z", "A" a "Z", "-", "_"
- El nombre del archivo no puede contener más de un punto ("."), que separa el nombre de la extensión. El nombre de archivo debe contener como mínimo un carácter. No es necesaria una extensión.
- El nombre de la ruta no puede comenzar por "/", "\" o ".".
- El nombre de la ruta no puede contener "..".
- El nombre de la ruta no puede contener una subcarpeta en la carpeta UserFiles.

Ejemplos de rutas y nombres de archivo válidos: "UserFiles/Lift16_DataBase.txt", "UserFiles/2017-04-13_ErrorLog.bin"

Se pueden escribir hasta 16 MB (16.777.216 bytes) ejecutando "FileWriteC"; los datos se escriben segmento a segmento. El tamaño del segmento es específico de cada dispositivo. Una CPU S7-1200, por ejemplo, utiliza bloques de 8 KB (8.192 bytes). Si el número de elementos de datos que deben escribirse es mayor que el tamaño del segmento, será necesario llamar la instrucción varias veces en el programa. Por razones de coherencia, no conviene acceder a los datos escritos hasta llamar la instrucción por última vez. Si la longitud del archivo disponible no es suficiente, se extiende hasta el tamaño necesario.

Los parámetros "Offset" y "Length" especifican la ubicación en el archivo en el que deben escribirse los datos. El recurso ocupado se libera de nuevo una vez completado el proceso de escritura.

La instrucción "FileWriteC" solo inicia la operación de escritura si se cumple la siguiente condición: "Offset" + "Length" <= 16 MB.

La instrucción "FileWriteC" funciona de forma asíncrona. El procesamiento se extiende a lo largo de varias llamadas y comienza con un flanco ascendente en el parámetro "REQ".

Los parámetros "Busy" y "Done" indican el estado de la tarea.

Si se produce un error durante la ejecución, los parámetros "Error" y "Status" lo indican.

Tabla 9-243 Tipos de datos de los parámetros

Parámetros y tipo		Tipo de datos	Descripción
REQ	Input	BOOL	Petición de parámetro de control Permite escribir un archivo en la SIMATIC Memory Card con un flanco ascendente.
Name	Input	STRING	Nombre del archivo que debe escribirse, incluida la ruta completa
Offset	Input	UDINT	Offset de byte a partir del cual hay que escribir el archivo
Length	Input	UDINT	Longitud del área que debe escribirse en bytes "Length" = 0 significa que se escribe toda el área de origen especificada con el parámetro "Data".
Done	Output	BOOL	1: la instrucción se ha ejecutado correctamente.

Parámetros y tipo		Tipo de datos	Descripción
Busy	Output	BOOL	Parámetro de estado • 0: ejecución de la instrucción finalizada o no iniciada aún. • 1: ejecución de la instrucción no finalizada aún.
Error	Output	BOOL	Parámetro de estado • 0: sin errores. • 1: se ha producido un error al ejecutar la instrucción. Se proporciona información detallada a través del parámetro "Status".
Status	Output	WORD	Código de error
ResultLength	Output	UDINT	Longitud de los datos escritos en bytes
Data	InOut	VARIANT	Área de origen Tipos de datos permitidos: BYTE y ARRAY of BYTE

Tabla 9-244 Códigos de condición

Código de error* (W#16#...)	Explicación
0	Sin errores
7000	No se está procesando ninguna tarea
7001	Inicio del procesamiento de la tarea. Parámetro Busy = 1, Done = 0.
7002	Llamada intermedia (REQ irrelevante): instrucción ya activa; Busy tiene el valor "1".
8091	La ruta no existe o no es válida.
8092	El parámetro "Name" no es del tipo de datos "STRING", es demasiado largo o contiene caracteres no válidos.
8093	• El parámetro "Offset" apunta más allá del final del archivo que debe escribirse. • La creación del archivo se rechaza porque "Offset" es mayor que cero.
8094	• "Length" es mayor que el valor máximo permitido. Para una CPU S7-1500, por ejemplo, el valor máximo permitido es 16 MB (16.777.216 bytes). • "Length" + "Offset" es mayor que el valor máximo permitido.
80A1	Error de escritura; es posible que los datos del archivo de la SIMATIC Memory Card se hayan sobrescrito parcialmente.
80B1	El área de origen especificada por el parámetro "Data" no alcanza la longitud requerida en el parámetro "Length".
80B3	No hay suficiente espacio en la SIMATIC Memory Card o en la memoria de carga interna.
80B4	La SIMATIC Memory Card o el archivo están protegidos contra escritura.
80C0	No es posible acceder al archivo.
80C3	Se ha alcanzado el número máximo de instrucciones "FileWriteC" activas al mismo tiempo.
8A24	"Data" apunta a un área no permitida, por ejemplo, a la memoria de carga o a los datos locales.
8A51	Tipo de datos no válido del parámetro "Data".
8A52	La variable del parámetro Data no es suficiente. Puede haberse escrito una parte de los datos del área de origen.
* Los códigos de error del editor de programas se muestran como valores enteros o hexadecimales. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".	

Consulte también

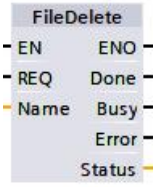
Nuevas funciones (Página 35)

9.13.3 FileDelete: Borrar archivo de la Memory Card

La instrucción "FileDelete" se utiliza para borrar un archivo existente de la SIMATIC Memory Card.

Tabla 9-245 Instrucción FileDelete

KOP/FUP	SCL	Descripción

	`"FileDelete_DB_1"(`
	`REQ:=_bool_in_,`
	`Name:=_string_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`Error=>_bool_out_,`
	`Status=>_word_out_)`
		La instrucción "FileDelete" se utiliza para borrar un archivo existente de la SIMATIC Memory Card.

El archivo no puede estar abierto. En el parámetro "Name" no se soportan comodines, lo que significa que los nombres como "UserFiles/*.txt" y "UserFiles/?.txt" no están permitidos.

La instrucción "FileDelete" solo está permitida en las carpetas "Receta" y "UserFiles". Estas carpetas pueden contener otras carpetas, p. ej., "UserFiles/Test/file1.txt".

Nota

Borrar registros de datos

No está permitido borrar un archivo de la carpeta "DataLog" con la instrucción "FileDelete". Para borrar registros de datos hay que utilizar la instrucción "DataLogDelete".

"FileDelete" es una instrucción asíncrona. Su ejecución puede extenderse a lo largo de varias llamadas. y comienza con un flanco ascendente en el parámetro "REQ".

Los parámetros "Busy" y "Done" indican el estado de la tarea.

Si se produce un error durante la ejecución, los parámetros "Error" y "Status" lo indican.

Tabla 9-246 Tipos de datos de los parámetros

Parámetros y tipo		Tipo de datos	Descripción
REQ	Input	BOOL	Petición de parámetro de control La ejecución se inicia con un flanco ascendente en REQ.
Name	Input	STRING	Ruta y nombre del archivo que debe borrarse
Done	Output	BOOL	Parámetro de estado • 1: la instrucción se ha ejecutado correctamente.
Busy	Output	BOOL	Parámetro de estado • 0: la instrucción no se está ejecutando. • 1: la instrucción se está ejecutando.

Parámetros y tipo		Tipo de datos	Descripción
Error	Output	BOOL	Parámetro de estado • 0: Sin errores • 1: se ha producido un error al ejecutar la instrucción. Para más detalles consulte el parámetro "Status".
Status	Output	WORD	Código de error

Tabla 9-247 Códigos de condición

Código de error* (W#16#...)	Explicación
0000	La instrucción se ha completado correctamente
7000	No se está procesando ninguna tarea
7001	Inicio del procesamiento de la tarea: Busy = 1, Done = 0.
7002	Llamada intermedia (REQ irrelevante): Instrucción ya activa; Busy tiene el valor "1".
8090	Archivo bloqueado, p. ej., porque está abierto
8091	La ruta no existe o no es válida.
8092	El archivo no existe en la ruta
80A2	Error de escritura
80A3	El archivo es demasiado grande (>= 2147483648 bytes) y no puede borrarse utilizando "FileDelete".
80B4	La SIMATIC Memory Card está protegida contra escritura
80C3	Se ha alcanzado el número máximo de instrucciones FileDelete activas al mismo tiempo.
* Los códigos de error del editor de programas se muestran como valores enteros o hexadecimales.	

Consulte también

Nuevas funciones (Página 35)

Instrucciones tecnológicas 10

10.1 Contaje (contadores rápidos)

Las instrucciones básicas con contadores, descritas en "Operaciones con contador" (Página 226), están limitadas a eventos de contaje que se producen a una velocidad inferior al ciclo de la CPU S7-1200. La función de contador rápido (HSC) ofrece la posibilidad de contar impulsos que ocurren a una velocidad mayor que el ciclo del PLC. Además, el HSC puede configurarse para que mida la frecuencia y el período de los impulsos que se han producido, o bien para que Motion Control pueda utilizar el HSC para leer una señal de encóder del motor.

Para utilizar la función de HSC, este debe activarse y configurarse primero en la ficha Propiedades de la CPU, en la pantalla de configuración de dispositivos. Para empezar a configurar el HSC, consulte "Configurar un contador rápido" (Página 554).

Después de descargar la configuración hardware, el HSC puede contar impulsos o medir frecuencias sin necesidad de llamar instrucciones. Cuando el HSC está en modo de contaje o período, el valor de contaje se captura automáticamente y se actualiza en la memoria imagen de proceso (de las entradas) cada ciclo. Si el HSC está en modo de frecuencia, el valor de la memoria imagen de proceso es la frecuencia en Hz.

Además de contar y medir, el HSC puede generar eventos de alarma de proceso, funcionar según el estado de los puntos de entrada físicos y generar un impulso de salida de acuerdo con un evento de contador especificado (solo CPU V4.2 o superiores). La instrucción tecnológica CTRL_HSC_EXT permite al programa de usuario controlar el HSC en relación al programa. CTRL_HSC_EXT actualiza parámetros del HSC y devuelve los valores más actuales cuando se ejecuta. La instrucción CTRL_HSC_EXT puede utilizarse cuando el HSC está en modo de contaje, período o frecuencia.

Nota

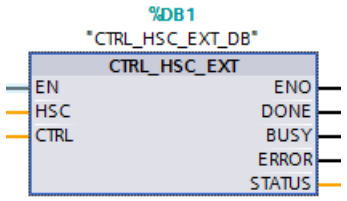
La instrucción CTRL_HSC_EXT sustituye la antigua instrucción CTRL_HSC para proyectos con CPU V4.2 y posteriores. La instrucción CTRL_HSC_EXT incluye toda la funcionalidad de la instrucción CTRL_HSC y algunas funciones adicionales. La antigua instrucción CTRL_HSC solo está disponible para la compatibilidad con proyectos S7-1200 antiguos y no debería utilizarse en proyectos nuevos.

10.1.1 Instrucción CTRL_HSC_EXT (controlar contador rápido)

10.1.1.1 Descripción general de la instrucción

Tabla 10-1 Instrucción CTRL_HSC_EXT

KOP / FUP	SCL	Descripción

	`"CTRL_HSC_1_DB" (`
	`hsc:=_hw_hsc_in_,`
	`done:=_done_out_,`
	`busy:=_busy_ out_,`
	`error:=_error_out_,`
	`status:=_status_out_,`
	`ctrl:=_variant_in_);`
		Toda instrucción CTRL_HSC_EXT (controlar contadores rápidos (avanzado)) utiliza una estructura de datos definida por el sistema y almacenada en un DB global definido por el usuario para guardar los datos de contador. Los tipos de datos HSC_Count, HSC_Period o HSC_Frequency se asignan a la instrucción CTRL_HSC_EXT como parámetros de entrada.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "CTRL_HSC_1_DB" es el nombre del DB de instancia.

Tabla 10-2 Tipos de datos para los parámetros

Parámetro	Declaración	Tipo de datos	Descripción
HSC	IN	HW_HSC	Identificador del HSC
CTRL	IN_OUT	Variant	Entrada de SFB y datos de retorno. Nota: Encontrará más información en "Tipos de datos de sistema (SDT) de la instrucción CTRL_HSC_EXT (Página 544)".
DONE	OUT	Bool	1= Indica que el SFB está terminado. Siempre 1 porque el SFB es síncrono
BUSY	OUT	Bool	Siempre 0, la función nunca está ocupada
ERROR	OUT	Bool	1 = Indica un error
STATUS	OUT	Word	Código de condición de ejecución Nota: consulte la tabla inferior "Códigos de condición de ejecución" para obtener más información.

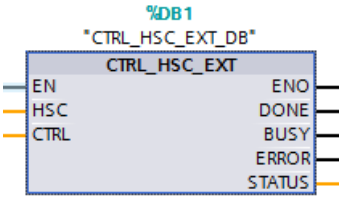
Tabla 10-3 Códigos de condición de ejecución

STATUS (W#16#)	Descripción
0	No hay error
80A1	El identificador HSC no direcciona un HSC
80B1	Valor de NewDirection no permitido
80B4	Valor de NewPeriod no permitido
80B5	Valor de NewOpModeBehavior no permitido
80B6	Valor de NewLimitBehavior no permitido
80D0	El SFB 124 no está disponible

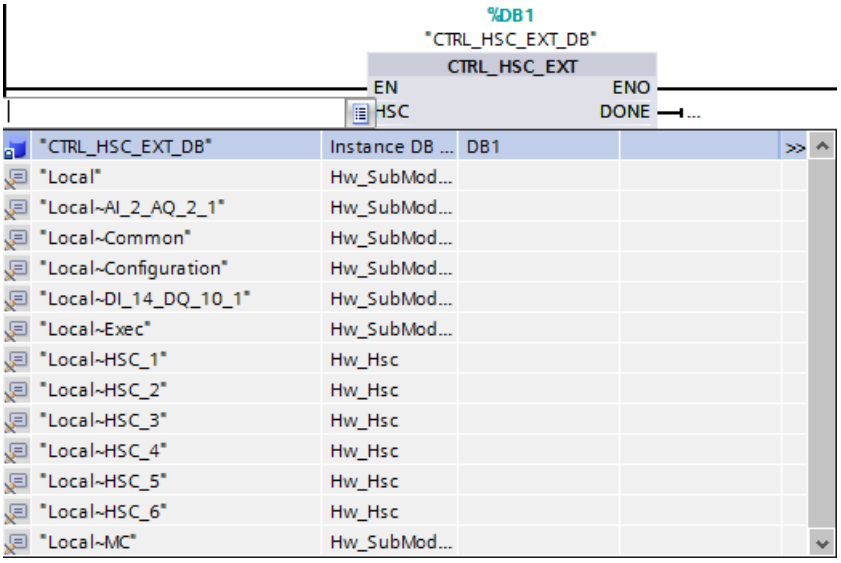
10.1.1.2 Ejemplo

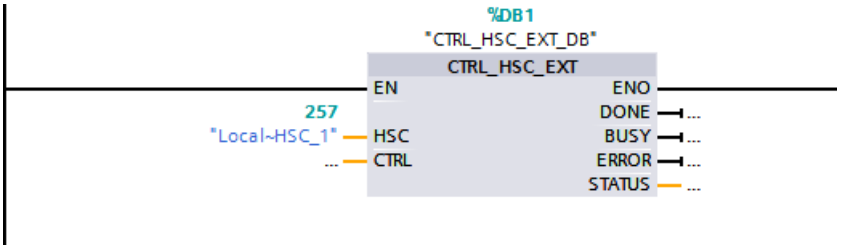
Para utilizar la instrucción CTRL_HSC_EXT, proceda del siguiente modo:

1. Coloque la instrucción CTRL_HSC_EXT en el segmento KOP, con lo que también se crea el siguiente bloque de datos de instancia: "CTRL_HSC_EXT_DB":



2. Adjunte el identificador de hardware del HSC, que se encuentra en las propiedades del HSC, al pin "HSC" de la instrucción KOP. También puede seleccionar uno de los seis objetos "Hw_Hsc" del menú desplegable de este pin de entrada. El nombre de variable predeterminado para HSC1 es "Local~HSC_1":



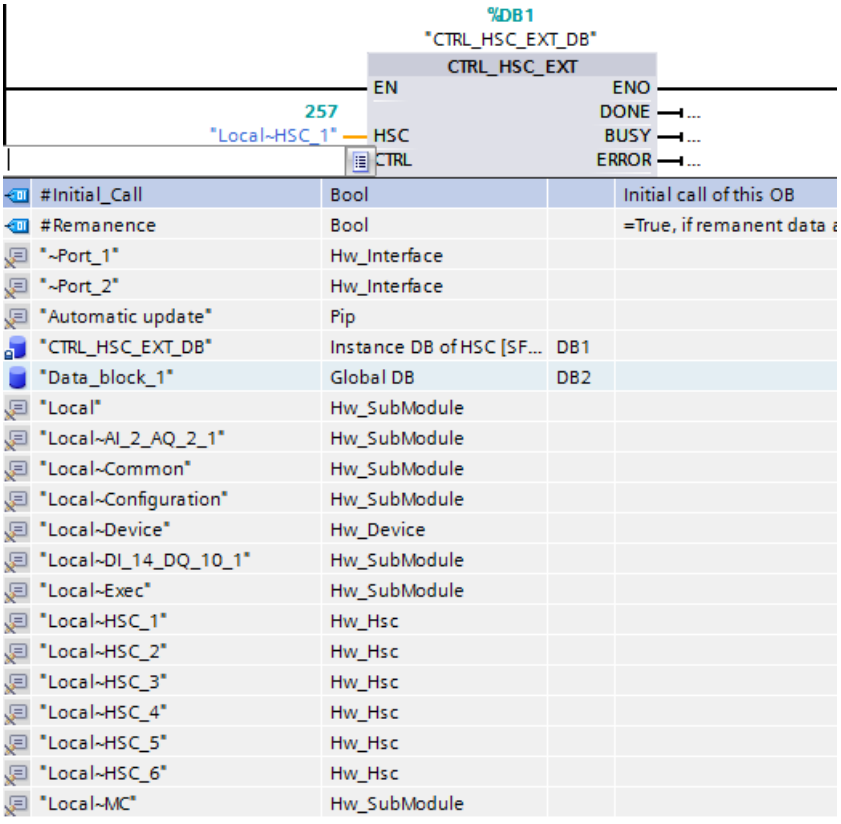


3. Cree un bloque de datos global denominado "Data_block_1" (también puede utilizar un bloque de datos global que ya exista):
   - Dentro de "Data_block_1", busque una fila vacía y agregue una variable que tenga por nombre "MyHSC".
   - En la columna "Tipo de datos", agregue uno de los siguientes tipos de datos del sistema (SDT). Seleccione el SDT que equivalga al tipo de contaje configurado para el HSC. Encontrará más información sobre los SDT del HSC más adelante. La lista desplegable no contiene estos tipos, de modo que tiene que asegurarse de escribir el nombre del SDT exactamente como se muestra: HSC_Count, HSC_Period o HSC_Frequency
   - Después de introducir el tipo de datos, expanda la variable "MyHSC" para ver todos los campos que contiene la estructura de datos. Aquí encontrará el tipo de datos de cada uno de los campos y podrá modificar los valores iniciales predeterminados:

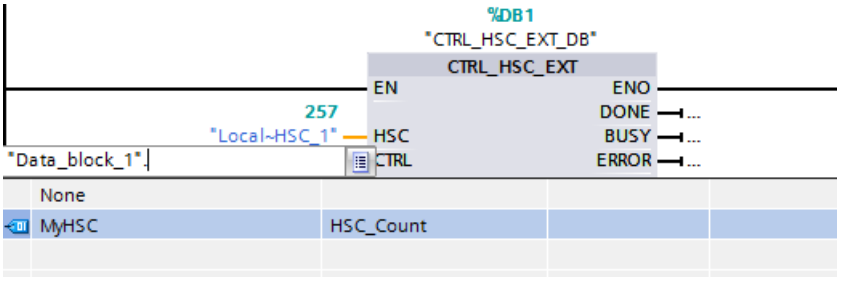
Data_block_1

	Nombre	Tipo de datos	Valor de arranque
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. Asigne la variable "'Data_block_1'. MyHSC" al pin de entrada CTRL de la instrucción CTRL_HSC_EXT:
   - Seleccione "Data_block_1".



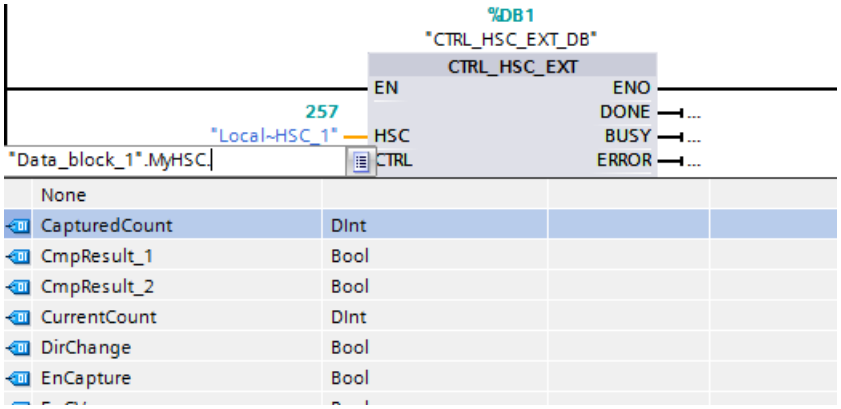
– Seleccione "MyHSC".



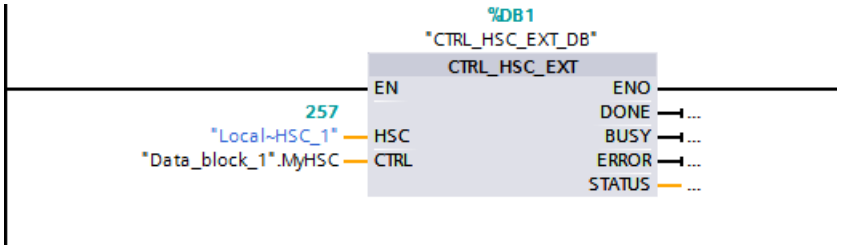
– Borre el período (".") que sigue a: "'Data_block_1'.MyHSC". A continuación, haga clic fuera del cuadro o pulse la tecla ESC una vez y pulse seguidamente la tecla Intro.

Nota

Después de borrar el período (".") que sigue a: "'Data_Block_1'.MyHSC", no pulse únicamente la tecla Intro. Esta acción vuelve a poner el período (".") en el cuadro.



– Ahora se muestra en la parte inferior la entrada CTRL completa.



Una vez se ha configurado el HSC en el PLC ya puede ejecutarse la instrucción CTRL_HSC_EXT. Si se produce un error, ENO se pone a "0" y la salida STATUS indica el código de condición.

Consulte también

Tipos de datos de sistema (SDT) de la instrucción CTRL_HSC_EXT (Página 544)

10.1.1.3 Tipos de datos de sistema (SDT) de la instrucción CTRL_HSC_EXT

Los siguientes tipos de datos del sistema (SDT) solo se utilizan con el pin CTRL de la instrucción CTRL_HSC_EXT. Para utilizarlos, cree un bloque de datos de usuario y agregue un objeto con el tipo de datos del SDT que equivalga al modo configurado (tipo de contaje) para el HSC. STEP 7 no muestra estos tipos de datos en el menú desplegable. Introduzca el nombre del SDT exactamente de la forma indicada.

Las entradas del SDT del HSC se indican con el prefijo "En" o "New". Las entradas con el prefijo "En" activan una función del HSC o actualizan el parámetro en cuestión. El prefijo "New" identifica el valor de actualización. Para que el nuevo valor sea efectivo, el bit "En" correspondiente debe ser True y el valor de "New" debe ser válido. Cuando se ejecuta la instrucción CTRL_HSC_EXT, el programa aplica cambios en las entradas y actualiza las salidas con la referencia adjunta adecuada del SDT.

SDT: HSC_Count

El tipo de datos "HSC_Count" equivale a un HSC configurado para el modo de contaje. El modo de contaje ofrece las siguientes funcionalidades:

- Acceder al contaje de impulsos actual
- Congelar el contaje de impulsos actual en un evento de entrada
- Restablecer el valor inicial del contaje de impulsos actual en un evento de entrada
- Acceder a los bits de estado, que indican que se han producido determinados eventos del HSC
- Desactivar el HSC mediante una entrada de software o hardware
- Modificar el sentido de contaje mediante una entrada de software o hardware
- Modificar el contaje de impulsos actual
- Modificar el valor inicial (que se utiliza cuando la CPU pasa al estado RUN o cuando se dispara la función de sincronización)
- Modificar dos valores de referencia (o predefinidos) independientes utilizados para la comparación
- Modificar los límites superior e inferior de contaje
- Modificar el funcionamiento del HSC cuando el contaje de impulsos alcanza dichos límites
- Generar un evento de alarma de proceso cuando el contaje de impulsos actual alcanza un valor de referencia (predefinido)
- Generar un evento de alarma de proceso cuando se activa la entrada de sincronización (reset)
- Generar un evento de alarma de proceso cuando el sentido de contaje cambia debido a una entrada externa
- Generar un impulso de salida individual cuando se produce un evento de contaje concreto

Cuando se produce un evento y se ejecuta la instrucción CTRL_HSC_EXT, la instrucción activa un bit de estado. La próxima vez que se ejecute la instrucción CTRL_HSC_EXT, esta desactivará el bit de estado, a menos que el evento se produzca de nuevo antes de que se ejecute la instrucción.

Tabla 10-4 Estructura de HSC_Count

Elemento de estructura	Declaración	Tipo de datos	Descripción
CurrentCount	OUT	Dint	Devuelve el valor de contaje actual del HSC
CapturedCount	OUT	Dint	Devuelve el valor del contador capturado en el evento de entrada especificado
SyncActive	OUT	Bool	Bit de estado: Se ha activado la entrada de sincronización
DirChange	OUT	Bool	Bit de estado: Ha cambiado el sentido de contaje
CmpResult1	OUT	Bool	Bit de estado: CurrentCount es igual al evento Reference1 ocurrido
CmpResult2	OUT	Bool	Bit de estado: CurrentCount es igual al evento Reference2 ocurrido
OverflowNeg	OUT	Bool	Bit de estado: CurrentCount ha alcanzado el límite inferior
OverflowPos	OUT	Bool	Bit de estado: CurrentCount ha alcanzado el límite superior

Elemento de estructura	Declaración	Tipo de datos	Descripción
EnHSC	IN	Bool	Cuando es True activa el HSC para contar impulsos; cuando es False desactiva el contaje
EnCapture	IN	Bool	Cuando es True activa la entrada de captura; cuando es False la entrada de captura no tiene efecto alguno
EnSync	IN	Bool	Cuando es True activa la entrada de sincronización; cuando es False la entrada de sincronización no tiene efecto alguno
EnDir	IN	Bool	Activa el valor NewDirection para que sea efectivo
EnCV	IN	Bool	Activa el valor NewCurrentCount para que sea efectivo
EnSV	IN	Bool	Activa el valor NewStartValue para que sea efectivo
EnReference1	IN	Bool	Activa el valor NewReference1 para que sea efectivo
EnReference2	IN	Bool	Activa el valor NewReference2 para que sea efectivo
EnUpperLmt	IN	Bool	Activa el valor NewUpperLimit para que sea efectivo
EnLowerLmt	IN	Bool	Activa el valor New_Lower_Limit para que sea efectivo
EnOpMode	IN	Bool	Activa el valor NewOpModeBehavior para que sea efectivo
EnLmtBehavior	IN	Bool	Activa el valor NewLimitBehavior para que sea efectivo
EnSyncBehavior	IN	Bool	Este valor no se utiliza.
NewDirection	IN	Int	Sentido de contaje: 1 = contaje ascendente; -1 = contaje descendente; todos los demás valores están reservados.
NewOpMode-Behavior	IN	Int	Funcionamiento del HSC en caso de desbordamiento 1 = el HSC deja de contar (el HSC debe desactivarse y volver a activarse para seguir contando); 2 = el HSC sigue funcionando; todos los demás valores están reservados.
NewLimitBehavior	IN	Int	Resultado del valor de CurrentCount en caso de desbordamiento: 1 = poner CurrentCount al límite opuesto; 2 = poner CurrentCount a StartValue; todos los demás valores están reservados.
NewSyncBehavior	IN	Int	Este valor no se utiliza.
NewCurrentCount	IN	Dint	Valor de contaje actual
NewStartValue	IN	Dint	Valor de arranque: valor inicial del HSC
NewReference1	IN	Dint	Valor de Reference1
NewReference2	IN	Dint	Valor de Reference2
NewUpperLimit	IN	Dint	Valor del límite superior de contaje
New_Lower_Limit	IN	Dint	Valor del límite inferior de contaje

SDT: HSC_Period

El tipo de datos "HSC_Period" equivale a un HSC configurado para el modo de período. La instrucción CTRL_HSC_EXT permite al programa acceder a un número determinado de impulsos de entrada durante un intervalo de medición específico. Con esta instrucción se calcula el período de tiempo entre impulsos de entrada con una resolución muy alta en el ámbito de los nanosegundos.

Tabla 10-5 Estructura de HSC_Period

Elemento de estructura	Declaración	Tipo de datos	Descripción
ElapsedTime	OUT	UDInt	Ver la descripción más abajo.
EdgeCount	OUT	UDInt	Ver la descripción más abajo.
EnHSC	IN	Bool	Cuando es True activa el HSC para medir períodos; cuando es False desactiva la medición de períodos.
EnPeriod	IN	Bool	Activa el valor NewPeriod para que sea efectivo
NewPeriod	IN	Int	Especifica el intervalo de medición en milisegundos. Los únicos valores permitidos son 10, 100 o 1000 ms.

ElapsedTime devuelve el tiempo en nanosegundos que transcurre entre los últimos eventos de contaje de intervalos de medición secuenciales. Si no se produce ningún evento de contaje durante un intervalo de medición, ElapsedTime devuelve el tiempo acumulado desde el último evento de contaje. ElapsedTime tiene un rango entre 0 y 4.294.967.280 nanosegundos (0x0000 0000 a 0xFFFF FFF0). El valor de retorno 4.294.967.295 (0xFFFF FFFF) indica que se ha producido un desbordamiento del período. El desbordamiento señaliza que el tiempo entre flancos de impulso es mayor que 4,295 segundos y que no es posible calcular el período con esta instrucción. Los valores de 0xFFFF FFF1 a 0xFFFF FFFE están reservados.

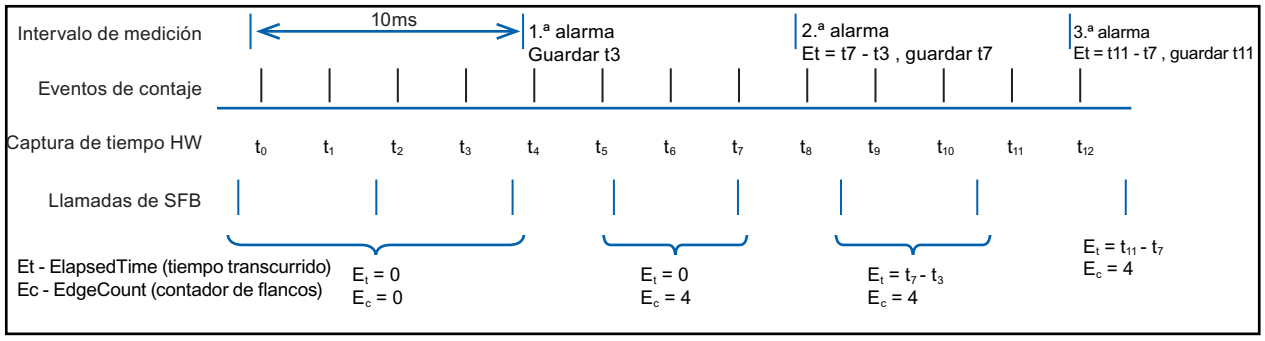
EdgeCount devuelve el número de eventos de contaje que se reciben durante el intervalo de medición. El período solo puede calcularse cuando el valor de EdgeCount es mayor que cero. Si ElapsedTime es "0" (no se han recibido impulsos de entrada) o 0xFFFF FFFF (desbordamiento del período), EdgeCount no es válido.

Cuando EdgeCount sea válido, utilice la fórmula siguiente para calcular el período en nanosegundos: Período = ElapsedTime/EdgeCount

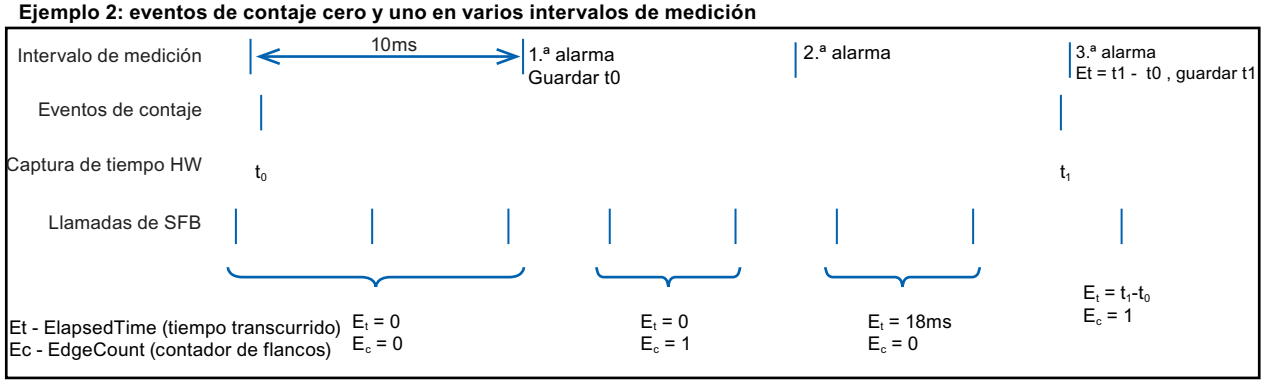
El valor calculado para el período de tiempo es un promedio de los períodos de tiempo de todos los impulsos que se producen durante el intervalo de medición. Si el período de un impulso de entrada es mayor que el intervalo de medición (10, 100 o 1000 ms), el cálculo del período requiere varios intervalos de medición.

Los ejemplos siguientes muestran cómo la instrucción mide períodos:

Ejemplo 1: varios eventos de contaje en un intervalo de medición



Ejemplo 2: eventos de contaje cero y uno en varios intervalos de medición



Reglas:
1. Si E1t = 0, el período no es válido
2. Si no, período = Et / Ec

SDT: HSC_Frequency

El tipo de datos "HSC_Frequency" equivale a un HSC configurado para el modo de frecuencia. La instrucción CTRL_HSC_EXT permite al programa acceder a la frecuencia de impulsos de entrada, medida durante un período específico.

El modo de frecuencia de la instrucción CTRL_HSC_EXT ofrece las siguientes funcionalidades:

Tabla 10-6 Estructura de HSC_Frequency

Elemento de estructura	Declaración	Tipo de datos	Descripción
Frecuencia	OUT	DInt	Devuelve una frecuencia en Hz, medida durante el intervalo de medición. Cuando el HSC cuenta en sentido descendente, la instrucción devuelve una frecuencia negativa.
EnHSC	IN	Bool	Cuando es True activa el HSC para medir la frecuencia; cuando es False desactiva la medición de la frecuencia.
EnPeriod	IN	Bool	Activa el valor NewPeriod para que sea efectivo
NewPeriod	IN	Int	Especifica el intervalo de medición en milisegundos. Los únicos valores permitidos son 10, 100 o 1000 ms.

La instrucción CTRL_HSC_EXT mide la frecuencia utilizando la misma técnica de medición que el modo de período para buscar ElapsedTime y EdgeCount. La instrucción calcula la frecuencia

en forma de valor entero con signo en Hz de acuerdo con la fórmula siguiente: Frecuencia = EdgeCount/ElapsedTime

Si se necesita un valor en coma flotante para la frecuencia, la fórmula anterior puede utilizarse para la frecuencia cuando el HSC está en modo de período. Tenga en cuenta que, en modo de período, ElapsedTime se devuelve en nanosegundos y puede ser necesario escalar el valor.

10.1.2 Funcionamiento del contador rápido

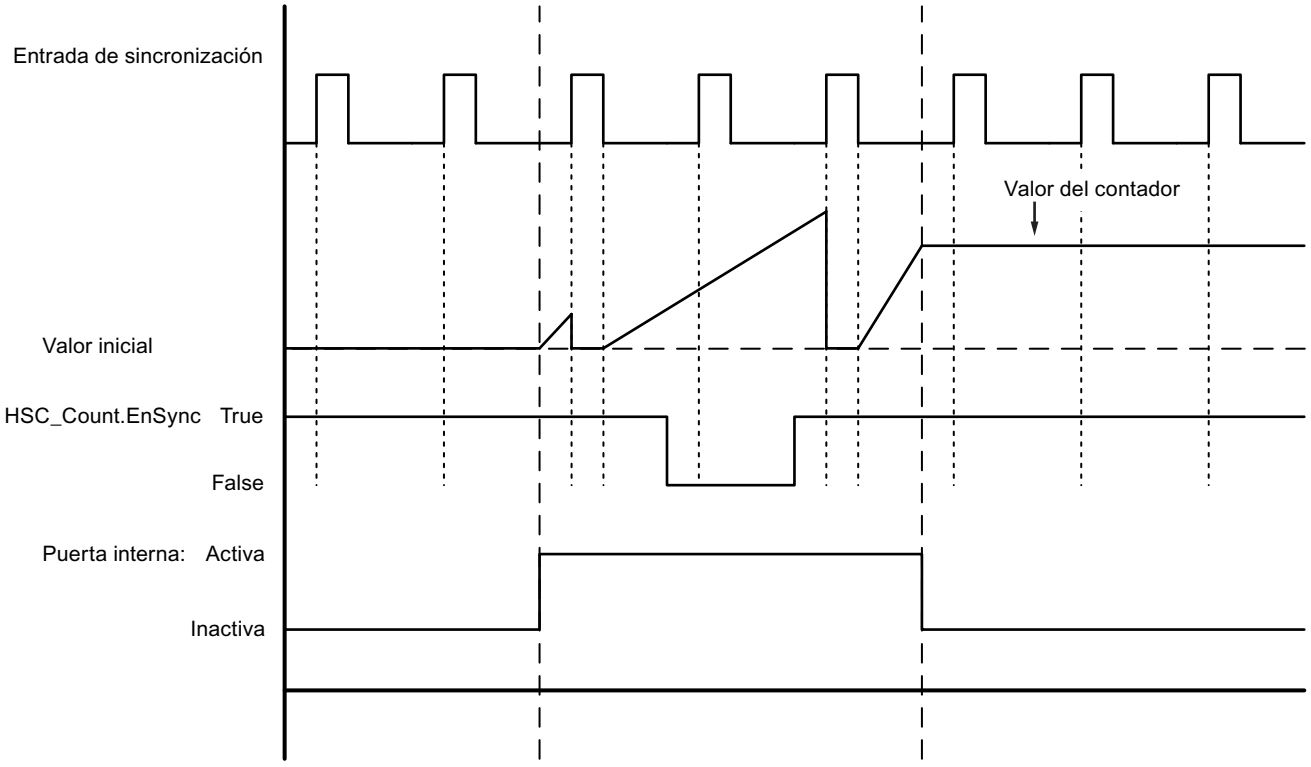
10.1.2.1 Función de sincronización

La función de sincronización (Sync) se utiliza para poner el contador al valor inicial con una señal de entrada externa. El valor inicial puede modificarse ejecutando la instrucción CTRL_HSC_EXT. Esto permite al usuario sincronizar el valor de contaje actual a una consigna cuando se produce una señal de entrada externa.

La sincronización siempre tiene lugar cuando se produce una señal de entrada y es efectiva independientemente del estado de la puerta interna. El bit "HSC_Count.EnSync" debe ponerse a True para activar la función de sincronización.

La instrucción CTRL_HSC_EXT pone el bit de estado HSC_Count.SyncActive a True una vez ha finalizado la sincronización. La instrucción CTRL_HSC_EXT pone el bit de estado HSC_Count.SyncActive a False si la sincronización no se ha producido desde la última vez que se ejecutó la instrucción.

La figura siguiente muestra un ejemplo de sincronización cuando la señal de entrada está configurada para una nivel "high" activo:



Nota

Los filtros de entrada configurados retardan la señal de control de la entrada digital.

Esta función de entrada solo está disponible cuando el HSC está configurado para el modo de contaje.

En Funciones de entrada (Página 561) encontrará información sobre cómo configurar la función de sincronización.

10.1.2.2 Función de puerta

Muchas aplicaciones requieren que los procesos de contaje se inicien o detengan en función de otros eventos. En estos casos, el contaje se inicia y detiene utilizando la función de puerta interna. Cada canal del HSC tiene dos puertas: una puerta software y una puerta hardware. El estado de estas puertas determina el estado de la puerta interna. Consulte la tabla inferior.

La puerta interna está abierta si las puertas software y hardware están abiertas las dos o no están configuradas. Si la puerta interna está abierta, se inicia el contaje. Si la puerta interna está cerrada, todos los demás impulsos de contaje se ignoran y el contaje se detiene.

Tabla 10-7 Estados de la función de puerta

Puerta hardware	Puerta software	Puerta interna
Abierta/no configurada	Abierta	Abierta
Abierta/no configurada	Cerrada	Cerrada

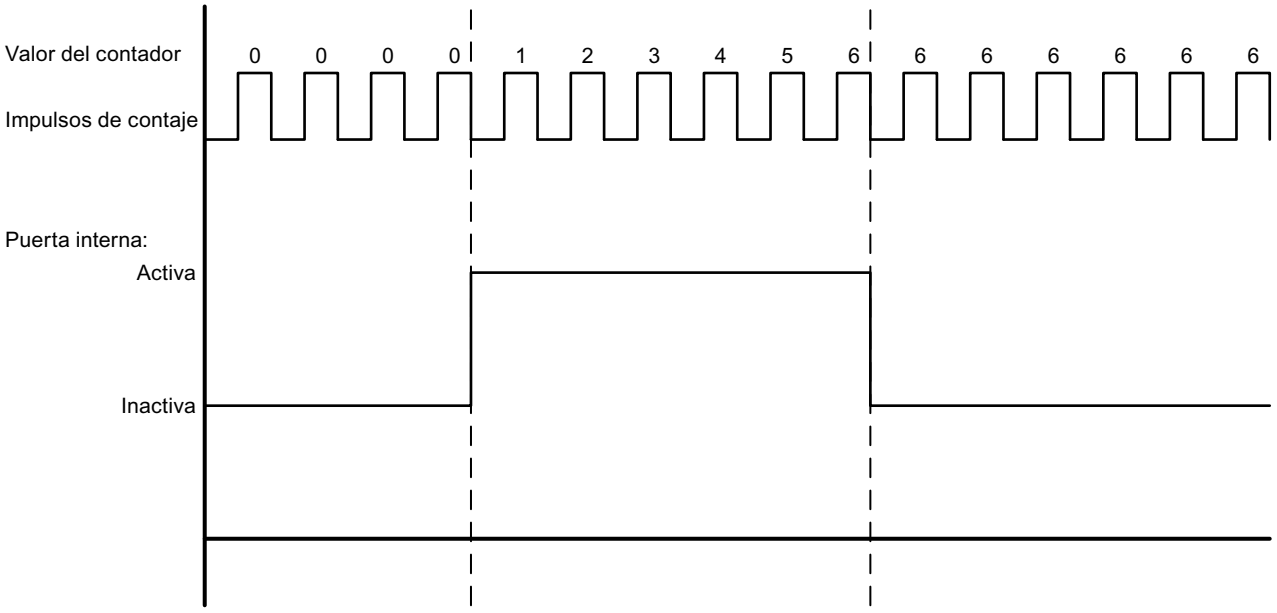
Puerta hardware	Puerta software	Puerta interna
Cerrada	Abierta	Cerrada
Cerrada	Cerrada	Cerrada

El concepto "abierta" significa que la puerta está activa. De forma análoga, el concepto "cerrada" significa que la puerta está inactiva.

La puerta software se controla con el bit de activación "HSC_Count.EnHSC" del SDT anexado a la instrucción CTRL_HSC_EXT. Para abrir la puerta software, ponga el bit "HSC_Count.EnHSC" a True; para cerrar la puerta software, ponga el bit "HSC_Count.EnHSC" a False. Ejecute la instrucción CTRL_HSC_EXT para actualizar el estado de la puerta software.

La puerta hardware es opcional y puede activarse o desactivarse en la sección de propiedades del HSC. Para controlar un proceso de contaje únicamente con la puerta hardware, la puerta software debe permanecer abierta. Si no se configura una puerta hardware, se considera que siempre está abierta y el estado de la puerta interna es el mismo que el de la puerta software.

La figura inferior muestra un ejemplo de cómo la puerta hardware se abre y se cierra con una entrada digital. La entrada digital está configurada para una nivel "high" activo:



Nota

Los filtros de entrada configurados retardan la señal de control de la entrada digital.

La función de puerta hardware solo está disponible cuando el HSC está configurado para el modo de contaje. En los modos de período y frecuencia, el estado de la puerta interna es el mismo que el de la puerta software.

En el modo de período, la puerta software se controla mediante "HSC_Period.EnHSC".

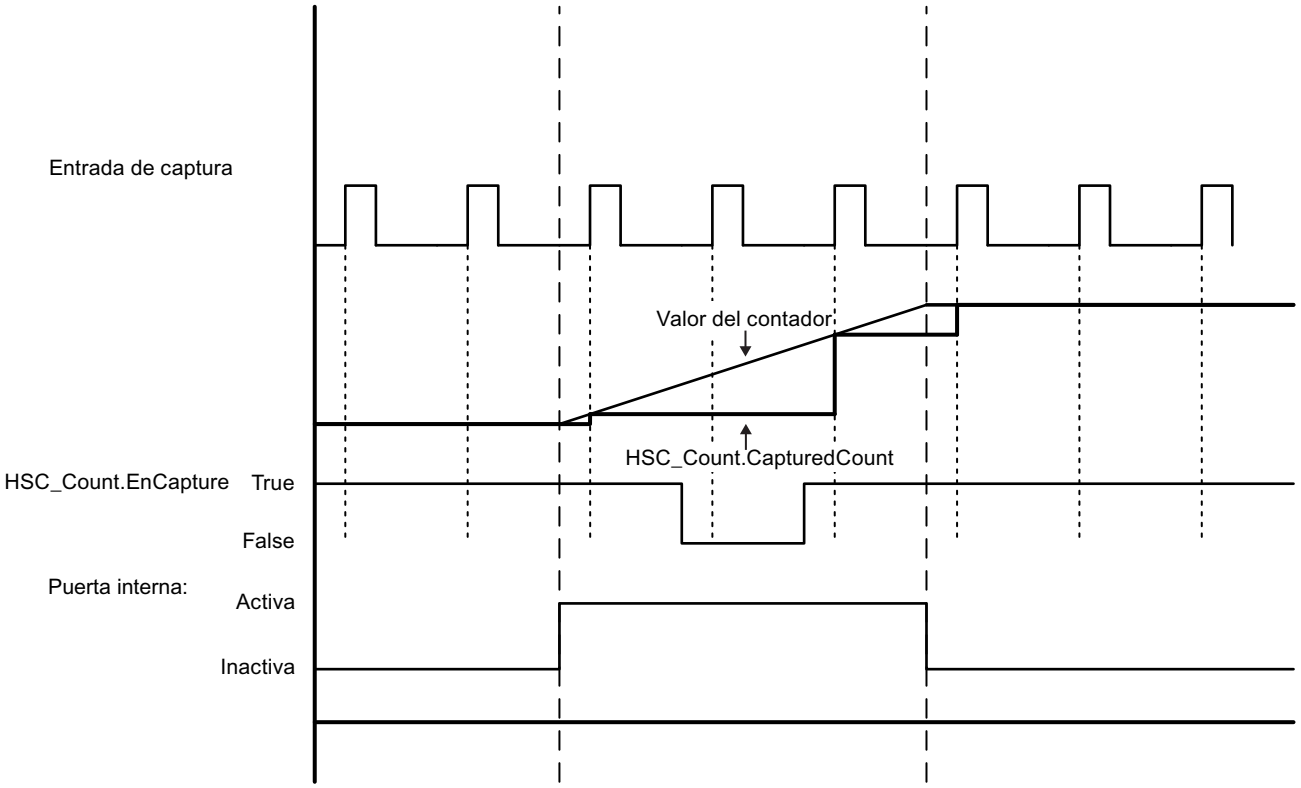
En el modo de frecuencia, la puerta software se controla mediante "HSC_Frequency.EnHSC".

En Funciones de entrada (Página 561) encontrará información sobre cómo configurar la función de puerta.

10.1.2.3 Función de captura

La función de captura se utiliza para guardar el valor actual del contador con una señal de referencia externa. Cuando está configurada y activada mediante el bit "HSC_Count.EnCapture", la función de captura provoca la captura del contaje actual cuando se produce un flanco de entrada externa. La función de captura es efectiva independientemente del estado de la puerta interna. El programa guarda el valor del contador sin cambios cuando la puerta está cerrada. Después de ejecutar la instrucción CTRL_HSC_EXT, el programa guarda el valor capturado en "HSC_Count.CapturedCount".

La figura inferior muestra un ejemplo de la función de captura configurada para capturar un flanco ascendente. La entrada de captura no dispara una captura del contaje actual cuando el bit "HSC_Count.EnCapture" está puesto a False mientras se ejecuta la instrucción CTRL_HSC_EXT.



Nota

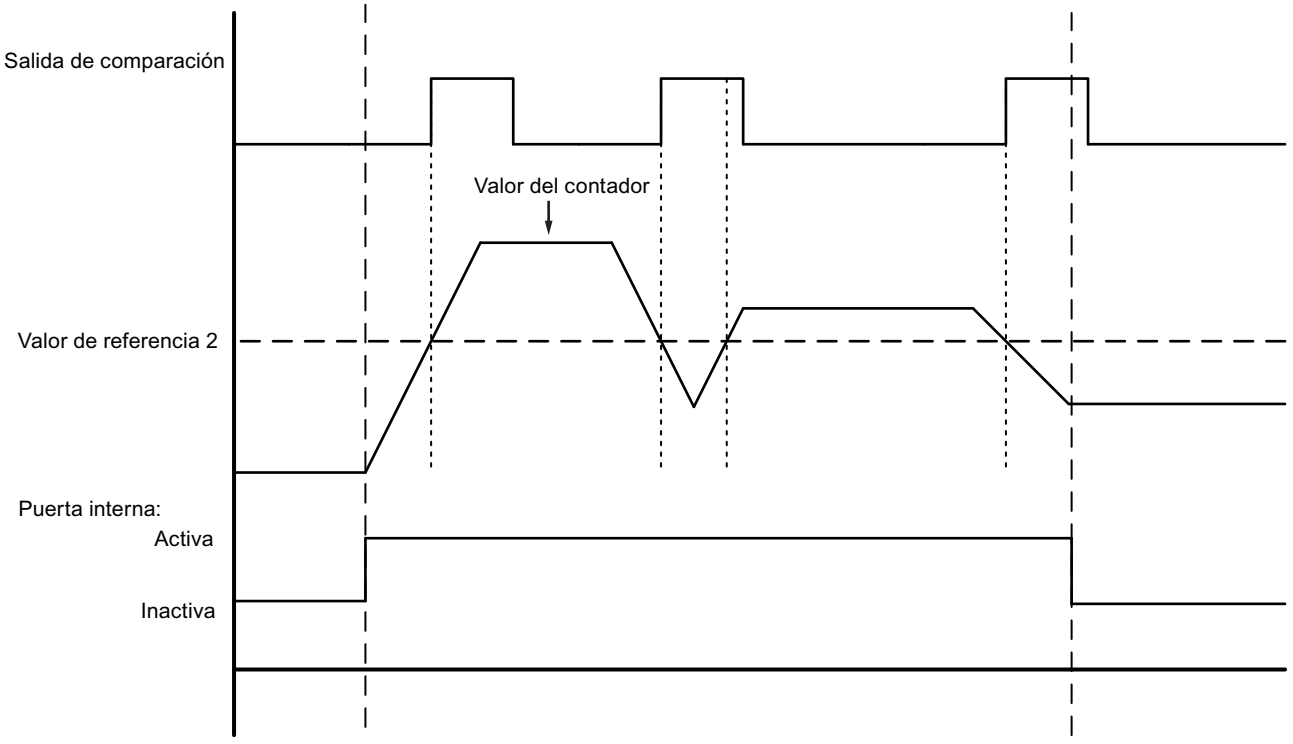
Los filtros de entrada configurados retardan la señal de control de la entrada digital.

Esta función de entrada solo puede utilizarse cuando el HSC está configurado para el modo de contaje.

En Funciones de entrada (Página 561) encontrará información sobre cómo configurar la función de captura.

10.1.2.4 Función de comparación

Cuando está activada, la función de salida de comparación genera un impulso individual configurable cada vez que se produce el evento configurado. Los eventos incluyen un contaje igual a uno de los valores de referencia o los desbordamientos del contador. Si hay un impulso en curso y el evento se repite, no se genera ningún impulso para dicho evento.



Nota

Esta función de salida solo puede utilizarse cuando el HSC está configurado para el modo de contaje.

En Función de salida (Página 562) encontrará información sobre cómo configurar la función de comparación.

10.1.2.5 Aplicaciones

Una aplicación típica consiste en utilizar el HSC para supervisar las respuestas de un encóder rotativo incremental. El encóder rotativo ofrece un número determinado de contajes por revolución que pueden utilizarse como entrada de reloj para el HSC. También hay un impulso de reset que ocurre una vez por revolución y que puede utilizarse como entrada de sincronización para el HSC.

Al arrancar, el programa de usuario debe cargar el valor de referencia inicial en el HSC y poner las salidas a sus estados iniciales. Las salidas permanecen en su estado mientras el contaje actual sea menor que el valor de referencia. El HSC emite una alarma cuando el contaje actual es igual al valor de referencia, al ocurrir un evento de sincronización (reset) y también al producirse un cambio de sentido.

Cuando todos los valores del contador son iguales al valor de referencia, se dispara un evento de alarma. Dentro del OB de alarma, el programa de usuario debe cargar el siguiente valor de referencia en el HSC y poner las salidas a su próximo estado.

Cuando se dispara la entrada de sincronización, el valor de contaje actual se pone al valor inicial y se genera un evento de alarma. Dentro de este OB de alarma, el programa de usuario debe cargar el valor de referencia inicial en el HSC y poner las salidas a su estado inicial. En este momento, el HSC ha regresado a su estado inicial y el ciclo se repite mientras el HSC sigue contando.

Puesto que las alarmas ocurren con una frecuencia mucho menor que la frecuencia de contaje del HSC, es posible implementar un control preciso de las operaciones rápidas con un impacto relativamente bajo en el ciclo de la CPU. El método de asociar alarmas permite cargar cada valor predeterminado nuevo en un subprograma por separado, lo que simplifica el control del estado. Alternativamente, todos los eventos de alarma se pueden ejecutar en un solo subprograma.

La función de puerta, que se dispara desde el programa de usuario o mediante una señal de entrada externa, puede desactivar el contaje de los impulsos del encóder. Es posible ignorar los movimientos del eje desactivando la puerta. Esto significa que, aunque el encóder siga enviando impulsos al HSC, el valor de contaje se mantiene al valor que tenía antes de desactivar la puerta. Cuando la puerta se activa, el contaje se reinicia a partir del valor que tenía antes de que se desactivara la puerta.

Cuando está activada, la función de captura provoca la captura del contaje actual cuando se produce una entrada externa. Un proceso (p. ej. un subprograma de calibración) puede utilizar esta función para determinar los impulsos que se producen entre eventos.

Cuando está activada, la función de salida de comparación genera un impulso individual configurable cada vez que el contaje actual alcanza uno de los valores de referencia o desbordamientos (excede los límites de contaje). Este impulso puede utilizarse como señal para iniciar otro proceso siempre que se produzca un evento concreto del HSC.

El sentido de contaje se controla desde el programa de usuario o mediante una señal de entrada externa.

Si se configura el HSC para el modo de frecuencia se obtiene la velocidad del eje rotativo. Esta función proporciona un valor entero con signo en unidades de Hz. Puesto que la señal de reset se produce una vez por revolución, al medir la frecuencia de la señal de reset se obtiene una indicación rápida de la velocidad del eje en revoluciones por segundo.

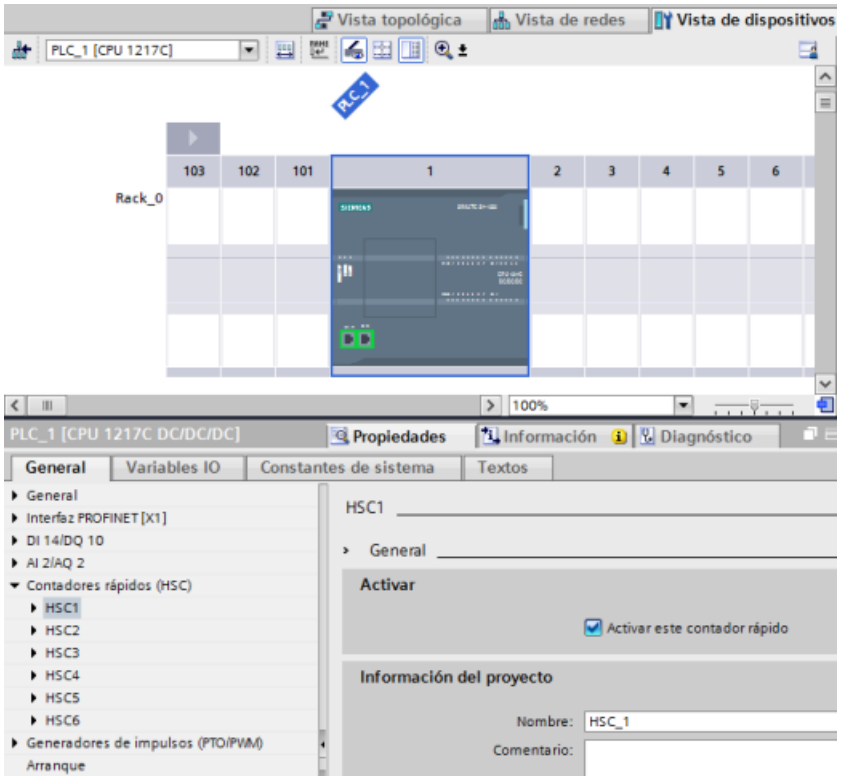
Si desea obtener un valor en coma flotante de la frecuencia, configure el HSC para el modo de período. Puede utilizar los valores de ElapsedTime y EdgeCount devueltos en el modo de período para calcular la frecuencia.

10.1.3 Configurar un contador rápido

Para configurar un contador rápido (HSC):

- Seleccione la configuración de dispositivos en el árbol del proyecto.
- Seleccione la CPU que desea configurar.
- Haga clic en la ficha Propiedades situada en la ventana de inspección (ver la figura inferior).
- Seleccione el HSC que desea activar en la lista que se muestra debajo de la ficha General (ver la figura inferior).

Puede configurar hasta seis contadores rápidos (del HSC1 al HSC6). Active un HSC seleccionando la opción "Habilitar este contador rápido". Si está activado, STEP 7 asigna un nombre predeterminado unívoco al HSC. El nombre puede cambiarse editándolo en el campo "Nombre:", pero debe ser un nombre unívoco. Los nombres de los HSC activados se convierten en variables con el tipo de datos "HW_Hsc" en la tabla de variables "Constantes de sistema" y están disponibles para ser utilizados como parámetro HSC de la instrucción CTRL_HSC_EXT. Encontrará más información en "Configurar el funcionamiento de la CPU (Página 153)":



Después de activar el HSC, STEP 7 establece el contaje de fase simple como configuración predeterminada. Una vez se ha establecido el filtro de entrada digital para la entrada de reloj del HSC, es posible descargar el programa en el PLC y entonces la CPU ya estará lista para contar. Para modificar la configuración del HSC, consulte el apartado siguiente "Tipo de contaje".

La tabla siguiente ofrece una relación de las entradas y salidas que están disponibles para cada configuración:

Tabla 10-8 Modos de contaje del HSC

Tipo	Entrada 1	Entrada 2	Entrada 3	Entrada 4	Entrada 5	Salida 1	Función
Fase simple con control interno del sentido	Reloj	-	-	-	-	-	Contaje, frecuencia o período
			Sincronización	Puerta	Captura	Comparación	Contaje

Tipo	Entrada 1	Entrada 2	Entrada 3	Entrada 4	Entrada 5	Salida 1	Función
Fase simple con control externo del sentido	Reloj	Sentido	-	-	-	-	Contaje, frecuencia o período
			Sincronización	Puerta	Captura	Comparación	Contaje
Dos fases	Reloj adelante	Reloj atrás	-	-	-	-	Contaje, frecuencia o período
			Sincronización	Puerta	Captura	Comparación	Contaje
Contador A/B	Fase A	Fase B	-	-	-	-	Contaje, frecuencia o período
			Sincronización[1]	Puerta	Captura	Comparación	Contaje
Contador A/B cuádruple	Fase A	Fase B	-	-	-	-	Contaje, frecuencia o período
			Sincronización[1]	Puerta	Captura	Comparación	Contaje

[1] Para un encóder: fase Z, inicio

10.1.3.1 Tipo de contaje

Hay cuatro tipos de contaje o modos. Cuando se cambia el modo también cambian las opciones de configuración disponibles para el HSC:

- Contaje: Cuenta el número de impulsos e incrementa o decrementa el valor de contaje, según sea el estado del control de sentido. Las entradas y salidas externas pueden inicializar el contaje, desactivarlo, iniciar una captura del contaje actual y generar un impulso individual para un evento concreto. Los valores de salida son el valor de contaje actual y el valor de contaje en el momento de producirse una captura.
- Período: Cuenta el número de impulsos de entrada durante un período de tiempo específico. Devuelve el contaje de impulsos y la duración en nanosegundos (ns). Los valores se capturan y calculan al final del período de tiempo especificado en el período de medición de frecuencia. El modo período está disponible para la instrucción CTRL_HSC_EXT, pero no para la instrucción CTRL_HSC.
- Frecuencia: Mide los impulsos de entrada y la duración y calcula la frecuencia de los impulsos. El programa devuelve la frecuencia en forma de entero doble con signo en unidades de Hz. El valor es negativo si el sentido de contaje es descendente. Los valores se capturan y calculan al final del período de tiempo especificado en el período de medición de frecuencia.
- Motion Control: Es utilizado por el objeto tecnológico Motion Control y no está disponible para las instrucciones HSC. Encontrará más información en "Motion Control".

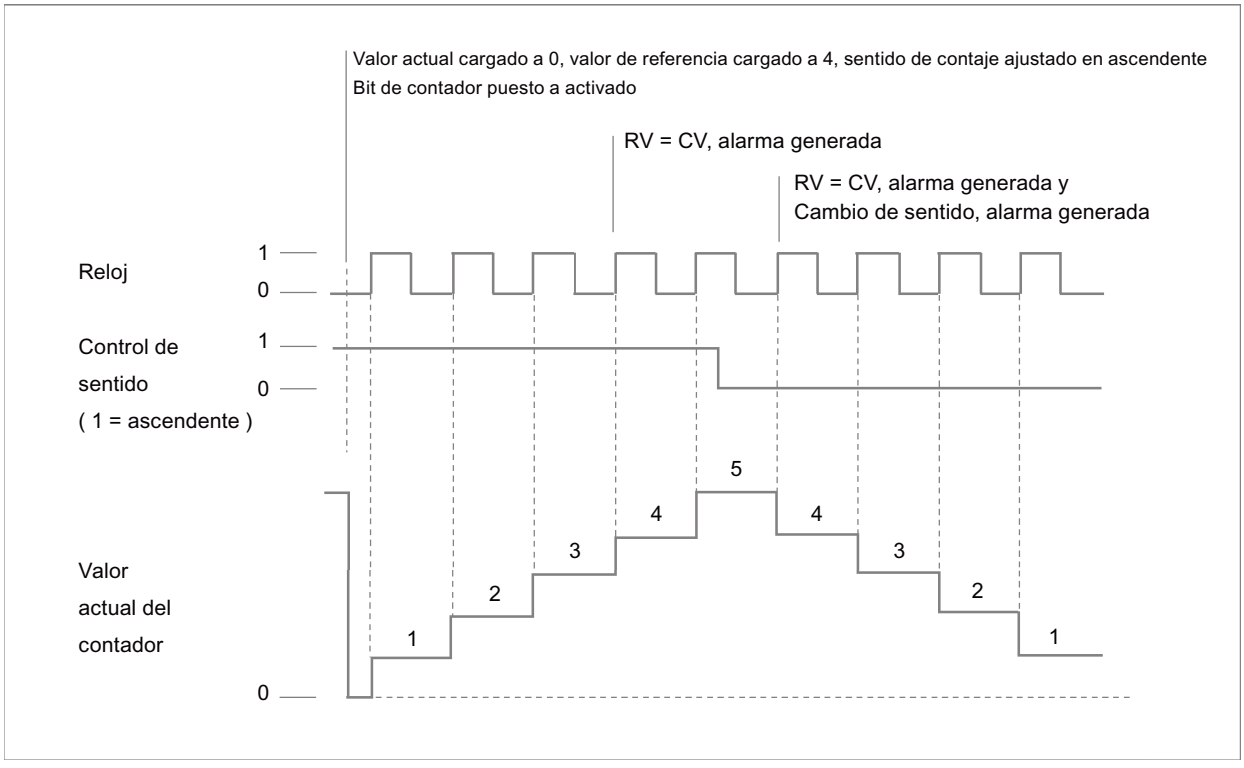
10.1.3.2 Fase de operación

Seleccione la fase de operación deseada del HSC. Las cuatro figuras inferiores muestran cuándo cambia el valor del contador, cuándo se produce el evento de que el valor actual (CV) es igual al valor de referencia (RV) y cuándo se produce el evento de cambio de sentido.

Fase simple

La fase simple (no disponible con Motion Control) cuenta impulsos:

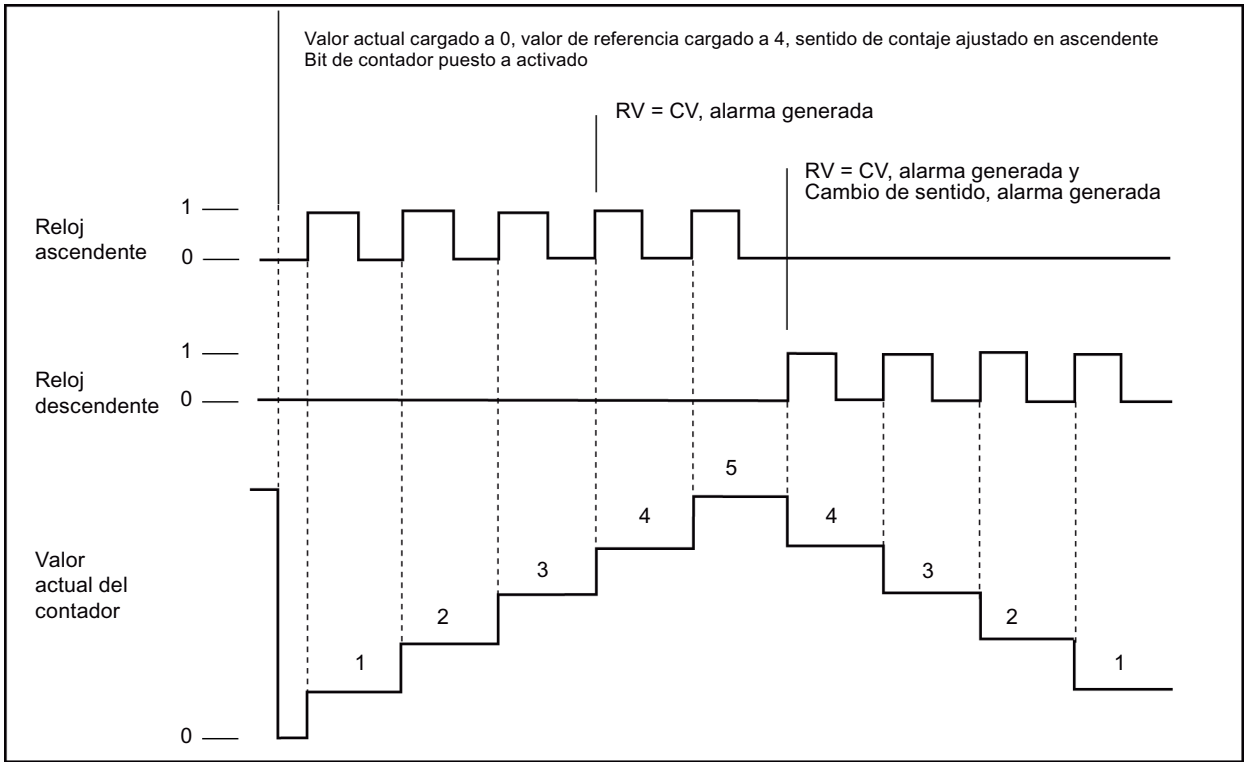
- Programa de usuario (control interno del sentido):
  - 1 es ascendente
  - -1 es descendente
- Entrada de hardware (control externo del sentido):
  - El nivel "high" es ascendente.
  - El nivel "low" es descendente.



Dos fases

Las dos fases cuentan:

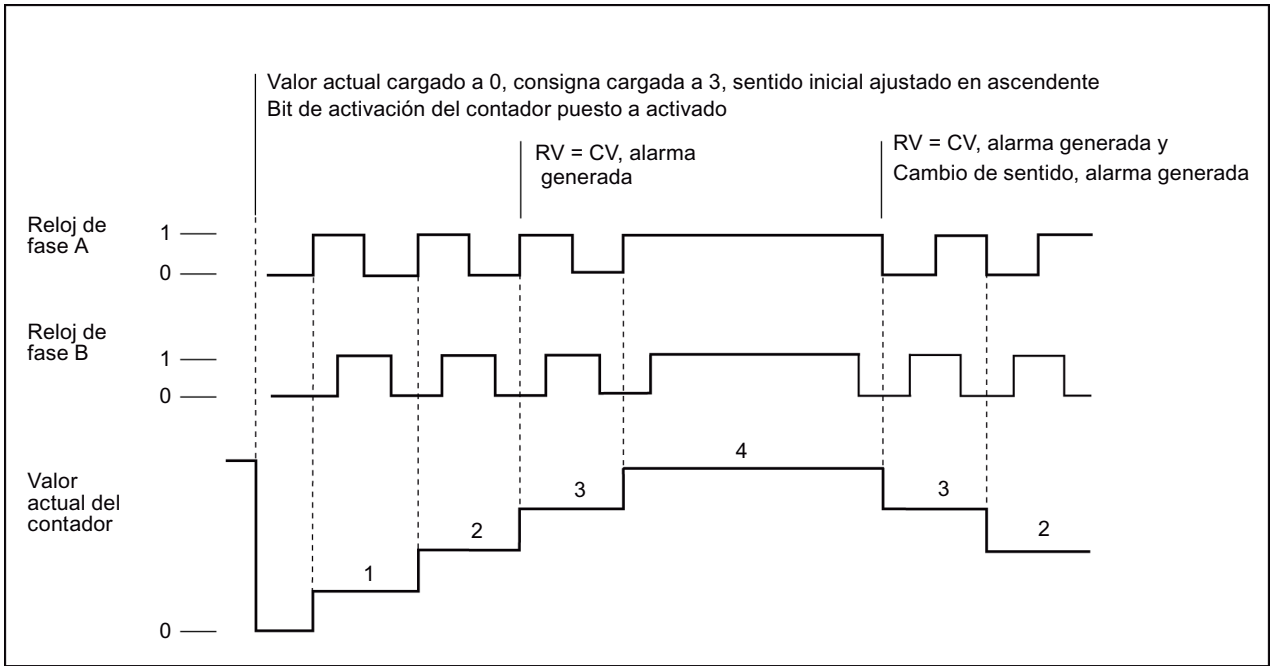
- ascendente en la entrada de reloj adelante
- descendente en la entrada de reloj atrás



Contador A/B

La fase A/B en cuadratura cuenta:

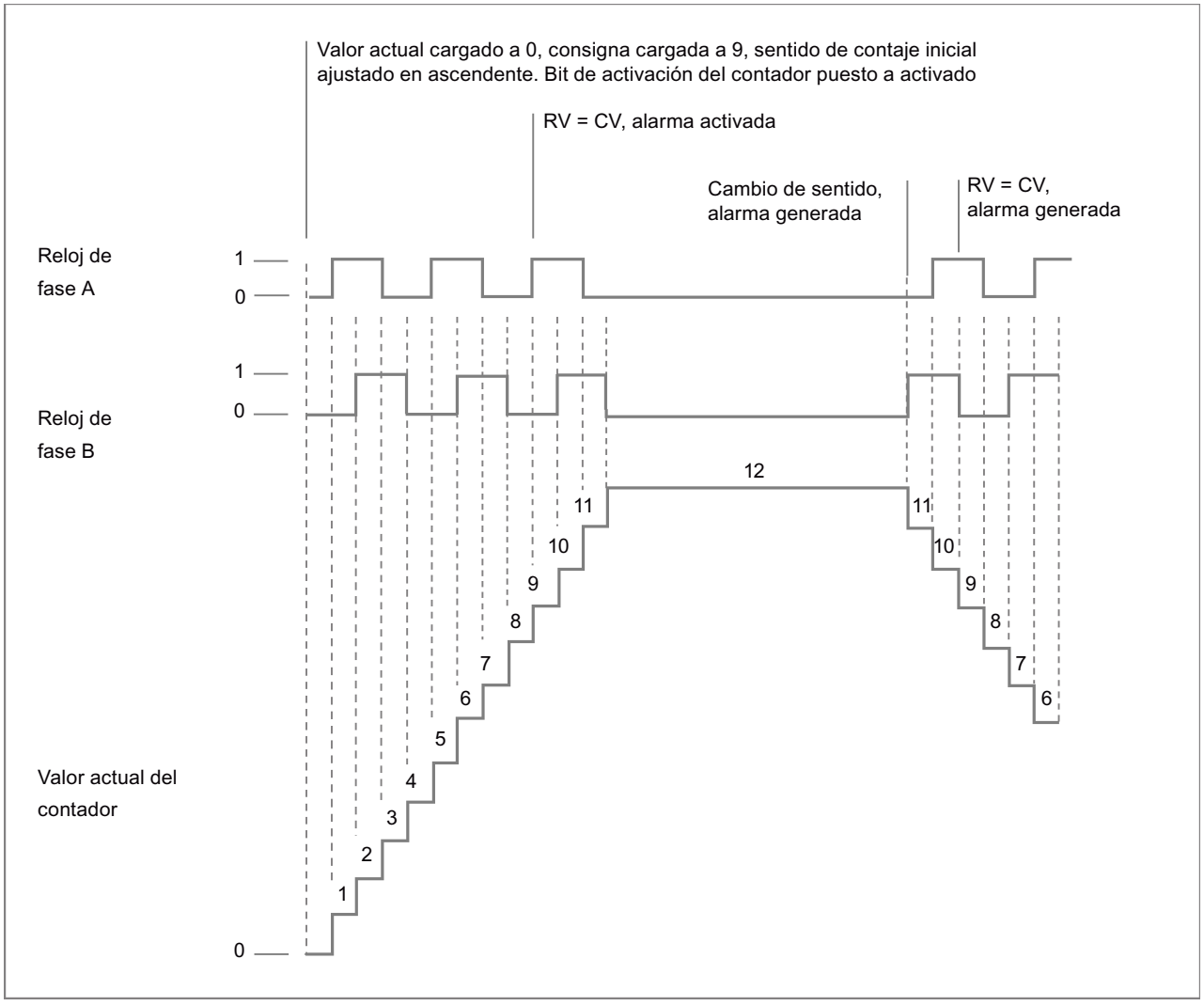
- ascendente con un flanco ascendente en la entrada de reloj A cuando la entrada de reloj B es "low"
- descendente con un flanco descendente en la entrada de reloj A cuando la entrada de reloj B es "low"



Contador A/B cuádruple

La fase A/B en cuadratura cuádruple cuenta:

- ascendente con un flanco ascendente en la entrada de reloj A cuando la entrada de reloj B es "low"
- ascendente con un flanco descendente en la entrada de reloj A cuando la entrada de reloj B es "high"
- ascendente con un flanco ascendente en la entrada de reloj B cuando la entrada de reloj A es "high"
- ascendente con un flanco descendente en la entrada de reloj B cuando la entrada de reloj A es "low"
- descendente con un flanco ascendente en la entrada de reloj B cuando la entrada de reloj A es "low"
- descendente con un flanco descendente en la entrada de reloj B cuando la entrada de reloj A es "high"
- descendente con un flanco ascendente en la entrada de reloj A cuando la entrada de reloj B es "high"
- descendente con un flanco descendente en la entrada de reloj A cuando la entrada de reloj B es "low"



10.1.3.3 Valores iniciales

Cada vez que la CPU pasa a RUN, carga los valores iniciales. Los valores iniciales solo se utilizan en modo de contaje:

- Valor inicial del contador: El programa pone el valor de contaje actual al valor inicial del contador cuando la CPU pasa de STOP a RUN o cuando el programa dispara la entrada de sincronización.
- Valor de referencia inicial Cuando el contaje actual alcanza el valor de referencia, si las funciones correspondientes están configuradas el programa genera una alarma y/o un impulso de salida.
- Valor de referencia inicial 2: Cuando el contaje actual alcanza el valor de referencia 2, si la función está configurada el programa genera un impulso de salida.

- Valor límite superior inicial: Valor de contaje máximo. El valor predeterminado es el valor más alto posible de +2.147.483.647 impulsos.
- Valor límite inferior inicial: Valor de contaje mínimo. El valor predeterminado es el valor más bajo posible de -2.147.483.648 impulsos.

Los valores anteriores y el comportamiento del contador cuando alcanza un límite solo están disponibles en modo de contaje. Los valores y el comportamiento se ajustan con la instrucción CTRL_HSC_EXT mediante el SDT HSC_Count.

10.1.3.4 Funciones de entrada

Las entradas de reloj y sentido determinan los eventos y el sentido de contaje, según sea la fase de operación. Solo es posible utilizar las entradas de sincronización, captura y puerta en modo de contaje; también pueden activarse y configurarse individualmente las entradas para diferentes tipos de disparos.

Entrada de sincronización

La entrada de sincronización (Sync) pone el valor de contaje actual al valor de arranque (o inicial) del contador. Por lo general, la entrada de sincronización se utiliza para poner el contador a "0". La sincronización puede dispararse cuando el pin de entrada tiene uno de los estados siguientes:

- High
- Low
- Pasa de "low" a "high"
- Pasa de "high" a "low"
- Pasa de "high" a "low" o de "low" a "high"

Entrada de captura

La entrada de captura pone el valor de contaje capturado al valor de contaje guardado en el momento de disparar la entrada de captura. La captura puede dispararse cuando el pin de entrada tiene uno de los estados siguientes:

- Pasa de "low" a "high"
- Pasa de "high" a "low"
- Pasa de "high" a "low" o de "low" a "high"

Entrada de puerta

La entrada de puerta detiene el contaje del HSC. La puerta puede dispararse cuando el pin de entrada tiene uno de los estados siguientes:

- High
- Low

10.1.3.5 Función de salida

La función de salida de comparación es la única salida del HSC y solo está disponible en modo de contaje.

Salida de comparación

La salida de comparación puede configurarse para que genere un impulso individual cuando se produzca uno de los eventos siguientes:

- El contador es igual al valor de referencia (sentido de contaje ascendente)
- El contador es igual al valor de referencia (sentido de contaje descendente)
- El contador es igual al valor de referencia (sentido de contaje ascendente o descendente)
- El contador es igual al valor de referencia 2 (sentido de contaje ascendente)
- El contador es igual al valor de referencia 2 (sentido de contaje descendente)
- El contador es igual al valor de referencia 2 (sentido de contaje ascendente o descendente)
- Desbordamiento positivo
- Desbordamiento negativo

El impulso de salida puede configurarse con un tiempo de ciclo de entre 1 y 500 ms; el tiempo de ciclo predeterminado es de 10 ms. Es posible establecer la duración de impulso, o ciclo de trabajo, en un valor cualquiera entre 1 y 100%; la duración de impulso predeterminada es del 50%.

Si se producen varios eventos de salida de comparación dentro del tiempo de ciclo especificado, los impulsos de salida de dichos eventos se pierden debido a que el impulso actual todavía no ha finalizado su ciclo. Una vez ha finalizado el impulso (ha transcurrido el tiempo de ciclo configurado), el generador de impulsos está disponible para generar un impulso nuevo.

10.1.3.6 Eventos de alarma

En la sección de configuración de eventos existe la posibilidad de seleccionar un OB de alarma de proceso en el menú desplegable (o bien crear un OB nuevo) y asignarlo a un evento del HSC. La prioridad de la alarma está comprendida en un rango entre 2 y 26, siendo 2 la prioridad más baja y 26 la más alta. En función de la configuración del HSC estarán disponibles los eventos siguientes:

- Evento de que el valor del contador es igual al valor de referencia: Se produce un evento de que el valor del contador es igual al valor de referencia cuando el valor de contaje del HSC alcanza el valor de referencia. El valor de referencia se establece durante la configuración en la sección Valor de referencia inicial, o bien al actualizar "NewReference1" con la instrucción CTRL_HSC_EXT. Encontrará más información en el apartado "Fase de operación (Página 557)".
- Evento de sincronización: Se produce una sincronización (Sync) siempre que se activa y dispara la entrada de sincronización.
- Evento de cambio de sentido: Se produce un evento de cambio de sentido cuando cambia el sentido de contaje. Encontrará más información en el apartado "Fase de operación (Página 557)".

10.1.3.7 Asignación de los pines de entrada de hardware

Para cada entrada del HSC que se active hay que seleccionar el punto de entrada deseado, que se encuentra en la CPU o en una Signal Board opcional (los módulos de comunicación y señales no soportan las entradas del HSC). Cuando se selecciona un punto, STEP 7 muestra el valor de frecuencia máximo junto a la selección. Posiblemente sea necesario ajustar el filtro de entrada digital para que todas las frecuencias de señales válidas puedan pasar el filtro. Para configurar los filtros de entrada del HSC, consulte "Configurar los tiempos de filtrado de entrada digital (Página 155)".

Nota

Los canales de entrada de CPU y SB (firmware V4 o posterior) tienen tiempos de filtrado de entrada configurables.

Las versiones de firmware anteriores tenían canales de entrada HSC y tiempos de filtrado fijos que no podían cambiarse.

Las versiones V4 o posteriores permiten asignar canales de entrada y tiempos de filtrado. El ajuste predeterminado del filtro de entrada es de 6,4 ms, que limita la frecuencia de contaje máxima a 78 Hz. Es posible modificar los ajustes de filtrado para contar frecuencias más altas o más bajas en función del diseño del sistema.

⚠ ADVERTENCIA

Riesgos asociados con los cambios en el ajuste de tiempos de filtrado de los canales de entradas digitales

Si se reajusta el tiempo de filtrado de un canal de entrada digital, puede que sea necesario presentar un nuevo valor de entrada de nivel "0" durante un tiempo acumulado de 20 ms para que el filtro esté totalmente operativo ante nuevas entradas. Durante este tiempo, puede que no se detecten o no se cuenten los eventos de impulso breve "0" cuya duración sea inferior a 20 ms.

El cambio de los tiempos de filtrado puede originar un funcionamiento inesperado de los equipos o del proceso, lo que puede causar la muerte o lesiones graves al personal o daños a los equipos.

Para asegurar que un tiempo de filtrado nuevo tenga efecto inmediato, desconecte y vuelva a conectar la CPU.

Utilice la tabla siguiente para garantizar que los canales de entrada de CPU y SB conectados soportan las frecuencias de impulso máximas de las señales del proceso:

Tabla 10-9 Entrada de CPU: frecuencia máxima

CPU	Canal de entrada de CPU	Fase de operación: fase simple o dos fases	Fase de operación: contador A/B o contador A/B cuádruple
1211C	De Ia.0 a Ia.5	100 kHz	80 kHz
1212C	De Ia.0 a Ia.5	100 kHz	80 kHz
	Ia.6, Ia.7	30 kHz	20 kHz
1214C y 1215C	De Ia.0 a Ia.5	100 kHz	80 kHz
	De Ia.6 a Ib.5	30 kHz	20 kHz
1217C	De Ia.0 a Ia.5	100 kHz	80 kHz
	De Ia.6 a Ib.1	30 kHz	20 kHz
	De Ib.2 a Ib.5 (De 0,2+, 0,2- a 0,5+, 0,5-)	1 MHz	1 MHz

Tabla 10-10 Entrada de Signal Board SB: frecuencia máxima (tarjeta opcional)

Signal Board (SB)	Canal de entrada de SB	Fase de operación: fase simple o dos fases	Fase de operación: contador A/B o contador A/B cuádruple
SB 1221, 200 kHz	De Ie.0 a Ie.3	200 kHz	160 kHz
SB 1223, 200 kHz	Ie.0, Ie.1	200 kHz	160 kHz
SB 1223	Ie.0, Ie.1	30 kHz	20 kHz

Cuando se asigna un punto de entrada a una función HSC es posible asignar el mismo punto de entrada a varias funciones HSC. Así, por ejemplo, está permitido asignar I0.3 a la entrada de sincronización de HSC1 y a la entrada de sincronización de HSC2 con el fin de sincronizar el contaje de ambos HSC al mismo tiempo; sin embargo, se generará una advertencia de compilador.

Dentro de lo posible, evite la asignación de múltiples funciones de entrada del mismo HSC al mismo punto de entrada. Así, por ejemplo, también está permitido asignar I0.3 a la entrada de sincronización y a la entrada de puerta de HSC1 con el fin de sincronizar el contaje y desactivar

el contaje al mismo tiempo. Esta configuración es posible, pero podría provocar resultados no deseados.

⚠ ADVERTENCIA

Riesgos asociados con la asignación de varias funciones a un solo canal de entrada digital

La asignación de varias funciones de entrada del mismo HSC a un punto de entrada común puede provocar resultados impredecibles. Cuando se produce un disparo en un punto con varias funciones asignadas a dicho disparo, no es posible saber el orden en el que el PLC ejecuta las funciones. Esto se conoce como condición de carrera y suele ser una situación no deseable.

Esta condición de carrera puede originar un funcionamiento inesperado de los equipos o del proceso, lo que puede causar la muerte o lesiones graves al personal o daños a los equipos.

Para evitar una condición de carrera, no asigne más de dos funciones de entrada del mismo HSC al mismo pin de entrada. Si un HSC tiene dos funciones de entrada asignadas al mismo pin, establezca los disparos de forma que nunca puedan darse al mismo tiempo. Recuerde que un flanco descendente se produce en la misma instancia que empieza un nivel "low" y que un flanco ascendente se produce en la misma instancia que empieza un nivel "high".

Nota

Los puntos de entrada y salida digitales utilizados por los contadores rápidos (HSC) se asignan durante la configuración de la CPU. Cuando se asignan puntos de entrada y salida a dispositivos HSC, no es posible modificar los valores de dichos puntos utilizando la función de forzado permanente en una tabla de observación. El HSC controla por completo estos puntos de entrada y salida.

10.1.3.8 Asignación de los pines de salida de hardware

Si se activa la salida de comparación debe seleccionarse un punto de salida disponible. Una vez se ha configurado un punto de salida para utilizar con un HSC (u otros objetos tecnológicos como un generador de impulsos), dicho punto de salida es propiedad exclusiva del objeto en cuestión. Ningún otro componente puede utilizar el punto de salida y este no puede forzarse a un valor concreto. Si se configura un canal de salida individual para varios HSC o para utilizar en un HSC y una salida de impulsos, el programa genera un error de compilador.

10.1.3.9 Direcciones de la memoria de entrada del HSC

Todo HSC utiliza una sección de palabra doble de la memoria de entrada que guarda el contaje actual. Si se configura el HSC para frecuencia, esta se guarda en la ubicación mencionada de la memoria de entrada. El rango disponible para direcciones de entrada es de I0.0 a I1023.7 (la dirección inicial máxima es I1020.0). El HSC no puede utilizar una dirección de entrada que se solape con una dirección de entrada asignada a otro componente. Encontrará más información sobre la memoria imagen de proceso en "Ejecución del programa de usuario (Página 67)".

La tabla siguiente muestra las direcciones predeterminadas asignadas para cada HSC:

Tabla 10-11 Direcciones predeterminadas del HSC

Contador rápido (HSC)	Tipos de datos de valores actuales	Dirección de valor actual predeterminada
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 Identificador de hardware

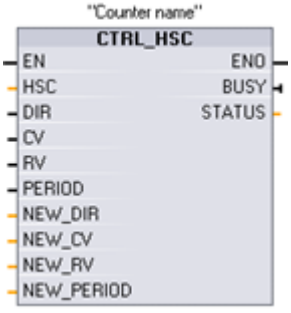
Cada HSC tiene un identificador de hardware único que utilizan las instrucciones HSC_CTRL y HSC_CTRL_EXT. Encontrará la variable PLC para el identificador de hardware en "Constantes de sistema". Un HSC con el nombre "HSC_1" tiene la variable "Local~HSC_1" y el tipo de datos "Hw_Hsc". Esta variable también se muestra en el menú desplegable cuando se selecciona la entrada HSC de las instrucciones CTRL_HSC_EXT.

10.1.4 Instrucción CTRL_HSC (controlar contadores rápidos) antigua

10.1.4.1 Descripción general de la instrucción

Tabla 10-12 Instrucción CTRL_HSC (para contaje general)

KOP / FUP	SCL	Descripción

	`"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_);`
		Toda instrucción CTRL_HSC (Controlar contadores rápidos) utiliza una estructura almacenada en un DB para conservar los datos de contador. El DB se asigna cuando la instrucción CTRL_HSC se coloca en el editor.

[1] Cuando se inserta la instrucción, STEP 7 muestra el cuadro de diálogo "Opciones de llamada" para crear el DB asociado.

[2] En el ejemplo SCL, "CTRL_HSC_1_DB" es el nombre del DB de instancia.

Tabla 10-13 Tipos de datos para los parámetros

Parámetro	Declaración	Tipo de datos	Descripción
HSC	IN	HW_HSC	Identificador del HSC
DIR[1, 2]	IN	Bool	1 = solicitar nuevo sentido de contaje
CV[1]	IN	Bool	1 = solicitar activación del nuevo valor del contador
RV[1]	IN	Bool	1 = solicitar activación del nuevo valor de referencia
PERIOD[1]	IN	Bool	1 = solicitar activación del nuevo periodo (solo para el modo de medición de frecuencia)
NEW_DIR	IN	Int	Nuevo sentido: 1= hacia delante, -1= hacia atrás
NEW_CV	IN	DInt	Nuevo valor del contador
NEW_RV	IN	DInt	Nuevo valor de referencia
NEW_PERIOD	IN	Int	El valor de período nuevo es en milisegundos (solo para el modo de medición de frecuencia). Los únicos valores permitidos son 10, 100 o 1000 milisegundos: 1000 = 1 segundo 100 = 0,1 segundo 10 = 0,01 segundo
BUSY[3]	OUT	Bool	La función está ocupada
STATUS	OUT	Word	Código de condición de ejecución

[1] Si no se solicita la actualización de un parámetro, se ignoran los valores de entrada correspondientes.

[2] El parámetro DIR es válido solo si el sentido de contaje se ajusta a "Programa de usuario (control interno de sentido)". La forma de utilizar este parámetro se determina en la configuración de dispositivos HSC.

[3] En un HSC en la CPU o en la Signal Board, el parámetro BUSY siempre está puesto a 0.

Los parámetros de cada HSC se configuran en la configuración de dispositivos de la CPU para la función de contaje/frecuencia, las opciones de reset, la configuración de eventos de alarma, las E/S de hardware y la dirección de valor de contaje.

Algunos de los parámetros del HSC se pueden modificar mediante el programa de usuario para ofrecer un control de programa del proceso de contaje:

- Ajustar el sentido de contaje al valor NEW_DIR
- Ajustar el valor de contaje actual al valor NEW_CV
- Ajustar el valor de referencia al valor NEW_RV
- Ajustar el valor Periodo (para el modo de medición de frecuencia) al valor NEW_PERIOD

Si las siguientes marcas booleanas están puestas a 1 cuando se ejecuta la instrucción CTRL_HSC, el valor NEW_xxx correspondiente se carga en el contador. Las órdenes múltiples (varias marcas se activan simultáneamente) se procesan en una sola ejecución de la instrucción CTRL_HSC.

- DIR = 1 es una orden para cargar un valor NEW_DIR, 0 = ningún cambio
- CV = 1 es una orden para cargar un valor NEW_CV, 0 = ningún cambio
- RV = 1 es una orden para cargar un valor NEW_RV, 0 = ningún cambio
- PERIOD = 1 es una orden para cargar un valor NEW_PERIOD, 0 = ningún cambio

Si se produce un error, ENO se pone a "0" y la salida STATUS indica un código de condición:

Tabla 10-14 Execution condition codes

STATUS (W#16#)	Descripción
0	No hay error
80A1	El identificador HSC no direcciona un HSC
80B1	Valor de NEW_DIR no permitido
80B2	Valor de NEW_CV no permitido
80B3	Valor de NEW_RV no permitido
80B4	Valor de NEW_PERIOD no permitido
80C0	Acceso múltiple al contador rápido Este error puede ocurrir si el modo de contaje (Página 556) se ajusta a "Período" o "Motion Control". Estos tipos no son válidos para la instrucción CTRL_HSC y solo son soportados por la instrucción CTRL_HSC_EXT.
80D0	Contador rápido (HSC) no habilitado en la configuración hardware de la CPU

10.1.4.2 Uso de CTRL_HSC

Generalmente, la instrucción CTRL_HSC se inserta en un OB de alarma de proceso que se ejecuta cuando se dispara el evento de alarma de proceso del contador. Por ejemplo, si un evento CV=RV dispara la alarma del contador, un OB de alarma de proceso ejecuta la instrucción CTRL_HSC y puede cambiar el valor de referencia cargando un valor NEW_RV.

El valor de contaje actual no está disponible en los parámetros de CTRL_HSC. La dirección de la memoria imagen de proceso que almacena el valor de contaje actual se asigna al configurar el hardware del contador rápido. Se puede emplear la lógica del programa para leer directamente el valor de contaje. El valor devuelto al programa será un contaje correcto para el instante en el que se leyó el contador. El contador continuará contando eventos rápidos. Por tanto, el valor de contaje real puede cambiar antes de que el programa finalice un proceso utilizando un valor de contaje antiguo.

10.1.4.3 Valor de contaje actual del HSC

La CPU almacena el valor actual de cada HSC en una dirección de entrada (I). La tabla siguiente muestra las direcciones predeterminadas asignadas al valor actual de cada HSC. La dirección I del valor actual se puede cambiar modificando las propiedades de la CPU en la "Configuración de dispositivos".

Los contadores rápidos usan un valor DInt para almacenar el valor de contaje actual. Un valor de contaje DInt tiene un rango entre -2147483648 y +2147483647. A partir del firmware V4.2 de la CPU es posible configurar los límites del rango. Encontrará más información en "Valores iniciales (Página 560)".

El contador pasa del valor máximo positivo al valor máximo negativo al realizar el contaje ascendente y del valor máximo negativo al valor máximo positivo al realizar el contaje

descendente. La frecuencia se devuelve en unidades de hertzios (p. ej. 123,4 Hz se devuelven como 123).

Tabla 10-15 Direcciones predeterminadas del HSC

HSC	Tipos de datos de valores actuales	Dirección de valor actual predeterminada
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

10.2 Motion Control

10.2.1 Sinopsis de Motion Control

TIA Portal, en combinación con la funcionalidad Motion Control de la CPU S7-1200, ayuda a controlar motores paso a paso y servomotores:

- Los ejes de posicionamiento y los objetos tecnológicos de la tabla de peticiones se configuran en TIA Portal. La CPU S7-1200 utiliza esos objetos tecnológicos para controlar las salidas que controlan los accionamientos.
- En el programa de usuario, los ejes se controlan mediante instrucciones Motion Control y se inician comandos de movimiento del accionamiento.

10.2.2 Componentes de hardware para Motion Control

La configuración hardware básica para una aplicación Motion Control con la CPU S7-1200 se compone de los siguientes elementos.

CPU S7-1200

La CPU S7-1200 combina la función de un autómata programable con la funcionalidad Motion Control para el funcionamiento de accionamientos. La funcionalidad Motion Control controla y vigila los accionamientos.

Signal Board

Las Signal Boards permiten agregar entradas y salidas adicionales a la CPU. Las salidas digitales pueden usarse como salidas de generación de impulsos para controlar accionamientos de la forma necesaria. En CPU con salidas de relé, la señal de impulso no puede devolverse a las salidas integradas porque los relés no soportan las frecuencias de conmutación necesarias. Para poder trabajar con la Pulse Train Output (PTO) en estas CPU es necesario utilizar una Signal Board con salidas digitales.

Las salidas analógicas pueden usarse para controlar accionamientos analógicos conectados de la forma necesaria.

PROFINET

Utilice la interfaz PROFINET para establecer la conexión online entre la CPU S7-1200 y la programadora. Además de las funciones online de la CPU, hay funciones adicionales de puesta en marcha y diagnóstico para Motion Control.

PROFINET sigue soportando el perfil PROFIdrive para conectar accionamientos y encóders compatibles con PROFIdrive.

Accionamientos y encóders

Los accionamientos hacen que los ejes se muevan. Los encóders proporcionan la posición real para la regulación de posición de los ejes.

La tabla siguiente muestra las posibilidades de conexión para accionamientos y encóders:

Conexión de accionamiento	Control en bucle cerrado/abierto de los ejes	Conexión de encóder
Pulse Train Output (PTO) (motores paso a paso y servomotores con interfaz de impulsos)	Controlado por posición	-
Salida analógica (AQ)	Controlado por posición	• Encóder en contador rápido (HSC) • Encóder en módulo tecnológico (TM) • Encóder en PROFINET
PROFINET	Controlado por posición	• Interfaz en el accionamiento • Encóder en contador rápido (HSC) • Encóder en módulo tecnológico (TM) • Encóder en PROFINET

10.2.3 Instrucciones de Motion Control

10.2.3.1 Vista general de instrucciones de MC

Las instrucciones de Motion Control utilizan un bloque de datos tecnológico asociado y las salidas del tren de impulsos (PTO) específicas de la CPU para controlar el movimiento de un eje:

- MC_Power (Página 572) activa y desactiva un eje de Motion Control.
- MC_Reset (Página 573) resetea todos los errores de Motion Control. Se acusan todos los errores de Motion Control que pueden acusarse.
- MC_Home (Página 573) establece la relación entre el programa de control del eje y el sistema de posicionamiento mecánico del eje.
- MC_Halt (Página 574) cancela todos los procesos de movimiento y detiene el movimiento del eje. La posición de parada no está definida.
- MC_MoveAbsolute (Página 574) inicia el movimiento hacia una posición absoluta. La orden finaliza cuando se alcanza la posición de destino.
- MC_MoveRelative (Página 575) inicia un movimiento de posicionamiento relativo a la posición inicial.
- MC_MoveVelocity (Página 575) hace que el eje se mueva a la velocidad indicada.
- MC_MoveJog (Página 576) ejecuta el modo jog para fines de test y arranque.
- MC_CommandTable (Página 576) ejecuta comandos de eje como una secuencia de movimientos.
- MC_ChangeDynamic (Página 578) cambia la configuración de la dinámica del eje.
- MC_WriteParam (Página 577) escribe un número de parámetros determinado para cambiar las funciones del eje desde el programa de usuario.
- MC_ReadParam (Página 577) lee un número determinado de parámetros que indican la posición y la velocidad (entre otros valores actuales) del eje definidos en la entrada del eje.

Niveles de firmware de CPU

Si tiene una CPU S7-1200 con firmware V4.1 o posterior, seleccione la versión V5.0 para todas las instrucción de movimiento.

Si tiene una CPU S7-1200 con firmware V4.0 o anterior, seleccione la versión correspondiente V4.0, V3.0, V2.0 o V1.0 de cada instrucción de movimiento.

Si tiene una CPU S7-1200 con firmware V4.2 o posterior, seleccione la versión V6.0 para cada instrucción de movimiento.

Si tiene una CPU S7-1200 con firmware V4.4 o posterior, seleccione la versión V7.0 para cada instrucción de movimiento.

Si tiene una CPU S7-1200 con firmware V4.5 o posterior, seleccione la versión V8.0 para cada instrucción de movimiento.

Nota

Las instrucciones de Motion Control V1.0 a V3.0 controlan activamente la salida de la instrucción. Cuando se produce un error en el bloque, la salida de habilitación (ENO) se pone a OFF. Las salidas ERROR, ErrorID y ErrorInfo del bloque indican un error. La salida ENO permite evaluar el estado de la instrucción y ejecutar después en serie instrucciones posteriores.

Con las instrucciones de Motion Control V4.0 y V5.0, la salida ENO se mantiene en TRUE mientras se está ejecutando, independientemente de su estado de error. Esto podría hacer que un programa que utilizaba Motion Control V3.0 o anterior, que dependa del estado de ENO, funcione de forma incorrecta. Para solucionar esta situación, cuando se usa Motion Control V4.0 o posterior deben utilizarse las salidas DONE y ERROR para evaluar el estado de la instrucción y no la salida ENO.

10.2.3.2 MC_Power (habilitar/bloquear eje)

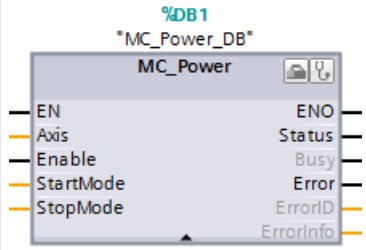
La instrucción MC_Power activa y desactiva un eje de Motion Control.

Nota

Si el eje se desconecta debido a un error, se habilitará de nuevo automáticamente una vez que el error haya sido eliminado y acusado. Para ello es necesario que el parámetro de entrada Enable haya conservado el valor TRUE durante el proceso.

Tabla 10-16 Instrucción MC_Power

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción MC_Power de Motion Control habilita o deshabilita un eje. Antes de poder habilitar o deshabilitar el eje, garantice las siguientes condiciones:
		• El objeto tecnológico está configurado correctamente.
		• No hay ningún error pendiente que impida la habilitación.
		La ejecución de MC_Power no puede ser cancelada por una solicitud de Motion Control. La deshabilitación del eje (parámetro de entrada Enable = FALSE) cancela todas las órdenes de Motion Control para el objeto tecnológico asociado.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "MC_Power_DB" es el nombre del DB de instancia.

10.2.3.3 MC_Reset (confirmar error)

La instrucción MC_Reset resetea todos los errores de Motion Control. Se acusan todos los errores de Motion Control que pueden acusarse.

Tabla 10-17 Instrucción MC_Reset

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_Reset para acusar "Error operativo con parada de eje" y "Error de configuración". Los errores que requieren acuse pueden encontrarse en la "Lista de ErrorIDs y ErrorInfos" en "Solución".
		Antes de utilizar la instrucción MC_Reset es necesario haber eliminado la causa de cualquier error de configuración presente sujeto a acuse (por ejemplo, cambiando un valor de aceleración no válido en el objeto tecnológico "Eje" a un valor válido).
		En V3.0 y posteriores, el comando Restart permite descargar la configuración del eje a la memoria de trabajo en el modo de operación RUN.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "MC_Reset_DB" es el nombre del DB de instancia.

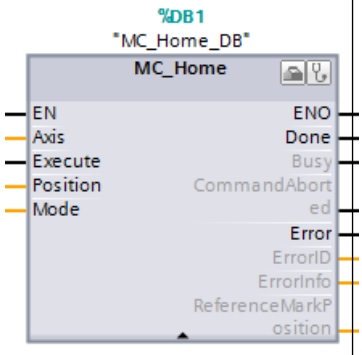
La orden MC_Reset no puede ser interrumpida por otra orden de Motion Control. Las nuevas órdenes MC_Reset no interrumpen ninguna otra orden de Motion Control activa.

10.2.3.4 MC_Home (referenciar eje)

La instrucción MC_Home establece la relación entre el programa de control del eje y el sistema de posicionamiento mecánico del eje.

Tabla 10-18 Instrucción MC_Home

KOP / FUP	SCL	Descripción

	`"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_`
	`);`
		Utilice la instrucción MC_Home para cuadrar las coordenadas del eje con la posición física real del accionamiento. Se requiere un referenciado para posicionar el eje de forma absoluta:
		Para utilizar la instrucción MC_Home primero es necesario haber habilitado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

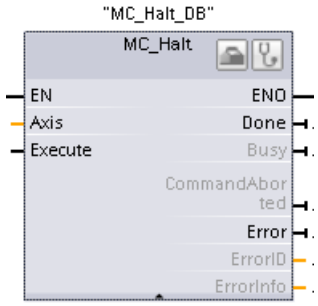
[2] En el ejemplo SCL, "MC_Home_DB" es el nombre del DB de instancia.

10.2.3.5 MC_Halt (detener eje)

La instrucción MC_Halt cancela todos los procesos de movimiento y detiene el movimiento del eje. La posición de parada no está definida.

Tabla 10-19 Instrucción MC_Halt

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_Halt para parar todo el movimiento y detener el eje por completo. La posición de eje totalmente detenido no está definida.
		Para utilizar la instrucción MC_Halt primero es necesario haber habilitado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

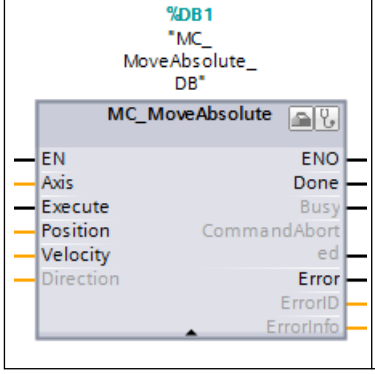
[2] En el ejemplo SCL, "MC_Halt_DB" es el nombre del DB de instancia.

10.2.3.6 MC_MoveAbsolute (posicionamiento absoluto del eje)

La instrucción MC_MoveAbsolute inicia el movimiento hacia una posición absoluta. La tarea finaliza cuando se alcanza la posición de destino.

Tabla 10-20 Instrucción MC_MoveAbsolute

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_MoveAbsolute para iniciar un movimiento de posicionamiento del eje a una posición absoluta.
		Para utilizar la instrucción MC_MoveAbsolute primero es necesario haber habilitado y referenciado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

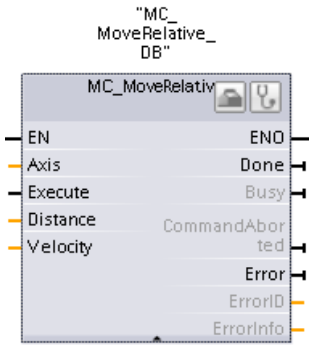
[2] En el ejemplo SCL, "MC_MoveAbsolute_DB" es el nombre del DB de instancia.

10.2.3.7 MC_MoveRelative (posicionamiento relativo del eje)

La instrucción MC_MoveRelative inicia un movimiento de posicionamiento relativo a la posición inicial.

Tabla 10-21 Instrucción MC_MoveRelative

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_MoveRelative para iniciar un movimiento de posicionamiento relativo a la posición inicial.
		Para utilizar la instrucción MC_MoveRelative primero es necesario haber habilitado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

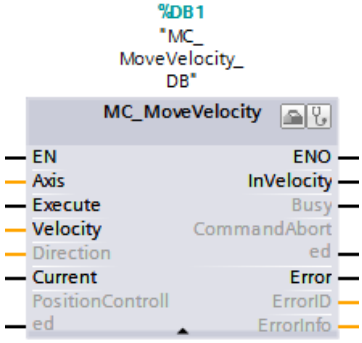
[2] En el ejemplo SCL, "MC_MoveRelative_DB" es el nombre del DB de instancia.

10.2.3.8 MC_MoveVelocity (mover el eje a la velocidad predefinida)

La instrucción MC_MoveVelocity hace que el eje se mueva a la velocidad indicada.

Tabla 10-22 Instrucción MC_MoveVelocity

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_MoveVelocity para mover el eje constantemente a la velocidad especificada.
		Para utilizar la instrucción MC_MoveVelocity primero es necesario haber habilitado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

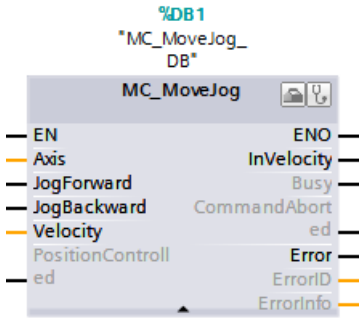
[2] En el ejemplo SCL, "MC_MoveVelocity_DB" es el nombre del DB de instancia.

10.2.3.9 MC_MoveJog (desplazamiento del eje en modo Jog)

La instrucción MC_MoveJog ejecuta el modo jog para fines de test y arranque.

Tabla 10-23 Instrucción MC_MoveJog

KOP / FUP	SCL	Descripción

	`"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_);`
		Utilice la instrucción MC_MoveJog para mover el eje constantemente a la velocidad específica en modo paso a paso. Esta instrucción se suele utilizar con fines de prueba y mantenimiento.
		Para utilizar la instrucción MC_MoveJog primero es necesario haber habilitado el eje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

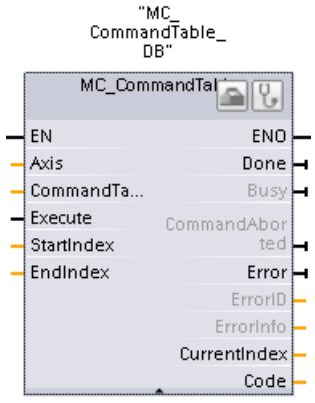
[2] En el ejemplo SCL, "MC_MoveJog_DB" es el nombre del DB de instancia.

10.2.3.10 MC_CommandTable (ejecutar comandos de eje como secuencia de movimiento)

La instrucción MC_CommandTable ejecuta comandos de eje como una secuencia de movimientos.

Tabla 10-24 Instrucción MC_CommandTable

KOP / FUP	SCL	Descripción

	`"MC_CommandTable_DB"(`
	`Axis:=_multi_fb_in_,`
	`CommandTable:=_multi_fb_in_,`
	`Execute:=_bool_in_,`
	`StartStep:=_uint_in_,`
	`EndStep:=_uint_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`CommandAborted=>_bool_out_,`
	`Error=>_bool_out_,`
	`ErrorID=>_word_out_,`
	`ErrorInfo=>_word_out_,`
	`CurrentStep=>_uint_out_,`
	`StepCode=>_word_out_);`
		Ejecuta una serie de movimientos individuales para un eje controlado por motor que se combinan en una secuencia de movimientos.
		Los movimientos individuales están configurados en la tabla de comandos de un objeto tecnológico para la salida del tren de impulsos (TO_CommandTable_PTO).

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

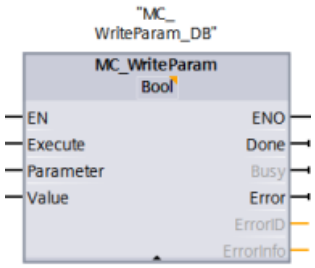
[2] En el ejemplo SCL, "MC_CommandTable_DB" es el nombre del DB de instancia.

10.2.3.11 MC_WriteParam (escribir parámetros de un objeto tecnológico)

La instrucción MC_WriteParam se usa para escribir un número determinado de parámetros para cambiar la funcionalidad del eje desde el programa del usuario.

Tabla 10-25 Instrucción MC_WriteParam

KOP / FUP	SCL	Descripción

	`"MC_WriteParam_DB"(`
	`Parameter:=_variant_in_,`
	`Value:=_variant_in_,`
	`Execute:=_bool_in_,`
	`Done:=_bool_out_,`
	`Error:=_real_out_,`
	`ErrorID:=_word_out_,`
	`ErrorInfo:=_word_out_);`
		La instrucción MC_WriteParam se utiliza para escribir en parámetros públicos (por ejemplo, los valores de DB de aceleración y usuario).

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

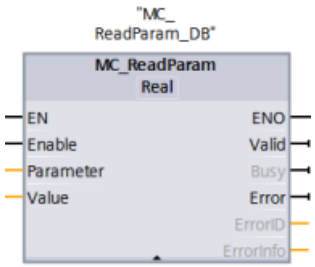
[2] En el ejemplo de SCL, "MC_WriteParam_DB" es el nombre del DB de instancia.

10.2.3.12 Instrucción MC_ReadParam (leer parámetros de un objeto tecnológico)

La instrucción MC_ReadParam se usa para leer un número determinado de parámetros que indican la posición y la velocidad (entre otros valores actuales) del eje definidos en la entrada del eje.

Tabla 10-26 Instrucción MC_ReadParam

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción MC_ReadParam se utiliza para leer valores de estado únicos, al margen del punto de control del ciclo.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

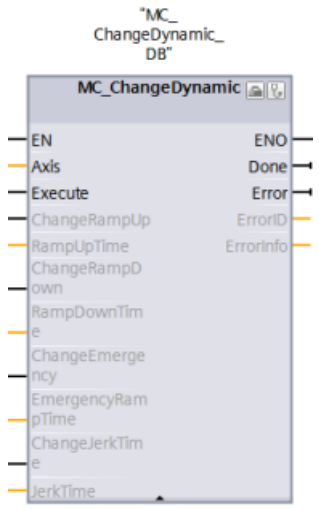
[2] En el ejemplo SCL, "MC_ReadParam_DB" es el nombre del DB de instancia.

10.2.3.13 MC_ChangeDynamic (cambiar la configuración de la dinámica del eje)

Con la instrucción MC_ChangeDynamic se cambian los ajustes dinámicos de un eje de Motion Control.

Tabla 10-27 Instrucción MC_ChangeDynamic

KOP / FUP	SCL	Descripción

	`"MC_ChangeDynamic_DB"(Axis:=_param_fb_in_,`
	`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_);`
		Modifica los ajustes dinámicos de un eje de Motion Control:
		• Cambia el valor del tiempo de aceleración
		• Cambia el valor del tiempo de deceleración
		• Cambia el valor del tiempo de deceleración de parada de emergencia
		• Cambiar el valor del tiempo de suavizado (tirones)

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

[2] En el ejemplo SCL, "MC_ChangeDynamic_DB" es el nombre del DB de instancia.

10.2.4 Información adicional sobre S7-1200 Motion Control

Para más información sobre S7-1200 Motion Control consulte SIMATIC STEP 7 S7-1200 Motion Control V6.0 en TIA Portal V15 (https://support.industry.siemens.com/cs/us/es/view/109773400).

En ese manual encontrará información detallada sobre los siguientes temas:

- Configuración de un generador de impulsos
- Control de movimiento en lazo abierto
- Configuración de los ejes
- Puesta en marcha
- Control de movimiento en lazo cerrado
- Configuración de los ejes
- ServoOB
- Funcionamiento regulado por velocidad
- Soporte de telegrama 4
- Eje de simulación

- Adaptación de datos
- Control de ejes mediante el módulo TM Pulse
- Configuración de TO_CommandTable_PTO
- Funcionamiento de Motion Control para S7-1200
- Salidas de CPU utilizadas para Motion Control
- Finales de carrera por hardware y software para Motion Control
- Referenciado
- Limitación de tirones
- Vigilancia de comandos activos
- Vigilancia de instrucciones MC con un parámetro de salida "Done"
- Vigilancia de MC_Velocity
- Vigilancia de MC_MoveJog
- ErrorID y ErrorInfos para Motion Control

Consulte también

SIMATIC STEP 7 S7-1200 Motion Control V6.0 en TIA Portal V15 (https://support.industry.siemens.com/cs/us/en/view/109773400)

10.3 PID Control

10.3.1 Funciones PID

STEP 7 ofrece las siguientes instrucciones de tipo proporcional-integral-derivativo (PID) para la CPU S7-1200:

- La instrucción PID_Compact se utiliza para controlar procesos técnicos con variables continuas de entrada y salida.
- La instrucción PID_3Step se utiliza para controlar dispositivos accionados por motor, como válvulas que requieren señales discretas para las acciones de apertura y cierre.
- La instrucción PID_Temp ofrece un regulador PID universal que permite manejar los requisitos específicos de control de temperatura.

10.3.2 Instrucciones PID

10.3.2.1 Funciones PID

STEP 7 ofrece las siguientes instrucciones de tipo proporcional-integral-derivativo (PID) para la CPU S7-1200:

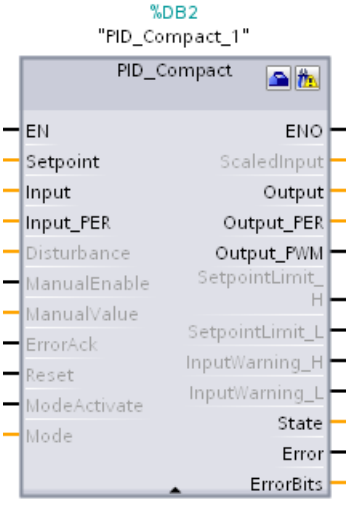
- La instrucción PID_Compact se utiliza para controlar procesos técnicos con variables continuas de entrada y salida.
- La instrucción PID_3Step se utiliza para controlar dispositivos accionados por motor, como válvulas que requieren señales discretas para las acciones de apertura y cierre.
- La instrucción PID_Temp ofrece un regulador PID universal que permite manejar los requisitos específicos de control de temperatura.

10.3.2.2 Instrucción PID_Compact

La instrucción PID_Compact ofrece un regulador PID universal con autoajuste integrado para modo automático y manual.

Tabla 10-28 Instrucción PID_Compact

KOP / FUP	SCL	Descripción

	`"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 ofrece un regulador PID con autoajuste para modo automático y modo manual. PID_Compact es un regulador PID T1 con anti-windup y ponderación de las acciones P y D.

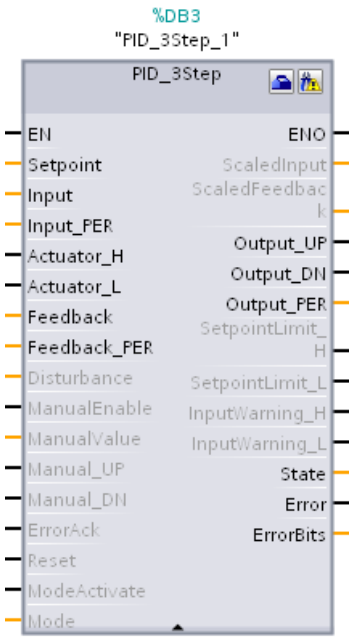
[1] STEP 7 crea automáticamente el objeto tecnológico y el DB de instancia al insertar la instrucción. El DB de instancia contiene los parámetros del objeto tecnológico.

[2] En el ejemplo SCL, "PID_Compact_1" es el nombre del DB de instancia.

10.3.2.3 Instrucción PID_3Step

La instrucción PID_3Step configura un regulador PID con capacidades de autoajuste que se ha optimizado para válvulas accionadas por motor y actuadores.

Tabla 10-29 Instrucción PID_3Step

KOP / FUP	SCL	Descripción
	**"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_);**	PID_3Step configura un regulador PID con capacidades de autoajuste que se ha optimizado para válvulas accionadas por motor y actuadores. Proporciona dos salidas booleanas. PID_3Step es un regulador PID T1 con anti-windup y ponderación de las acciones P y D.

[1] STEP 7 crea automáticamente el objeto tecnológico y el DB de instancia al insertar la instrucción. El DB de instancia contiene los parámetros del objeto tecnológico.

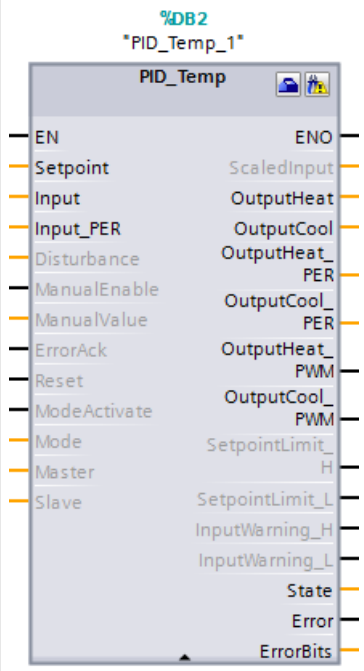
[2] En el ejemplo SCL, "PID_3Step_1" es el nombre del DB de instancia.

10.3.2.4 Instrucción PID_Temp

La instrucción PID_Temp ofrece un regulador PID universal que permite manejar los requisitos específicos de control de temperatura.

Tabla 10-30 Instrucción PID_Temp

KOP / FUP	SCL	Descripción

	`"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 ofrece las funcionalidades siguientes:
		• Calentar y refrigerar el proceso con diferentes actuadores
		• Autotuning integrado para manejar procesos de temperatura
		• Función de cascada para procesar más de una temperatura que depende del mismo actuador.

[1] STEP 7 crea automáticamente el objeto tecnológico y el DB de instancia al insertar la instrucción. El DB de instancia contiene los parámetros del objeto tecnológico.

[2] En el ejemplo SCL, "PID_Temp_1" es el nombre del DB de instancia.

10.3.3 Información adicional sobre la regulación PID de S7-1200

Consulte manual de funciones SIMATIC S7-1200, S7-1500 Regulación PID (https://support.industry.siemens.com/cs/us/es/view/108210036) para más información sobre la regulación PID de S7-1200.

En ese manual encontrará información detallada sobre los siguientes temas:

- Fundamentos de la regulación
- Configuración de un controlador por software

- Uso de PID_Compact:
  - Límites de valores de proceso
  - Parámetros ErrorBit
  - Parámetros de advertencia
- Uso de PID_3Step:
  - Parámetros ErrorBit
  - Parámetros de advertencia
- Uso de PID_Temp:
  - Parámetros ErrorBit
  - Parámetros de advertencia
- Configuración de reguladores PID_Compact, PID_3Step y PID_Temp
- Puesta en marcha de reguladores PID_Compact, PID_3Step y PID_Temp

Comunicación 11

11.1 Sinopsis

S7-1200 ofrece varios tipos de comunicación entre CPU, y programadoras, HMI y otras CPU.

PROFINET

PROFINET se usa para intercambiar datos a través del programa de usuario con otros interlocutores vía Ethernet:

- En el S7-1200, PROFINET soporta 16 dispositivos E/S con un máximo de 256 submódulos, y PROFIBUS admite 3 maestros PROFIBUS DP independientes, con 32 esclavos por maestro DP y un máximo de 512 módulos por maestro DP.
- Comunicación S7
- Protocolo User Datagram Protocol (UDP)
- ISO on TCP (RFC 1006)
- Transport Control Protocol (TCP)

Controlador PROFINET IO

En calidad de controlador IO que utiliza PROFINET IO, la CPU se comunica con un máximo de 16 dispositivos PN en la red PN local o a través de un acoplador PN/PN (link). Véase PROFIBUS and PROFINET International, PI (www.profinet.com) para más información.

PROFIBUS

PROFIBUS se usa para intercambiar datos a través del programa de usuario con otros interlocutores a través de la red PROFIBUS:

- Con CM 1242-5 la CPU funciona como esclavo PROFIBUS DP.
- Con CM 1243-5 la CPU funciona como maestro PROFIBUS DP de clase 1.
- Los esclavos PROFIBUS DP, los maestros PROFIBUS DP, así como AS-i (los tres módulos de comunicación del lado izquierdo) y PROFINET son redes de comunicación separadas que no se limitan unas a otras.

AS-i

el CM 1243-2 maestro AS-i del S7-1200 permite conectar una red AS-i a una CPU S7-1200.

Comunicación S7 de CPU a CPU

Se puede crear una conexión con un interlocutor y utilizar las instrucciones GET y PUT para la comunicación con CPU S7.

Comunicación TeleService

En TeleService vía GPRS, una estación de ingeniería en la que está instalado STEP 7 se comunica mediante la red GSM e Internet con una estación SIMATIC S7-1200 con un CP 1242-7. La conexión se ejecuta a través de un servidor de telecontrol que actúa de intermediario y que está conectado a Internet.

IO-Link

El maestro S7-1200 SM 1278 4xIO-Link permite a los dispositivos IO-Link conectarse con un S7-1200 CPU.

Prevenir riesgos de seguridad por ataques a la red física

⚠ ADVERTENCIA

Prevenir riesgos de seguridad por ataques a la red física

Si un atacante puede acceder físicamente a las redes, posiblemente podrá leer y escribir datos.

Algunas formas de comunicación (intercambio E/S a través de PROFIBUS, PROFINET, AS-i u otro bus de periferia, GET/PUT, bloque T y módulos de comunicación (CM)) carecen de funciones de seguridad. Este tipo de comunicaciones deben protegerse limitando el acceso físico. Si un atacante puede acceder físicamente a las redes con estos tipos de comunicación, posiblemente podrá leer y escribir datos.

Si no consigue proteger estas formas de comunicación, podrían producirse la muerte o lesiones corporales graves.

Para más información sobre seguridad y recomendaciones, consulte las "Operational Guidelines for Industrial Security" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) en la página del Siemens Industry Online Support.

Consulte también

Comunicación segura (secure) y comunicación antigua (legacy) (Página 586)

11.2 Comunicación segura (secure) y comunicación antigua (legacy)

La versión V4.5 de la CPU S7-1200 implementa comunicación segura entre los PLC y TIA Portal, SIMATIC Automation Tool y HMI. Esta implementación utiliza el protocolo estándar industrial TLS 1.3 (Transport Layer Security). La CPU S7-1200 ha incorporado mejoras en la comunicación segura desde V4.0. Para mantener la compatibilidad con dispositivos y clientes más antiguos, se continúa conservando la comunicación antigua (legacy). Tiene la opción de utilizar la comunicación segura (secure) o antigua (legacy). Siemens recomienda encarecidamente la comunicación segura para los clientes y dispositivos que la soporten.

Ventajas de la comunicación segura

Utilice la comunicación segura para alcanzar los objetivos siguientes:

- Confidencialidad: Los datos son secretos y posibles intrusos no pueden leerlos.
- Integridad: El mensaje que llega al destinatario es el mismo mensaje, sin modificaciones, que envió el remitente. El mensaje no sufre alteraciones por el camino.
- Autenticación de punto final: El interlocutor del punto final es exactamente quien dice ser y la parte a la que se quiere acceder. La identidad del interlocutor está verificada.

Hoy en día, las máquinas industriales y los sistemas de control conectados en red que gestionan datos sensibles están expuestos a un alto riesgo y plantean elevados requisitos de seguridad para el intercambio de datos.

La protección mediante firewall, conexión VPN o un módulo de seguridad era común en el pasado y lo sigue siendo. No obstante, además de la seguridad física, el uso de la comunicación segura para transferir datos a interlocutores externos de forma cifrada es cada vez más necesario.

La CPU utiliza certificados X.509 para establecer una comunicación segura entre la CPU y los clientes. Algunos clientes, como STEP 7 y SIMATIC Automation Tool, pueden exigir que el certificado que se encuentra en la CPU sea de confianza. Tenga en cuenta qué certificado descarga en su CPU de modo que pueda confiar en él cuando así se le solicite.

Consideraciones para el uso de la comunicación segura

Tenga en cuenta los siguientes aspectos relacionados con la comunicación segura:

- Las velocidades de comunicación pueden ser más lentas con comunicación segura (secure) que con la comunicación antigua (legacy).
- La comunicación PG/PC y HMI segura requiere TIA Portal V17 y CPU S7-1200 V4.5 o posteriores.

Las capacidades de comunicación segura dependen de varios factores:

- versión actual del firmware de la CPU física
- versión de firmware configurada de la CPU en el proyecto STEP 7

La tabla siguiente muestra la comunicación soportada para una CPU S7-1200 con firmware V4.5:

Configuración de dispositivo de la CPU en el proyecto STEP 7	Versiones de cliente			
	TIA Portal V17 / SIMATIC Automation Tool V4.0 SP3	**TIA Portal < V17 / SIMATIC Automation Tool < V4.0 SP3**	**HMI V17[1]**	**HMI < V17**
Versión de firmware configurada en el proyecto STEP 7: V4.5 Ajuste de configuración: "Permitir solo la comunicación PG/PC y HMI segura" seleccionado	Segura	Sin conexión	Segura	Sin conexión
Versión de firmware configurada en el proyecto STEP 7: V4.5 Ajuste de configuración: "Permitir solo la comunicación PG/PC y HMI segura" deseleccionado	Segura por defecto, legacy configurable	Antigua (legacy)	Segura	Antigua (legacy)
Versión de firmware configurada en el proyecto STEP 7: V4.4	Antigua (legacy)			
Sin proyecto	Segura o antigua (secure/legacy)	Antigua (legacy)	No aplicable	

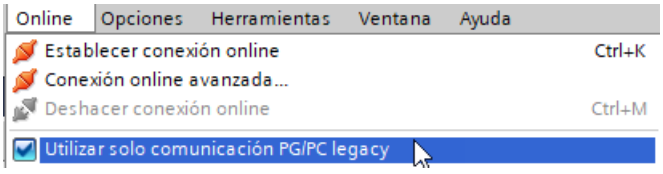
[1] No aplicable a HMI con comunicación S7

Debe configurar la versión de firmware de la CPU en el proyecto STEP 7 de modo que sea V4.5 para usar TLS 1.3. La CPU utiliza TLS 1.2 si la versión de firmware configurada en el proyecto es anterior a V4.5.

Comunicación segura (secure) o comunicación antigua (legacy)

Las funciones de seguridad de la CPU se configuran en la configuración de dispositivo (Página 160) de la CPU en TIA Portal, en Protección y seguridad. Los cuadros de diálogo de TIA Portal proporcionan acceso al sistema de información de TIA Portal para guiarle a través del proceso de configuración. Para simplificar la configuración, TIA Portal V17 o superior ofrece un asistente de seguridad (Página 160) para guiarle a través de las opciones de seguridad.

TIA Portal V17 y posteriores ajustan por defecto la comunicación PG/PC y HMI segura, pero por motivos de puesta en marcha se puede forzar a TIA Portal a que use comunicación PG/PC legacy seleccionando "Utilizar solo comunicación PG/PC legacy" en el menú Online.



Información adicional

Encontrará más detalles sobre la implementación de la comunicación segura en el sistema de información de TIA Portal. Encontrará información adicional sobre certificados, en particular, en los siguientes temas del sistema de información de TIA Portal:

- Confidencialidad mediante cifrado
- Administrar certificados con STEP 7
- Ejemplos de administración de certificados
- Autenticidad e integridad mediante firmas

Consulte también

Protección de datos de configuración confidenciales del PLC (Página 161)

Activar la comunicación PG/PC y HMI segura y crear certificados (Página 166)

Sustituir una CPU con protección de datos de configuración confidenciales del PLC (Página 1447)

11.3 Protocolos de comunicación y puertos utilizados por la comunicación Ethernet

La presente relación indica los protocolos y puertos soportados para la comunicación a través de interfaces PN/IE. Los puertos especificados son los números de puerto estándar que usa el PLC S7-1200. Muchos protocolos de comunicación y muchas implementaciones permiten utilizar otros números de puerto. Las tablas siguientes muestran diferentes capas, protocolos y puertos utilizados en el PLC S7-1200.

Tabla 11-1 Puertos y protocolos de la capa de transporte en S7-1200

Puerto(s)	Sentido	Protocolo	Aplicación	Descripción
25	Salida	TCP	SMTP	SMTP se utiliza para enviar correos electrónicos.
80	Entrada	TCP	HTTP	HTTP se utiliza para la comunicación con el servidor web interno de la CPU.
102[1]	Entrada/salida	TCP	ISO on TCP	ISO on TCP (conforme a RFC 1006) se utiliza para el intercambio de datos orientado a mensajes con CPU remotas, la comunicación S7 con ES y HMI.
123	Salida	UDP	NTP	NTP se utiliza para sincronizar la hora del sistema de la CPU con la hora de un servidor NTP.
161[1]	Entrada	UDP	SNMP	SNMP se utiliza para leer y ajustar datos de gestión de red (objetos gestionados por SNMP) mediante el administrador SNMP.
443	Entrada	TCP	HTTPS	HTTPS se utiliza para la comunicación con el servidor web interno de la CPU vía TLS.
465, 587	Salida	TCP	SMTPS	SMTP se utiliza para enviar correos electrónicos a través de conexiones seguras.
502	Entrada/salida	TCP	Modbus	Modbus/TCP es utilizado por instrucciones MB_CLIENT/ MB_SERVER en el programa de usuario.

Puerto(s)	Sentido	Protocolo	Aplicación	Descripción
4840[2]	Entrada	TCP	OPC UA	Estándar de comunicación que abarca desde el nivel corporativo hasta el nivel de campo.
34964[1]	Entrada/salida	UDP	PROFINET Context Manager	PROFINET Context Manager proporciona una herramienta de mapeo de puntos finales que permite establecer una relación de aplicación (PROFINET AR).

[1] Estos puertos están abiertos y son accesibles en la configuración inicial con dirección IP configurada. Otras aplicaciones deben activarse o configurarse como parte del programa de usuario S7-1200.

[2] El puerto 4840 es el predeterminado, aunque también puede configurarse.

Tabla 11-2 Rangos de puertos que podrían utilizarse en la Open User Communication (OUC) y en otros protocolos. Los parámetros de comunicación exactos son definidos por el usuario como parte del programa de usuario S7-1200.

Rango de puertos	Sentido	Protocolo	Aplicación	Descripción
1-999	Varía	TCP/UDP	OUC	El rango de puertos puede utilizarse con limitaciones, excluyendo los puertos ya empleados.
2000-5000	Varía	TCP/UDP	OUC	Rango de puertos recomendando para OUC
5001-49151	Varía	TCP/UDP	OUC	El rango de puertos puede utilizarse con limitaciones, excluyendo los puertos ya empleados.
49152-65535	Salida	TCP/UDP	Varía	Área de puertos dinámica utilizada para un punto final de conexión activa si la aplicación no determina el número de puerto local.

Tabla 11-3 Protocolos utilizados por S7-1200 en la capa de enlace de datos y de red (Layer 2, 3) del modelo OSI.

Protocolo	Sentido	Ethertype	Descripción
PROFINET DCP	Entrada/salida	0x8892	PROFINET utiliza DCP para detectar dispositivos PROFINET y asignarles los ajustes básicos.
LLDP	Salida	0x88CC	PROFINET utiliza LLDP para detectar y administrar relaciones de vecindad entre dispositivos PROFINET. LLDP utiliza la dirección MAX Multicast especial: 01-80-C2-00-00-0E.
PROFINET IO	Entrada/salida	0x8892	Los frames PROFINET IO sirven para transmitir datos E/S cíclicamente entre el controlador PROFINET IO y los dispositivos IO a través de Ethernet.
ICMP	Entrada	0x800	Internet Control Message Protocol sirve para fines de diagnóstico o control.

11.4 Conexiones de comunicación asíncronas

Sinopsis de los servicios de comunicación

La CPU soporta los siguientes servicios de comunicación:

Servicio de comunicación	Funcionalidad	Con PROFIBUS DP		Con Ethernet
		Módulo maestro CM 1243-5 DP	Módulo esclavo CM 1242-5 DP	
Comunicación PG	Puesta en marcha, test y diagnóstico	Sí	No	Sí
Comunicación HMI	Manejo y visualización	Sí	No	Sí
Comunicación S7	Intercambio de datos utilizando conexiones configuradas	Sí	No	Sí
Routing de funciones PG	Por ejemplo, test y diagnóstico más allá de los límites de la red	No	No	No
PROFIBUS DP	Intercambio de datos entre maestro y esclavo	Sí	Sí	No
PROFINET IO	Intercambio de datos entre controladores IO y dispositivos IO	No	No	Sí
Servidor web	Diagnóstico	No	No	Sí
SNMP [1] (Simple Network Management Protocol)	Protocolo estándar para diagnóstico y parametrización de la red	No	No	Sí
Routing S7	Mediante las tablas de routing, los interlocutores de la comunicación pueden comunicarse con cada uno de los dispositivos, aunque estos estén en subredes S7 distintas.	No	No	Sí
Comunicación abierta vía TCP/IP	Intercambio de datos vía Industrial Ethernet con protocolo TCP/IP (con FB cargables)	No	No	Sí
Comunicación abierta vía ISO on TCP	Intercambio de datos vía Industrial Ethernet con protocolo ISO on TCP (con FB cargables)	No	No	Sí
Comunicación abierta vía UDP	Intercambio de datos vía Industrial Ethernet con protocolo UDP (con FB cargables)	No	No	Sí
Servidor OPC UA	Intercambio de datos vía Industrial Ethernet con clientes OPC UA	No	No	Sí[2]

[1] La CPU soporta SNMP V1 sin traps.

[2] OPC UA solo se soporta si se usa el puerto Ethernet integrado en la CPU.

Conexiones disponibles

La CPU soporta el siguiente número máximo de conexiones de comunicación asíncronas simultáneas para PROFINET y PROFIBUS. El número máximo de recursos de conexión asignados a cada categoría es fijo y no puede modificarse. Sin embargo, es posible configurar las "Conexiones libres disponibles" para aumentar el número de cualquier categoría tal y como requiere su aplicación.

Algunos tipos de conexión tienen un número fijo de recursos reservados (algunas veces se denominan "garantizados"). Esto significa que la CPU está garantizada para soportar hasta el número de recursos reservados para el tipo de conexión. Por ejemplo, al menos 12 conexiones HMI pueden establecerse simultáneamente con la CPU. Pueden establecerse conexiones adicionales para un tipo de conexión por encima del número de recursos reservados pero esos recursos de conexión deben provenir del grupo de recursos "dinámico".

Los recursos dinámicos (algunas veces se denominan "libres") son una recopilación de recursos que pueden utilizase para cualquier tipo de conexión. Estos recursos son utilizados por conexiones que no cuentan con recursos reservados (como OPC UA) o por conexiones que han consumido todos sus recursos reservados.

A partir de V17, las CPU V4.5 cuentan con 34 recursos reservados.

Recursos de conexión

	Recursos de la estación			Recursos del módulo
	Reservados		Dinámicos ⚠	PLC_1 [CPU 1214C DC/DC/...
Número máximo de recursos:		34	34	68
	Máximo	Configurados	Configurados	Configurados
Comunicación PG:	4	-	-	-
Comunicación HMI:	12	0	0	0
Comunicación S7:	8	0	0	0
Open User Communication:	8	0	0	0
Comunicación web:	2	-	-	-
Comunicación cliente/servidor OPC ...	0	-	-	-
Otros tipos de comunicación:	-	-	0	0
Recursos utilizados en total:		0	0	0
Recursos disponibles:		34	34	68

Nota

El número total de conexiones de comunicación del S7-1200 no aumenta al agregar módulos CM/CP.

Nota

Conexiones OPC UA

Las conexiones OPC UA consumen recursos de los recursos dinámicos. Asegúrese de tener suficientes conexiones disponibles para la aplicación.

En función de los recursos de conexión asignados está disponible el siguiente número de conexiones por dispositivo:

Tipo	Reservado	Máximo[1]
Comunicación con programadora (PG)	4	4
Comunicación con interfaz hombre-máquina (HMI)	12	18
Comunicación S7	8	14
Open User Communication	8	14
Comunicación web	2	30
Comunicación cliente/servidor OPC UA	0	10
Otras comunicaciones	0	-

[1] Dado que las conexiones dinámicas se comparten, no es posible agotar todas las conexiones simultáneamente.

Una CPU, por ejemplo, tiene disponibles cuatro recursos de conexión de una programadora. Según sean las funciones de programadora que se utilicen, esta puede utilizar uno, dos o tres de los recursos de conexión disponibles. Siempre puede utilizarse una programadora.

Otro ejemplo es el número de HMI, tal como se muestra en la figura inferior. Los dispositivos HMI disponen de 12 recursos de conexión. Según sea el tipo o modelo de HMI disponible y las funciones HMI que se utilicen, cada HMI podría utilizar uno, dos o tres de sus recursos de conexión disponibles. En función del número de recursos de conexión disponibles que se utilicen, existe la posibilidad de utilizar más de cuatro dispositivos HMI a la vez. Sin embargo, siempre se garantiza la conexión de cuatro dispositivos HMI como mínimo. Un HMI puede emplear sus recursos de conexión disponibles (uno en cada caso de un total de tres) para las funciones siguientes:

- leer
- escribir
- alarmas más diagnóstico

Esto es solo un ejemplo. El número real de conexiones utilizadas puede varias según el tipo y la versión de HMI.

Ejemplo	HMI 1	HMI 2	HMI 3	HMI 4	HMI 5	Total de recursos de conexión disponibles
Recursos de conexión utilizados	2	2	2	3	3	12

Nota

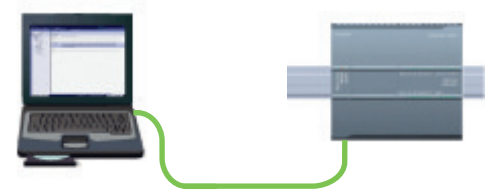
Conexiones de servidor web: la CPU proporciona conexiones para varios navegadores web. El número de navegadores que soporta la CPU simultáneamente depende de las conexiones que solicite o utilice un navegador web concreto y del número de recursos de conexión dinámica disponible en la CPU.

Nota

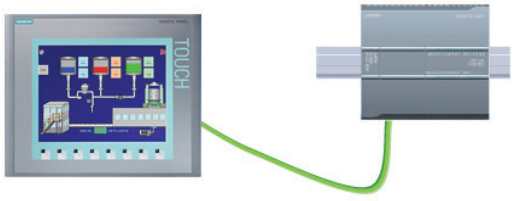
Las conexiones Open User Communication, conexión S7, HMI, programadora, OPC UA y servidor web pueden utilizar varios de los recursos de conexión dependiendo de las funciones que se estén utilizando en esos momentos.

11.5 PROFINET

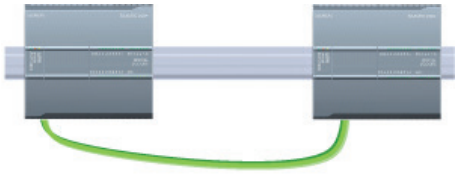
La CPU se puede comunicar con otras CPUs, con programadoras, con dispositivos HMI y con dispositivos no Siemens que utilicen protocolos de comunicación TCP estándar.



Programadora conectada a la CPU



HMI conectado a la CPU



Una CPU conectada a otra CPU

Conmutación Ethernet

Las CPU 1211C, 1212C y 1214C tienen un solo puerto Ethernet y no incluyen un switch Ethernet integrado. Una conexión directa entre una programadora o HMI y una CPU no requiere un conmutador Ethernet. Sin embargo, una red con más de dos CPUs o dispositivos HMI sí requiere un conmutador Ethernet.



① CPU 1215C

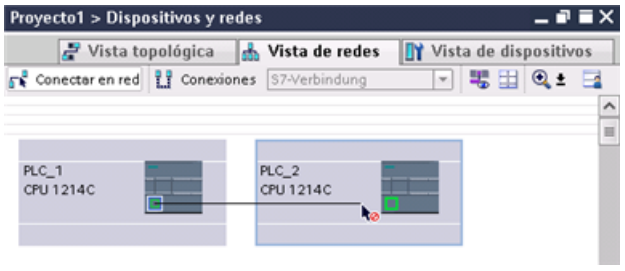
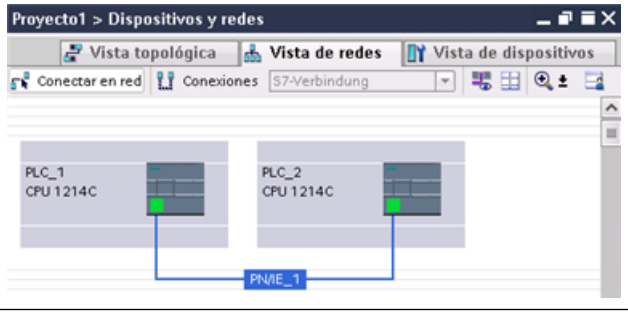
② CSM1277 Ethernet switch

La CPU 1215C y la CPU 1217C cuentan con un switch Ethernet de 2 puertos incorporado. Se puede tener una red con una CPU 1215C y otras dos CPU S7-1200. También puede utilizarse el switch Ethernet de 4 puertos CSM1277 de montaje en rack para conectar varias CPU y dispositivos HMI.

11.5.1 Crear una conexión de red

Utilice la "Vista de red" de la "Configuración de dispositivos" para crear las conexiones de red entre los dispositivos del proyecto. Tras crear la conexión de red, utilice la ficha "Propiedades" de la ventana de inspección para configurar los parámetros de la red.

Tabla 11-4 Crear una conexión de red

Acción	Resultado
Seleccione "Vista de red" para visualizar los dispositivos que deben conectarse.	
Seleccione el puerto de uno de los dispositivos y arrastre la conexión hasta el puerto del otro dispositivo.	
Suelte el botón del ratón para crear la conexión de red.	

11.5.2 Configurar la vía de conexión local/de interlocutor

Una conexión local/interlocutor (remoto) define una asignación lógica de dos interlocutores para establecer servicios de comunicación. Una conexión define lo siguiente:

- Interlocutores involucrados (uno activo, otro pasivo)
- Tipo de conexión (por ejemplo una conexión de PLC, HMI o dispositivo)
- Ruta de conexión

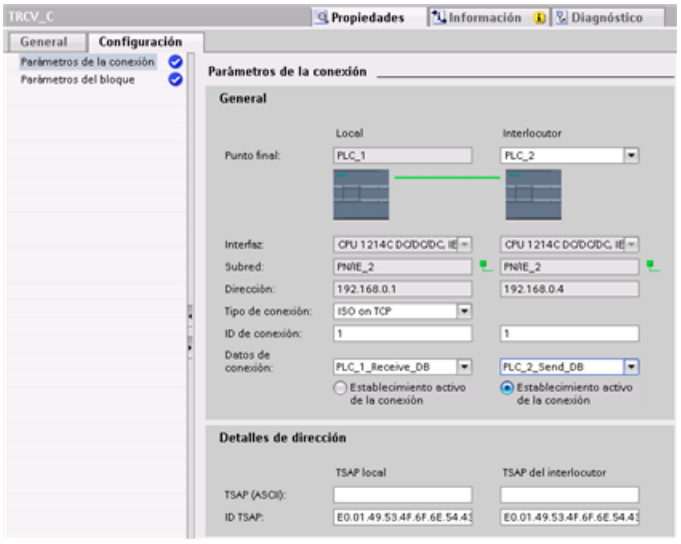
Los interlocutores ejecutan las instrucciones necesarias para configurar y establecer la conexión. El punto final activo y el punto final pasivo de la comunicación se especifican mediante parámetros. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

Si la conexión finaliza (p. ej. debido a una interrupción de la línea), el interlocutor activo intenta restablecer la conexión configurada. No es necesario volver a ejecutar la instrucción.

Rutas de conexión

Después de insertar una instrucción TSEND_C, TRCV_C o TCON en el programa de usuario, la ventana del inspector mostrará las propiedades de la conexión una vez que haya seleccionado cualquier parte de la instrucción. Los parámetros de comunicación se especifican en la ficha "Configuración" de "Propiedades" de la instrucción de comunicación.

Tabla 11-5 Configurar la ruta de conexión (mediante las propiedades de la instrucción)

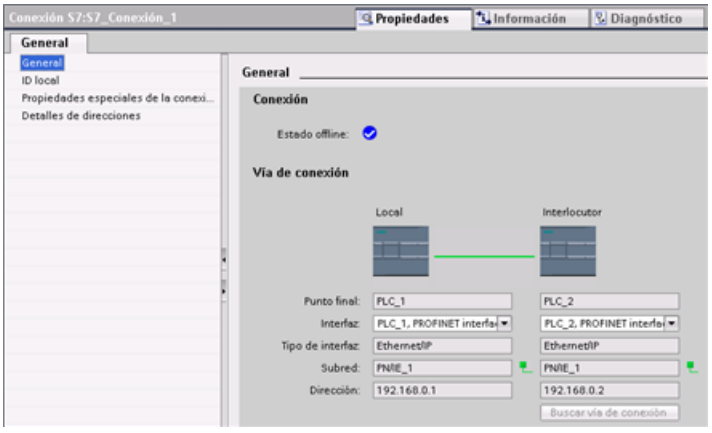
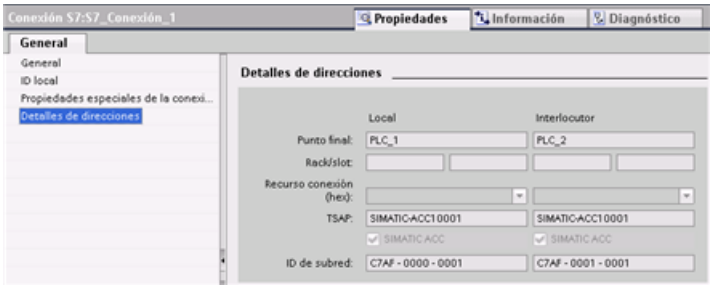
TCP, ISO on TCP y UDP	Propiedades de la conexión
Para los protocolos de Ethernet TCP, ISO on TCP y UDP, utilice las "Propiedades" de la instrucción (TSEND_C, TRCV_C o TCON) para configurar las conexiones "local/interlocutor". La figura muestra las "Propiedades de conexión" de la ficha "Configuración" en el caso de una conexión ISO on TCP.	

Nota

Cuando se configuran las propiedades de conexión de una CPU, STEP 7 permite seleccionar un DB de conexión específico en la CPU interlocutora (si hay alguno) o bien crear el DB de conexión para la CPU interlocutora. La CPU interlocutora ya debe estar creada en el proyecto y no puede ser una CPU "sin especificar".

Todavía hay que insertar una instrucción TSEND_C, TRCV_C o TCON en el programa de usuario de la CPU interlocutora. Al insertar la instrucción, seleccione el DB de conexión que se creó durante la configuración.

Tabla 11-6 Configurar la ruta de conexión para la comunicación S7 (configuración del dispositivo)

Comunicación S7 (GET y PUT)	**Propiedades de la conexión**
Para la comunicación S7, utilice el editor "Dispositivos y redes" de la red para configurar las conexiones local/interlocutor. Haga clic en el botón "Resaltado: conexión" para acceder a las "Propiedades". La ficha "General" ofrece varias propiedades: • "General" (la que se muestra) • "ID local" • "Propiedades de conexión especiales" • "Detalles de direcciones" (la que se muestra)	 

Consulte los "Protocolos" (Página 613) de la sección "PROFINET" o "Crear una conexión S7" (Página 799) en la sección "Comunicación S7 para obtener más información y una lista de las instrucciones de comunicación disponibles.

Tabla 11-7 Parámetros para la conexión múltiple de CPU

Parámetro		**Definición**
Dirección		Direcciones IP asignadas
General	Punto final	Nombre asignado a la CPU interlocutora (receptora)
	Interfaz	Nombre asignado a las interfaces
	Subred	Nombre asignado a las subredes
	Tipo de interfaz	*Solo comunicación S7*: Tipo de interfaz
	Tipo de conexión	Tipo de protocolo Ethernet
	ID de conexión	Número de ID
	Datos de conexión	Ubicación de almacenamiento de datos de las CPUs local e interlocutora
	Establecer una conexión activa	Botón de opción para seleccionar la CPU local o interlocutora como conexión activa

Parámetro		Definición
Detalles de dirección	Punto final	*Solo comunicación S7*: Nombre asignado a la CPU interlocutora (receptora)
	Rack/slot	*Solo comunicación S7*: Ubicación de rack y slot
	Recurso de conexión	*Solo comunicación S7*: Componente de TSAP que se utiliza en la configuración de una conexión S7 con una CPU S7-300 o S7-400
	Puerto (decimal):	TCP y UPD: Puerto de la CPU interlocutora en formato decimal
	TSAP[1] e ID de subred:	ISO on TCP (RFC 1006) y comunicación S7: TSAPs de las CPUs local e interlocutora en formato ASCII y hexadecimal

[1] Al configurar una conexión con una CPU S7-1200 para ISO on TCP, utilice solo caracteres ASCII en la extensión TSAP para los interlocutores pasivos.

Transport Service Access Points (TSAPs)

El uso de TSAP, el protocolo ISO on TCP y la comunicación S7 permite conexiones múltiples con una dirección IP única. Los TSAPs identifican unívocamente estas conexiones de puntos finales de comunicación a una dirección IP.

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los TSAPs que deben utilizarse. El TSAP de una conexión en la CPU se introduce en el campo "TSAP local". El TSAP asignado a la conexión en la CPU interlocutora se introduce en el campo "TSAP del interlocutor".

Números de puerto

Con los protocolos TCP y UDP, la configuración de los parámetros de conexión de la CPU local (activa) debe especificar la dirección IP remota y el número de puerto de la CPU interlocutora (pasiva).

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los puertos que deben utilizarse. El puerto de una conexión en la CPU se introduce en el campo "Puerto local". El puerto asignado a la conexión en la CPU interlocutora se introduce en el campo "Puerto del interlocutor".

11.5.3 Asignar direcciones IP (Internet Protocol)

11.5.3.1 Asignar direcciones IP a los dispositivos de programación y red

Si la programadora utiliza un adaptador integrado conectado a la VLAN de la planta, tanto la programadora como la CPU deben estar en la misma subred. La subred se asigna combinando la dirección IP con la máscara de subred del dispositivo. Consulte a su administrador de red local si necesita ayuda.

El ID de red equivale a los tres primeros octetos de la dirección IP, p. ej. **211.154.184**.16. Este ID de red identifica la red IP de forma unívoca. La máscara de subred suele tener un valor de **255.255.255.0**. No obstante, puesto que el equipo está integrado en una LAN corporativa, la máscara de subred puede tener distintos valores, p. ej. **255.255.254.0**, para configurar subredes

unívocas. La máscara de subred, combinada con la dirección IP del dispositivo y una combinación lógica Y, define los límites de una subred IP.

Nota

En Internet, puesto que las programadoras, dispositivos de red y routers IP se comunican con el mundo entero, debe asignar direcciones IP unívocas para evitar conflictos con otros usuarios de la red. Contacte con los especialistas del departamento IT de su empresa, que están familiarizados con la red corporativa, para asignar las direcciones IP.

⚠ ADVERTENCIA

Acceso no autorizado a la CPU mediante el servidor web

Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU (Página 163) protegidos por contraseña e ID de usuario de servidor web (Página 848) con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.
- Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida.

Nota

Un adaptador de red secundario sirve para cuando la CPU no debe estar en la LAN corporativa. Esta opción es especialmente útil durante la comprobación inicial o los tests de puesta en marcha.

Asignar o comprobar la dirección IP de la programadora utilizando "Mis sitios de red" (en el Escritorio)

Para asignar o comprobar la dirección IP de las programadoras, proceda del siguiente modo:

1. Abra el panel de control desde el menú Inicio.
2. Abra el "Centro de redes y recursos compartidos" y elija "Conexión de área local" para el adaptador de red conectado a la CPU
3. Haga clic en "Propiedades" en el cuadro de diálogo "Estado de conexión de área local".
4. En el cuadro de diálogo "Propiedades de conexión de área local", elija "Versión de protocolo de Internet 4 (TCP/IPv4)" para el campo "Esta conexión usa los siguientes elementos:".
5. Haga clic en el botón "Propiedades".
6. Seleccione "Obtener una dirección IP automáticamente" o seleccione "Usar la siguiente dirección IP" para introducir una dirección IP estática.
7. Si ha seleccionado "Usar la siguiente dirección IP", configure la dirección IP y la máscara de subred:
   - Ajuste la dirección IP para que utilice el mismo ID de red y la misma submáscara que la CPU. Por ejemplo, si la dirección IP de la CPU es **192.168.0**.1, se puede ajustar la dirección IP a **192.168.0**.200.
   - Seleccione una máscara de subred **255.255.255**.0.
   - Deje la pasarela predeterminada vacía.

Ahora ya se puede conectar a la CPU.

Nota

La tarjeta de interfaz de red y la CPU deben estar en la misma subred para que STEP 7 encuentre la CPU y se comunique con ella.

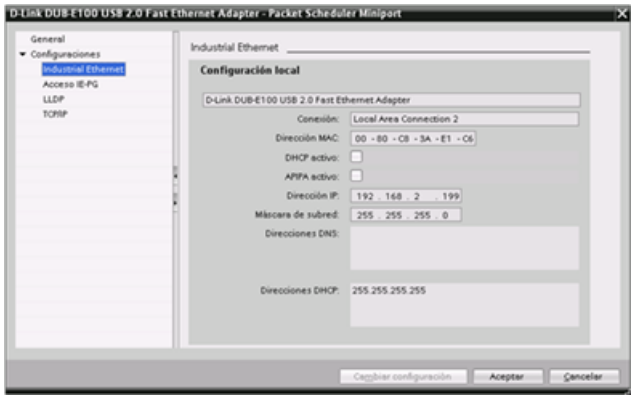
Consulte el personal de TI para que le ayude a configurar la red de modo que sea posible conectarse con la CPU S7-1200.

11.5.3.2 Comprobar la dirección IP de la programadora

Las direcciones MAC e IP de la programadora pueden consultarse con los siguientes comandos de menú:

1. En el "Árbol del proyecto", expanda "Acceso online".
2. Haga clic con el botón derecho en la red en cuestión y seleccione "Propiedades".
3. En el diálogo de la red, expanda "Configuraciones" y seleccione "Industrial Ethernet".

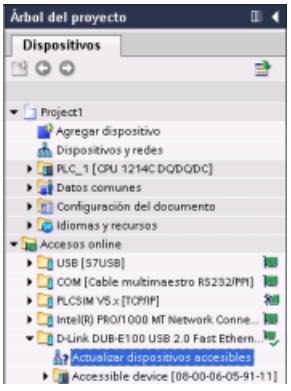
Se muestran las direcciones MAC e IP de la programadora.



11.5.3.3 Asignar una dirección IP a una CPU online

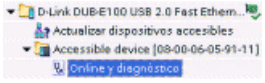
Es posible asignar una dirección IP a un dispositivo de red online. Esto es especialmente útil al configurar los dispositivos por primera vez.

1. En el árbol del proyecto, verifique que la CPU no tiene una dirección IP configurada. Expanda "Acceso online" > <Tarjeta de adaptador para la red en la que está ubicado el dispositivo y haga doble clic en "Actualizar dispositivos accesibles".

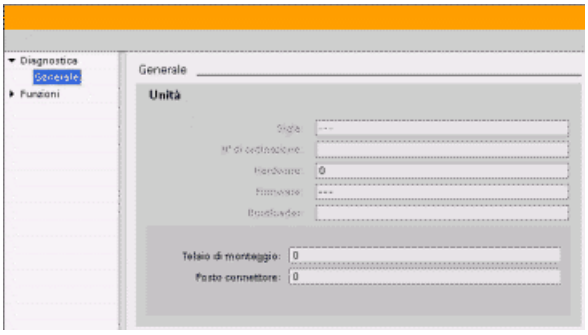


Si STEP 7 muestra una dirección MAC en lugar de una dirección IP, no se ha asignado ninguna dirección IP.

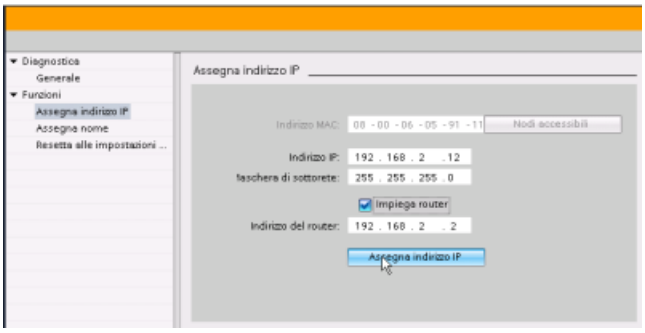
2. En el dispositivo accesible en cuestión, haga doble clic en "Online y diagnóstico"



3. En el cuadro de diálogo "Online y diagnóstico", elija "Funciones" > "Asignar dirección IP".

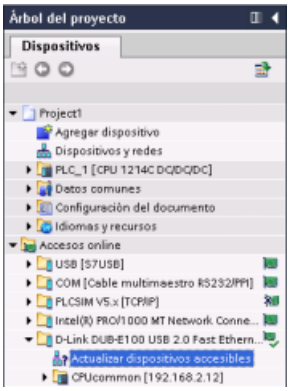


4. En el campo "Dirección IP", introduzca la nueva dirección IP y haga clic en el botón "Asignar dirección IP".



5. En el árbol del proyecto, verifique que STEP 7 ha asignado la nueva dirección IP a la CPU.

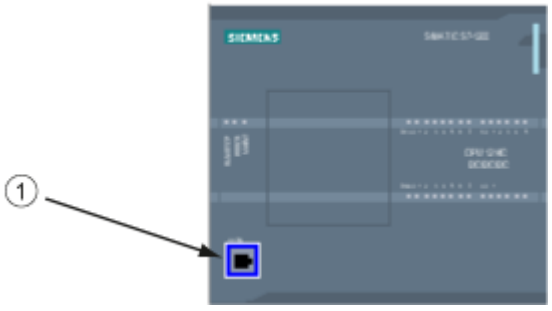
Haga doble clic en "Actualizar dispositivos accesibles" para visualizar la dirección IP que ha configurado.



11.5.3.4 Configurar una dirección IP para una CPU en el proyecto

Configurar la interfaz PROFINET

Para configurar parámetros para la interfaz PROFINET, seleccione la casilla PROFINET verde en la CPU. La ficha "Propiedades" de la ventana de inspección muestra el puerto PROFINET.



① Puerto PROFINET

Configurar la dirección IP

Dirección Ethernet (MAC): Todo dispositivo de una red PROFINET recibe una dirección MAC (Media Access Control o control de acceso al medio) del fabricante para su identificación. Una dirección MAC consta de seis grupos de dos dígitos hexadecimales, separados por guiones (-) o dos puntos (:), en orden de transmisión (p. ej. 01-23-45-67-89-AB o 01:23:45:67:89:AB).

Dirección IP: Todo dispositivo debe tener también una dirección IP (Internet Protocol o Protocolo Internet). Esta dirección permite al dispositivo transferir datos a través de una red enrutada y más compleja.

Toda dirección IP se divide en cuatro segmentos de ocho bits (octetos) y se expresa en formato decimal separado por puntos (p. ej. 211.154.184.16). La primera parte de la dirección IP se utiliza para la ID de red (¿en qué red se encuentra?) y, la segunda, para la ID del host (unívoca para cada dispositivo de la red). Una dirección IP 192.168.x.y es una designación estándar reconocida como parte de una red privada que no se enruta vía Internet.

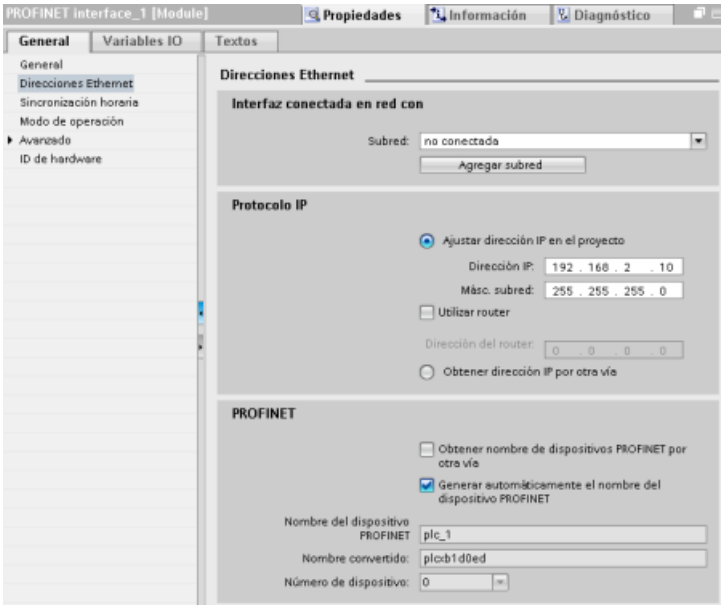
Máscara de subred: Una subred es una agrupación lógica de dispositivos de red conectados. Generalmente, los nodos de una subred están próximos físicamente en una red de área local (LAN). Una máscara (denominada "máscara de subred" o "máscara de red") define los límites de una subred IP.

Generalmente, una máscara de subred 255.255.255.0 se adecúa para una red local pequeña. Esto significa que los 3 primeros octetos de todas las direcciones IP de esta red deberían ser iguales. Los diferentes dispositivos de la red se identifican mediante el último octeto (campo de 8 bits). Por ejemplo, es posible asignar la máscara de subred 255.255.255.0 y direcciones IP comprendidas entre 192.168.2.0 y 192.168.2.255 a los dispositivos de una red local pequeña.

La única conexión entre las diferentes subredes se realiza a través de un router. Si se utilizan subredes, es preciso utilizar un router IP.

Router IP: Los routers interconectan las distintas LANs. Si se utiliza un router, un equipo de una LAN puede enviar mensajes a otras redes que, a su vez, pertenezcan a otras LANs. Si el destino de los datos se encuentra fuera de la LAN, el router reenvía los datos a otra red o grupo de redes desde donde pueden transferirse a su destino.

Los routers necesitan direcciones IP para poder transferir y recibir paquetes de datos.



Propiedades de direcciones IP: En la ventana de propiedades, seleccione la entrada de configuración "Direcciones Ethernet". STEP 7 muestra el diálogo de configuración de direcciones Ethernet, en el que el proyecto de software se asocia a la dirección IP de la CPU que lo recibirá.

Tabla 11-8 Parámetros para la dirección IP

Parámetro		Descripción
Subred	Nombre de la subred a la que está conectada el dispositivo. Haga clic en el botón "Agregar nueva subred" para crear una subred nueva. La opción predeterminada es "No conectado". Son posibles dos tipos de conexión: • El ajuste predeterminado "no conectado" ofrece una conexión local. • Una subred se requiere cuando la red comprende dos o más dispositivos.	
Protocolo IP	Dirección IP	Dirección IP asignada a la CPU
	Máscara de subred	Máscara de subred asignada
	Utilizar router IP	Haga clic en esta casilla de verificación para indicar el uso de un router IP
	Dirección del router	Dirección IP asignada al router (si es aplicable)

Nota

Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo. Dado que la CPU no dispone de una dirección IP preconfigurada, es necesario asociar el proyecto con la dirección MAC del dispositivo de destino. Si la CPU está conectada a un router de la red, también es preciso introducir la dirección IP del router.

El botón de opción "Asignar dirección IP por otra vía" permite cambiar la dirección IP online o mediante la instrucción "T_CONFIG (Página 712)" después de descargar el programa. Este método de asignación de la dirección IP solo sirve para la CPU.

⚠ ADVERTENCIA

Carga de una configuración de hardware con "Asignar dirección IP por otra vía"

Tras descargar una configuración de hardware con la opción "Asignar dirección IP por otra vía" habilitada, no se puede conmutar el estado operativo de la CPU de RUN a STOP o de STOP a RUN.

El equipo del usuario seguirá funcionando en estas condiciones y puede causar operaciones imprevistas de la máquina o el proceso, provocando como resultado daños materiales, lesiones corporales graves o incluso la muerte si no se toman las medidas adecuadas.

Asegúrese de que las direcciones IP de la CPU se han establecido anteriormente utilizando la CPU en un entorno de automatización actual. Esto puede hacerse utilizando el paquete de programación STEP 7, la herramienta SIMATIC Automation Tool o un dispositivo HMI conectado conjuntamente con la instrucción T_CONFIG.

⚠ ADVERTENCIA

Condición en la que la red PROFINET puede detenerse

Al modificar la dirección IP de una CPU online o desde el programa de usuario, puede crear una condición en la que se detenga la red PROFINET.

Si la dirección IP de una CPU cambia a una dirección IP fuera de la subred, la red PROFINET perderá la comunicación y se detendrá todo intercambio de datos. Existe la posibilidad de configurar el equipo del usuario para que siga funcionando en estas condiciones. La pérdida de la comunicación PROFINET puede causar operaciones imprevistas de la máquina o el proceso con el resultado de muerte, lesiones corporales graves o daños materiales si no se toman las precauciones adecuadas.

Si hay que cambiar manualmente la dirección IP, asegúrese de que la nueva dirección IP está dentro de la subred.

Configurar el puerto PROFINET

En el ajuste predeterminado, la CPU configura el o los puertos de la interfaz PROFINET para autonegotiation. Para que la autonegotiation funcione debidamente, hay que configurar las dos estaciones para que negocien. Si una estación tiene una configuración fija (p. ej. dúplex a 100 Mbps) y la otra estación está ajustada para negociar, la autonegotiation falla y el resultado es una operación semidúplex.

Para superar esta limitación de la autonegotiation, el S7-1200 ofrece una opción que la desactiva. Al desactivar la autonegotiation, el S7-1200 se configura automáticamente para la operación dúplex a 100 Mbps.

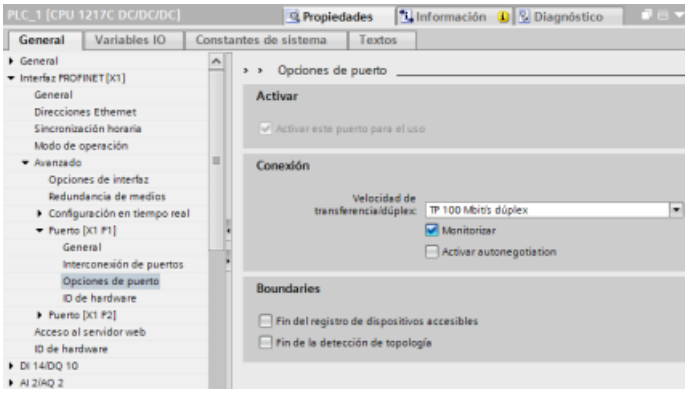
Es posible ajustar la velocidad de transferencia y el modo dúplex a un valor fijo para cada puerto:

1. Seleccione las opciones avanzadas y el puerto que desea configurar. A continuación, seleccione las opciones de puerto.
2. En el campo Conexión, Velocidad de transferencia / dúplex, seleccione una de las opciones siguientes:
   - Automático: La CPU y el dispositivo parejo determinan la velocidad de transferencia y el modo dúplex del puerto mediante autonegotiation.
   - TP 100 Mbps dúplex: Si se desactiva la autonegotiation, el puerto opera a 100 Mbps dúplex. Si se activa la autonegotiation, el puerto puede operar a 100 Mbps dúplex o a otra velocidad de transferencia / dúplex que se negocie entre la CPU y el dispositivo parejo (que deposita un aviso en el búfer de diagnóstico si está seleccionado "Monitor" (véase más abajo)).

3. Monitor: Cuando se selecciona esta casilla de verificación, se deposita un aviso en el búfer de diagnóstico en caso de que se produzca uno de los casos siguientes en el puerto:
   - No es posible establecer una conexión en el puerto
   - Una conexión establecida falla
   - Se selecciona "TP 100 Mbps dúplex" para la velocidad de transferencia / dúplex y la CPU establece una conexión utilizando autonegotiation con una velocidad de transferencia negociada diferente de 100 Mbps o el modo dúplex negociado igual a semidúplex.
4. Activar autonegotiation: Una vez se ha establecido el campo de velocidad de transferencia / dúplex a dúplex a 100 Mbps ya puede desactivarse la autonegotiation. Deseleccione la casilla de verificación "Activar autonegotiation" para desactivar la autonegotiation.

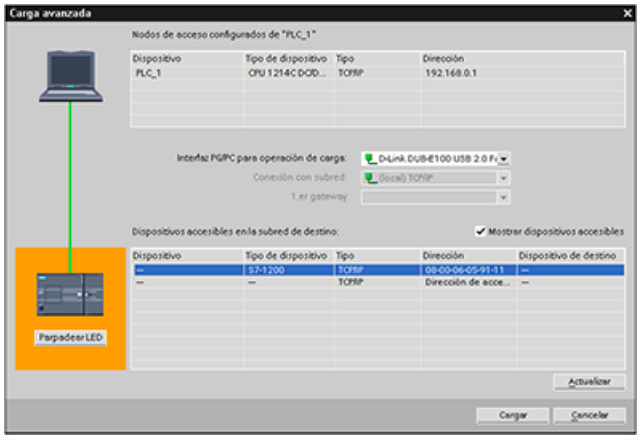
Nota

Si no se desactiva la autonegotiation, la CPU y el dispositivo parejo negocian la velocidad de transferencia y el modo dúplex del puerto.



11.5.4 Comprobar la red PROFINET

Tras finalizar la configuración, cargue el proyecto (Página 199) en la CPU. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.



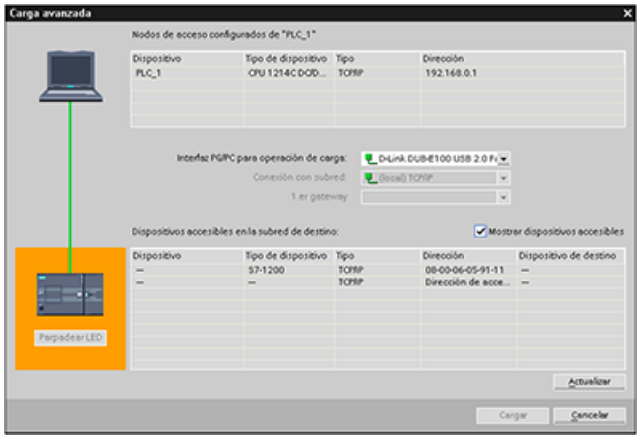
Asignar una dirección IP a un dispositivo online

La CPU S7-1200 no tiene dirección IP preconfigurada. La dirección IP de la CPU se debe asignar manualmente:

- Para asignar una dirección IP a un dispositivo online, véase "Configuración de dispositivos: Asignar una dirección IP a una CPU online" (Página 602) para conocer el procedimiento paso a paso.
- Para asignar una dirección IP en el proyecto, es preciso configurarla en la "Configuración de dispositivos", guardar la configuración y cargarla en el PLC. Véase "Configuración de dispositivos: Configurar una dirección IP para una CPU en el proyecto" (Página 603) para más información.

Utilizar el diálogo "Carga avanzada" para comprobar los dispositivos de red conectados

La función de la CPU S7-1200 "Cargar en dispositivo" y su diálogo "Carga avanzada" permiten visualizar todos los dispositivos de red accesibles y verificar si se han asignado direcciones IP unívocas a todos ellos. Para visualizar todos los dispositivos accesibles y disponibles con sus respectivas direcciones MAC o IP asignadas, active la casilla de verificación "Mostrar dispositivos accesibles".



Si el dispositivo de red deseado no se encuentra en esta lista, la comunicación con ese dispositivo se habrá interrumpido por algún motivo. En este caso es preciso examinar el dispositivo y la red para buscar errores de hardware y/o configuración.

11.5.5 Localizar la dirección Ethernet (MAC) en la CPU

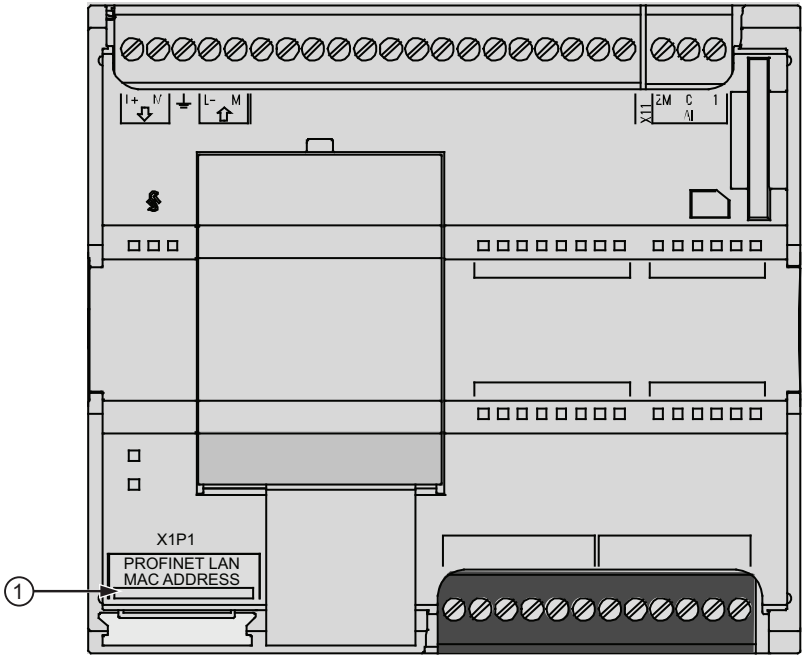
En las redes PROFINET, una dirección MAC (Media Access Control o control de acceso al medio) es un identificador que el fabricante asigna a la interfaz de red para su identificación. Generalmente, una dirección MAC codifica el número de identificación registrado del fabricante.

El formato estándar (IEEE 802.3) permite imprimir direcciones MAC de forma amigable y consta de seis grupos de dos dígitos hexadecimales, separados por guiones (-) o dos puntos (:), en orden de transmisión (p. ej. 01-23-45-67-89-ab ó 01:23:45:67:89:ab).

Nota

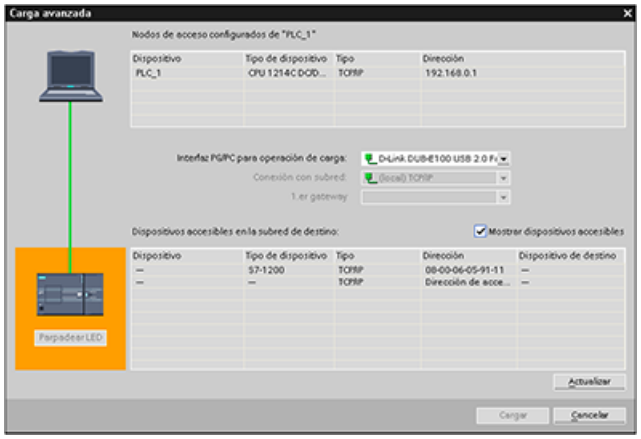
Toda CPU se suministra de fábrica con una dirección MAC unívoca y permanente. La dirección MAC de la CPU no se puede cambiar.

La dirección MAC está impresa en la esquina inferior izquierda en el frente de la CPU. Para ver la información de la dirección MAC es necesario abrir la tapa inferior.



① Dirección MAC

Inicialmente, la CPU no tiene dirección IP, sino sólo una dirección MAC ajustada de fábrica. Para la comunicación PROFINET es necesario que todos los dispositivos tengan asignada una dirección IP unívoca.



La función de la CPU "Cargar en dispositivo" y el diálogo "Carga avanzada en dispositivo" permiten visualizar todos los dispositivos de red accesibles y asegurar que se han asignado direcciones IP unívocas a todos ellos. Este diálogo muestra todos los dispositivos accesibles y disponibles con sus respectivas direcciones MAC o IP asignadas. Las direcciones MAC son especialmente importantes para identificar dispositivos que no disponen de la dirección IP unívoca necesaria.

11.5.6 Configurar la sincronización del Network Time Protocol (NTP)

⚠ ADVERTENCIA

Riesgo de ataques a las redes mediante la sincronización del Network Time Protocol (NTP)

Si un atacante puede acceder a las redes por sincronización del Network Time Protocol (NTP), posiblemente alterará el control del proceso cambiando la hora del sistema de la CPU. Las alteraciones en el control del proceso pueden causar la muerte, heridas graves o daños materiales.

La CPU S7-1200 deshabilita la función de cliente NTP de forma predeterminada. Si habilita la función NTP, solo la dirección IP que ha configurado podrá actuar como servidor NTP. Hay que configurar la función NTP de modo que permita corregir la hora del sistema de la CPU desde servidores remotos.

La CPU S7-1200 soporta alarmas horarias e instrucciones de reloj que dependen de la precisión de la hora del sistema de la CPU. Si se configura NTP y se acepta la sincronización horaria desde un servidor, hay que asegurarse de que el servidor es una fuente fiable. Si no se comprueba, se puede producir una brecha de seguridad que permita a un usuario desconocido alterar el control del proceso cambiando la hora del sistema de la CPU.

Para más información sobre seguridad y recomendaciones, consulte nuestras "Guías operacionales sobre seguridad industrial" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) en la página de atención al cliente de Siemens.

El Network Time Protocol (NTP) se utiliza extensamente para sincronizar los relojes de sistema de los equipos con servidores horarios de Internet. En el modo NTP, la CPU envía consultas de la hora en intervalos regulares (en modo de cliente) al servidor NTP de la subred (LAN). A partir de

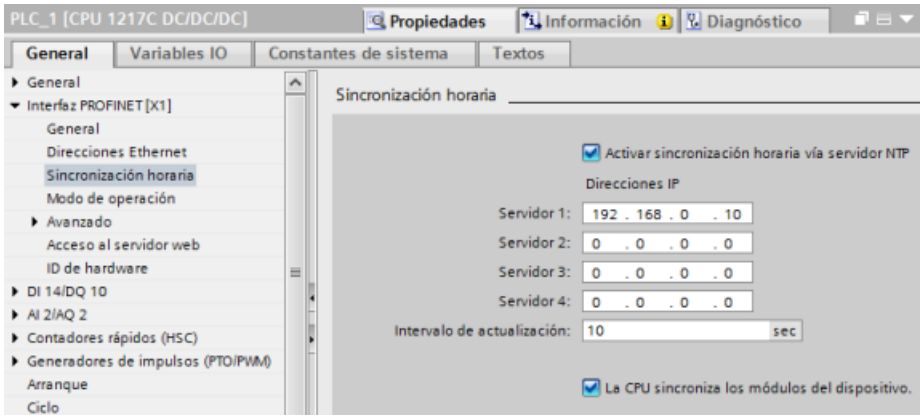
las respuestas del servidor, se calcula la hora más probable y precisa y se sincroniza la hora en la estación.

La ventaja de este método radica en que permite sincronizar la hora entre subredes.

Se configuran las direcciones IP de hasta cuatro servidores NTP. El intervalo de actualización define el tiempo entre las consultas de la hora (en segundos). El valor del intervalo puede variar entre 10 segundos y un día.

En el modo NTP los servidores generalmente transfieren la hora UTC (Universal Time Coordinated o tiempo universal coordinado), que corresponde a GMT (Greenwich Mean Time o tiempo medio de Greenwich).

En la ventana de propiedades de la configuración del dispositivo CPU, elija la entrada "Sincronización horaria". STEP 7 muestra el diálogo de configuración de la sincronización horaria:



Nota

La CPU recibe todas las direcciones IP al descargar el proyecto.

Tabla 11-9 Parámetros de la sincronización horaria

Parámetro	Definición
Activar sincronización horaria vía servidor NTP	Seleccione la casilla de verificación para activar la sincronización horaria vía servidor NTP.
Servidor 1	Dirección IP asignada al servidor de hora de red 1
Servidor 2	Dirección IP asignada al servidor de hora de red 2
Servidor 3	Dirección IP asignada al servidor de hora de red 3
Servidor 4	Dirección IP asignada al servidor de hora de red 4
Intervalo de actualización de la sincronización horaria	Valor del intervalo (seg)
La CPU sincroniza los módulos del dispositivo.	Seleccione la casilla de verificación para sincronizar el reloj del CP con el reloj de la CPU.

11.5.7 Tiempo de arranque de dispositivo, asignación de nombre y de dirección en PROFINET

PROFINET IO puede ampliar el tiempo de arranque del sistema (timeout configurable). Un número mayor de dispositivos o los dispositivos lentos influyen en el tiempo de arranque necesario para conmutar a RUN.

En V4.0 y posterior, puede tener 16 dispositivos PROFINET IO como máximo en la red S7-1200 PROFINET.

Cada estación (o dispositivo IO) arranca independientemente en el arranque, y esto afecta al tiempo global de arranque de la CPU. Si para el timeout configurable se ajusta un valor muy bajo, el tiempo global de arranque de la CPU puede no ser suficiente para que todas las estaciones completen su arranque. Si eso ocurre, aparecerán errores de estación no reales.

En "Propiedades de la CPU", bajo "Arranque", encontrará el "Tiempo de asignación de los parámetros para E/S distribuidas" (timeout). El timeout predeterminado configurable es 60.000 ms (1 minuto), pero el usuario puede modificarlo.

Asignación de nombre y dirección de dispositivo PROFINET en STEP 7

Todos los dispositivos PROFINET **deben** tener un nombre de dispositivo y una dirección IP. Utilice STEP 7 para especificar los nombres de dispositivo y configurar las direcciones IP. Los nombres de dispositivo se descargan a los dispositivos IO por medio de PROFINET DCP (Discovery and Configuration Protocol).

Asignación de dirección PROFINET en el arranque del sistema

El controlador transmite los nombres de los dispositivos a la red, y estos responden con sus direcciones MAC. Entonces el controlador asigna una dirección IP al dispositivo utilizando el protocolo PROFINET DCP:

- Si la dirección MAC tiene una dirección IP configurada, la estación arranca.
- Si la dirección MAC no tiene una dirección IP configurada, STEP 7 asigna la dirección configurada en el proyecto y, entonces, la estación arranca.
- Si se produce un problema durante este proceso, se produce un error de estación y no hay arranque. La situación provoca un rebase del timeout configurable.

11.5.8 Open User Communication

11.5.8.1 Protocolos

El puerto PROFINET integrado de la CPU soporta múltiples estándares de comunicación a través de una red Ethernet:

- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- User Datagram Protocol (UDP)

Tabla 11-10 Protocolos e instrucciones de comunicación para cada uno

Protocolo	Ejemplos de uso	Entrada de datos en el área de recepción	Instrucciones de comunicación	Tipo de direccionamiento
TCP	Comunicación de CPU a CPU Transporte de tramas	Modo ad hoc	Solo TRCV_C y TRCV	Asigna números de puerto a los dispositivos local (activo) e interlocutor (pasivo)
		Recepción de datos con la longitud especificada	TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV	
ISO on TCP	Comunicación de CPU a CPU Fragmentación y reensamblado de mensajes	Modo ad hoc	Solo TRCV_C y TRCV	Asigna TSAPs a los dispositivos local (activo) e interlocutor (pasivo)
		Controlado por protocolo	TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV	
UDP	Comunicación de CPU a CPU Comunicaciones de programa de usuario	User Datagram Protocol	TUSEND y TURCV	Asigna números de puerto a los dispositivos local (activo) e interlocutor (pasivo), pero no es una conexión fija
Comunicación S7	Comunicación de CPU a CPU Leer/escribir datos de/en una CPU	Transmisión y recepción de datos con la longitud especificada	GET y PUT	Asigna TSAPs a los dispositivos local (activo) e interlocutor (pasivo)
PROFINET IO	Comunicación de CPU a dispositivo PROFINET IO	Transmisión y recepción de datos con la longitud especificada	Incorporado	Incorporado

11.5.8.2 TCP y ISO on TCP

Transport Control Protocol (TCP) es un protocolo estándar descrito por RFC 793: Transmission Control Protocol. El objetivo principal de TCP es ofrecer un servicio de conexión seguro y fiable entre pares de procesos. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente puesto que está vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- Ofrece numerosas prestaciones más a las aplicaciones, además de una recuperación de errores, control de flujo y fiabilidad considerables.
- Protocolo orientado a la conexión
- Puede utilizarse muy flexiblemente con sistemas de terceros que soporten únicamente TCP

- Apto para routing
- Son aplicables sólo las longitudes de datos estáticas.
- Los mensajes se acusan.
- Las aplicaciones se direccionan usando números de puerto.
- La mayoría de los protocolos de aplicación (p. ej. TELNET y FTP) utilizan TCP.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

International Standards Organization (ISO) on Transport Control Protocol (TCP) (RFC 1006) (ISO on TCP) es un mecanismo que permite portar aplicaciones ISO a la red TCP/IP. Este protocolo tiene las características siguientes:

- Protocolo de comunicación eficiente vinculado estrechamente al hardware
- Adecuado para cantidades de datos medianas y grandes (hasta 8192 bytes)
- A diferencia de TCP, los mensajes tienen un indicador de fin y están orientados a los mensajes.
- Apto para routing; puede utilizarse en WAN
- Las longitudes de datos dinámicas son posibles.
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE.

Puesto que utiliza Transport Service Access Points (TSAPs), el protocolo TCP permite establecer varias conexiones con una sola dirección IP (hasta 64K conexiones). Gracias a RFC 1006, los TSAPs identifican unívocamente estas conexiones de puntos finales de comunicación a una dirección IP.

11.5.8.3 Servicios de comunicación y números de puerto utilizados

La CPU S7-1200 soporta los protocolos listados en la tabla inferior. Para cada protocolo, la CPU asigna los parámetros de dirección, el nivel de comunicación adecuado, la función de comunicación y el sentido de la misma.

Esta información permite adaptar las medidas de seguridad para la protección del sistema de automatización a los protocolos utilizados (como el cortafuegos). Solo las redes Ethernet o PROFINET tienen medidas de seguridad. Puesto que PROFIBUS no tiene medidas de seguridad, la tabla no incluye los protocolos PROFIBUS.

La tabla siguiente muestra los diferentes niveles y protocolos que utiliza la CPU:

Protocolo	Número de puerto	(2) Nivel de enlace (4) Nivel de transporte	Función	Descripción
Protocolos PROFINET				
DCP (Discovery and Configuration Protocol)	Irrelevante	(2) Ethernet II e IEEE 802.1Q y Ethertype 0x8892 (PROFINET)	Detección y configuración PROFINET para dispositivos accesibles	PROFINET utiliza DCP para detectar dispositivos y asignarles los ajustes básicos. DCP utiliza la dirección MAC Multicast especial: xx-xx-xx-01-0E-CF, xx-xx-xx = Identificador único de organización
LLDP (Link Layer Discovery Protocol)	Irrelevante	(2) Ethernet II e IEEE 802.1Q y Ethertype 0x88CC (PROFINET)	Protocolo de detección del nivel de enlace para PROFINET	PROFINET utiliza LLDP para detectar y administrar relaciones de vecindad entre dispositivos PROFINET. LLDP utiliza la dirección MAC Multicast especial: 01-80-C2-00-00-0E

11.5.8.4 Modo ad hoc

Generalmente, TCP e ISO on TCP reciben paquetes de datos con longitud especificada, que varía entre 1 y 8192 bytes. Sin embargo, las instrucciones de comunicación TRCV_C y TRCV también ofrecen un modo de comunicación "ad hoc" que puede recibir paquetes de datos con una longitud variable entre 1 y 1472 bytes.

Nota

Si se guardan los datos en un DB "optimizado" (solo simbólico), únicamente es posible recibir datos en matrices de tipos de datos Byte, Char, USInt y SInt.

Para configurar la instrucción TRCV_C o TRCV para el modo ad hoc, ajuste el parámetro de entrada de la instrucción ADHOC.

Si no se llama la instrucción TRCV_C o TRCV en modo ad hoc con frecuencia, es posible que se reciban varios paquetes en una misma llamada. Ejemplo: si tuviera que recibir cinco paquetes de 100 bytes en una llamada, TCP los transferiría como un solo paquete de 500 bytes, mientras que ISO on TCP reestructuraría los paquetes en cinco de 100 bytes cada uno.

11.5.8.5 ID de conexión para las instrucciones de Open User Communication

Cuando se insertan las instrucciones PROFINET TSEND_C, TRCV_C o TCON en el programa de usuario, STEP 7 crea un DB de instancia para configurar el canal de comunicaciones (o conexión) entre los dispositivos. Utilice las "Propiedades" (Página 596) de la instrucción para configurar los parámetros de la conexión. Entre los parámetros se encuentra la ID de conexión para dicha conexión.

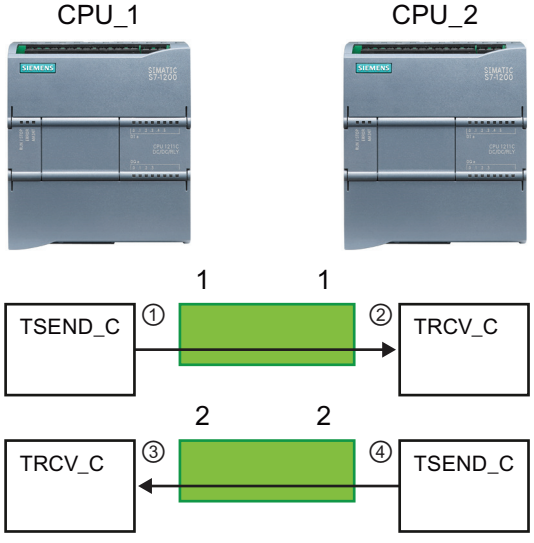
- La ID de conexión debe ser unívoca para la CPU. Cada conexión que se cree debe tener un DB y una ID de conexión distintos.
- Tanto la CPU local como la CPU interlocutora pueden utilizar el mismo número de ID de conexión para la misma conexión, aunque los números de ID de conexión no tienen que coincidir necesariamente. El número de ID de conexión solo es relevante para las instrucciones PROFINET dentro del programa de usuario de la CPU individual.
- Puede utilizarse cualquier número para la ID de conexión de la CPU. Sin embargo, si se configuran las ID de conexión secuencialmente empezando por "1", es más fácil saber el número de conexiones que se están utilizando en una CPU concreta.

Nota

Cada instrucción TSEND_C, TRCV_C o TCON del programa de usuario crea una conexión nueva. Es importante utilizar la ID de conexión correcta para cada conexión.

El ejemplo siguiente muestra la comunicación entre dos CPUs que utilizan dos conexiones separadas para transmitir y recibir datos.

- La instrucción TSEND_C de la CPU_1 enlaza con TRCV_C de la CPU_2 a través de la primera conexión ("ID de conexión 1" tanto en la CPU_1 como en la CPU_2).
- La instrucción TRCV_C de la CPU_1 enlaza con TSEND_C de la CPU_2 a través de la segunda conexión ("ID de conexión 2" tanto en la CPU_1 como en la CPU_2).



① TSEND_C en la CPU_1 crea una conexión y asigna una ID de conexión a dicha conexión (ID de conexión 1 para la CPU_1).

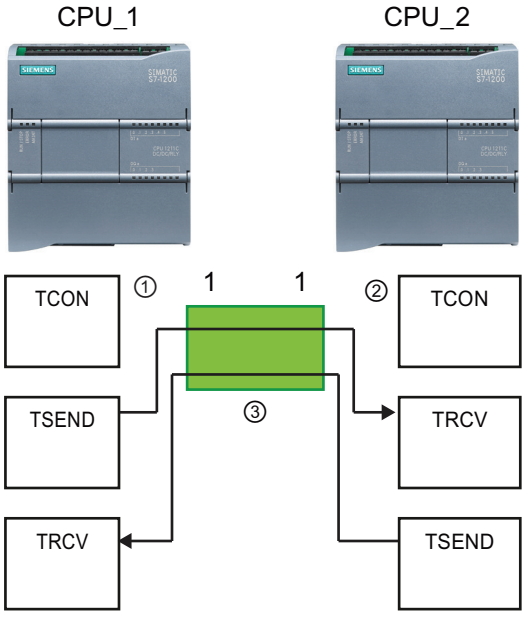
② TRCV_C en la CPU_2 crea la conexión para la CPU_2 y asigna la ID de conexión (ID de conexión 1 para la CPU_2).

③ TRCV_C en la CPU_1 crea una segunda conexión para CPU_1 y asigna una ID de conexión distinta para dicha conexión (ID de conexión 2 para la CPU_1).

④ TSEND_C en la CPU_2 crea una segunda conexión y asigna una ID de conexión distinta para dicha conexión (ID de conexión 2 para la CPU_2).

El ejemplo siguiente muestra la comunicación entre dos CPUs que utilizan 1 conexión tanto para transmitir como para recibir datos.

- Cada CPU utiliza una instrucción TCON para configurar la conexión entre las dos CPUs.
- La instrucción TSEND de la CPU_1 enlaza con la instrucción TRCV de la CPU_2 a través de la ID de conexión ("ID de conexión 1") configurada por la instrucción TCON de la CPU_1. La instrucción TRCV de la CPU_2 enlaza con la instrucción TSEND de la CPU_1 a través de la ID de conexión ("ID de conexión 1") configurada por la instrucción TCON de la CPU_2.
- La instrucción TSEND de la CPU_2 enlaza con la instrucción TRCV de la CPU_1 a través de la ID de conexión ("ID de conexión 1") configurada por la instrucción TCON de la CPU_2. La instrucción TRCV de la CPU_1 enlaza con la instrucción TSEND de la CPU_2 a través de la ID de conexión ("ID de conexión 1") configurada por la instrucción TCON de la CPU_1.



① TCON en la CPU_1 crea una conexión y asigna una ID de conexión a dicha conexión en la CPU_1 (ID=1).

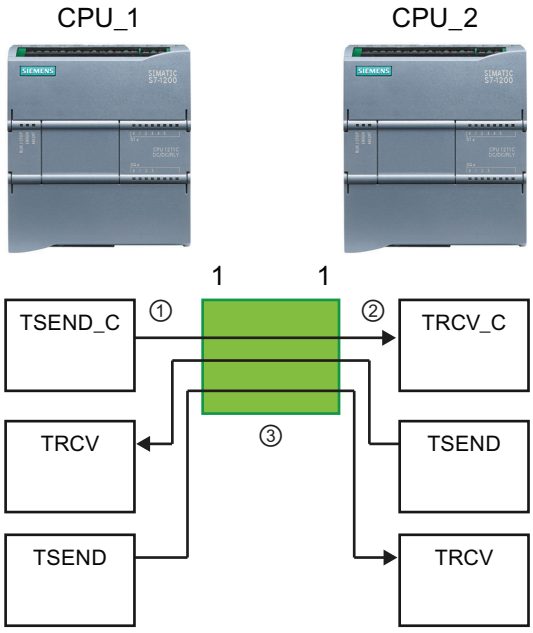
② TCON en la CPU_2 crea una conexión y asigna una ID de conexión a dicha conexión en la CPU_2 (ID=1).

③ TSEND y TRCV en la CPU_1 utilizan la ID de conexión creada por TCON en la CPU_1 (ID=1).

TSEND y TRCV en la CPU_2 utilizan la ID de conexión creada por TCON en la CPU_2 (ID=1).

Tal como se muestra en el ejemplo siguiente, también es posible utilizar instrucciones TSEND y TRCV individuales para comunicarse a través de una conexión creada por una instrucción TSEND_C o TRCV_C. Las instrucciones TSEND y TRCV no crean por sí solas una conexión nueva,

por lo que deben utilizarse el DB y la ID de conexión creados por una instrucción TSEND_C, TRCV_C o TCON.



① TSEND_C en la CPU_1 crea una conexión y asigna una ID de conexión a dicha conexión (ID=1).

② TRCV_C en la CPU_2 crea una conexión y asigna la ID de conexión a dicha conexión en la CPU_2 (ID=1).

③ TSEND y TRCV en la CPU_1 utilizan la ID de conexión creada por TSEND_C en la CPU_1 (ID=1).

TSEND y TRCV en la CPU_2 utilizan la ID de conexión creada por TRCV_C en la CPU_2 (ID=1).

11.5.8.6 Parámetros de la conexión PROFINET

Las instrucciones TSEND_C, TRCV_C y TCON requieren parámetros relacionados con la conexión para poder conectarse con el dispositivo interlocutor. La estructura TCON_Param asigna dichos parámetros para los protocolos TCP, ISO on TCP y UDP. Por norma general, para especificar estos parámetros se utiliza la pestaña "Configuración" (Página 596) de las "Propiedades" de la instrucción. Si la pestaña "Configuración" no está disponible, hay que suministrar la estructura TCON_Param en los parámetros de la instrucción.

Con V4.1 o posterior, la estructura TCON_IP_V4 asigna parámetros para el protocolo TCP, y la estructura TCON_IP_RFC asigna parámetros para el protocolo ISO on TCP.

Con V4.3 o posterior, la estructura TCON_IP_V4_SEC asigna parámetros adicionales para el protocolo TCP. Con el fin de establecer una comunicación TCP segura entre dos CPU S7-1200 hay que crear un bloque de datos con el tipo de datos de sistema TCON_IP_V4_SEC o, en cada CPU, realizar la parametrización y llamarlo directamente en la instrucción. Las instrucciones TCON, TSEND_C y TRCV_C soportan el tipo de datos de sistema TCON_IP_V4_SEC.

Con V4.4 o posterior, use las estructuras TCON_QDN y TCON_QDN_SEC para configurar las conexiones de comunicación para TCP y UDP mediante el nombre de dominio plenamente cualificado (fully qualified domain name) y use la estructura TCON_QDN_SEC para configurar las conexiones de comunicación para TCP mediante el nombre de dominio plenamente cualificado con comunicación segura.

TCON_Param

Tabla 11-11 Estructura de la descripción de la conexión (TCON_Param)

Byte	Parámetro y tipo de datos		Descripción
de 0 a 1	block_length	UInt	Longitud: 64 bytes (fijos)
de 2 a 3	id	CONN_OUC (Word)	Referencia a esta conexión: Rango de valores: De 1 (predeterminado) a 4095. Especifique el valor de este parámetro para la instrucción TSEND_C, TRCV_C o TCON en ID.
4	connection_type	USInt	Tipo de conexión: • 17: TCP (predeterminado) • 18: ISO on TCP • 19: UDP
5	active_est	Bool	ID del tipo de conexión: • TCP e ISO on TCP: – FALSE: conexión pasiva – TRUE: conexión activa (predeterminado) • UDP: FALSE
6	local_device_id	USInt	ID de la interfaz Industrial Ethernet o PROFINET local: 1 (predeterminado)
7	local_tsap_id_len	USInt	Longitud del parámetro local_tsap_id utilizado, en bytes; valores posibles: • TCP: 0 (activa, predeterminado) o 2 (pasiva) • ISO on TCP: de 2 a 16 • UDP: 2
8	rem_subnet_id_len	USInt	Este parámetro no se utiliza.
9	rem_staddr_len	USInt	Longitud de la dirección del punto final del interlocutor, en bytes: • 0: sin especificar (el parámetro rem_staddr es irrelevante) • 4 (predeterminado): dirección IP válida en el parámetro rem_staddr (solo para TCP e ISO on TCP)
10	rem_tsap_id_len	USInt	Longitud del parámetro rem_tsap_id utilizado, en bytes; valores posibles: • TCP: 0 (pasiva) o 2 (activa, predeterminado) • ISO on TCP: de 2 a 16 • UDP: 0
11	next_staddr_len	USInt	Este parámetro no se utiliza.

Byte	Parámetro y tipo de datos		Descripción
de 12 a 27	local_tsap_id	Array [1..16] of Byte	Componente de conexión para la dirección local: • TCP e ISO on TCP: n.º de puerto local (valores posibles: de 1 a 49151; valores recomendados: 2000...5000): – local_tsap_id[1] = byte High del número de puerto en notación hexadecimal; – local_tsap_id[2] = byte Low del número de puerto en notación hexadecimal; – local_tsap_id[3-16] = irrelevante • ISO on TCP: ID de TSAP local: – local_tsap_id[1] = B#16#E0; – local_tsap_id[2] = rack y slot de los puntos finales locales (bits 0 a 4: número de slot, bits 5 a 7: número de rack); – local_tsap_id[3-16] = extensión de TSAP, opcional • UDP: este parámetro no se utiliza. Nota: asegúrese de que todos los valores de local_tsap_id son unívocos dentro de la CPU.
de 28 a 33	rem_subnet_id	Array [1..6] of USInt	Este parámetro no se utiliza.
de 34 a 39	rem_staddr	Array [1..6] of USInt	Solo TCP e ISO on TCP: dirección IP del punto final del interlocutor. (No relevante para las conexiones pasivas.) Por ejemplo, la dirección IP 192.168.002.003 se guarda en los elementos siguientes de la matriz: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6]= irrelevante
de 40 a 55	rem_tsap_id	Array [1..16] of Byte	Componente de conexión para la dirección del interlocutor: • TCP: número de puerto del interlocutor. Rango: de 1 a 49151; valores recomendados: de 2000 a 5000): – rem_tsap_id[1] = byte High del número de puerto en notación hexadecimal – rem_tsap_id[2] = byte Low del número de puerto en notación hexadecimal; – rem_tsap_id[3-16] = irrelevante • ISO on TCP: ID de TSAP del interlocutor: – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = rack y slot del punto final del interlocutor (bits 0 a 4: número de slot, bits 5 a 7: número de rack) – rem_tsap_id[3-16] = extensión de TSAP, opcional • UDP: este parámetro no se utiliza.
de 56 a 61	next_staddr	Array [1..6] of Byte	Este parámetro no se utiliza.
de 62 a 63	spare	Word	Reservado: W#16#0000

TCON_IP_V4

Tabla 11-12 Estructura de la descripción de la conexión (TCON_IP_V4): para usar con TCP

Byte	Parámetro y tipo de datos		Descripción
de 0 a 1	InterfaceId	HW_ANY	Identificador HW del submódulo interfaz IE
de 2 a 3	ID	CONN_OUC (Word)	Referencia a esta conexión: rango de valores: De 1 (predeterminado) a 4095. Especifique el valor de este parámetro para la instrucción TSEND_C, TRCV_C o TCON en ID.
4	ConnectionType	Byte	Tipo de conexión: • 11: TCP/IP (predeterminado) • 17: TCP/IP (este tipo de conexión está incluido por razones legales. Se recomienda utilizar "11: TCP/IP (predeterminado)".) • 19: UDP
5	ActiveEstablished	Bool	Establecimiento de conexión activa/pasiva: • TRUE: conexión activa (predeterminado) • FALSE: conexión pasiva
	Dirección IP V4		
6	ADDR[1]	Byte	Octeto 1
7	ADDR[1]	Byte	Octeto 2
8	ADDR[1]	Byte	Octeto 3
9	ADDR[1]	Byte	Octeto 4
de 10 a 11	RemotePort	UInt	Número de puerto UDP/TCP remoto
de 12 a 13	LocalPort	UInt	Número de puerto UDP/TCP local

TCON_IP_V4_SEC

Tabla 11-13 Estructura de la descripción de la conexión (TCON_IP_V4_SEC): para usar con TCP

Byte	Parámetro y tipo de datos		Descripción
de 0 a 15	ConnPara	TCON_IP_v4	SDT para los parámetros de la conexión Información sobre el ID de interfaz: • Si se deja el ID de interfaz en el valor predeterminado 0, el sistema operativo de la CPU evalúa la dirección IP remota y las rutas IP que existen localmente para, seguidamente, especificar una interfaz Industrial Ethernet de la CPU y establecer la conexión OUC segura. En este caso, los datos de diagnóstico se asignan siempre a la primera interfaz Industrial Ethernet de la CPU. • Si se especifica un identificador de hardware de una interfaz Industrial Ethernet de la CPU o de un CP como ID de interfaz, la conexión OUC segura se establece utilizando la interfaz Industrial Ethernet asociada.
16	ActivateSecure-Conn	Bool	Activación de la comunicación segura para esta conexión Si este parámetro tiene el valor FALSE (predeterminado), los parámetros de seguridad posteriores son irrelevantes, porque la conexión no es segura. En este caso es posible establecer una conexión TCP o UDP no segura.
17	TLSServerReq-Client-Cert	Bool	Solo para el lado del servidor: solicitud de un certificado X.509-V3 del cliente TLS. FALSE (predeterminado)
de 18 a 19	ExtTLSCapabilities	Word	• Bit 0: solo para el lado del cliente. Un bit activado significa que el cliente valida el nombre alternativo del sujeto del certificado (subjectAlternateName) en el certificado X.509-V3 del servidor para comprobar la identidad del servidor. Los certificados se comprueban cuando se establece la conexión. 16#0 (predeterminado) • Bits 1 a 15: reservados para actualizaciones futuras
de 20 a 23	TLSServerCertRef	UDInt	• Lado del servidor: ID de su propio certificado X.509-V3 • Lado del cliente: ID del certificado X.509-V3 (normalmente un certificado CA) que utiliza el cliente TLS para validar la autenticación del servidor TLS. Si este parámetro es 0, el cliente TLS utiliza todos los certificados (CA) cargados actualmente en la memoria de certificados del cliente para validar la autenticación del servidor. 0 (predeterminado)
de 24 a 27	TLSClientCertRef	UDInt	• Lado del cliente: ID de su propio certificado X.509-V3 • Lado del servidor: ID del certificado X.509-V3 (o un grupo de certificados X.509-V3) que utiliza el servidor TLS para validar el cliente TLS. Si este parámetro es 0, el servidor TLS utiliza todos los certificados (CA) cargados actualmente en la memoria de certificados del servidor para validar la autenticación del cliente. 0 (predeterminado)

El parámetro de conexión CONNECT de los DB de instancia para las instrucciones TCON, TSEND_C y TRCV_C contiene una referencia al bloque de datos utilizado.

Nota

Es posible establecer conexiones TCP o UDP no seguras vía IPv4.

También puede utilizarse el SDT TCON_IP_V4_SEC para una conexión TCP o UDP no segura vía IPv4.

TCON_IP_RFC

Tabla 11-14 Estructura de la descripción de la conexión (TCON_IP_RFC): para usar con ISO on TCP

Byte	Parámetro y tipo de datos		Descripción
de 0 a 1	InterfaceId	HW_ANY	Identificador HW del submódulo interfaz IE
de 2 a 3	ID	CONN_OUC (Word)	Referencia a esta conexión: Rango de valores: De 1 (predeterminado) a 4095. Especifique el valor de este parámetro para la instrucción TSEND_C, TRCV_C o TCON en ID.
4	ConnectionType	Byte	Tipo de conexión: • 12: ISO on TCP (predeterminado) • 17: ISO on TCP (este tipo de conexión está incluido por razones legales. Se recomienda utilizar "12: ISO on TCP (predeterminado)".)
5	ActiveEstablished	Bool	Establecimiento de conexión activa/pasiva: • TRUE: conexión activa (predeterminado) • FALSE: conexión pasiva
de 6 a 7	Repuesto		No utilizado
	Dirección IP V4		
8	ADDR[1]	Byte	Octeto 1
9	ADDR[1]	Byte	Octeto 2
10	ADDR[1]	Byte	Octeto 3
11	ADDR[1]	Byte	Octeto 4
	Selector de transporte remoto		
de 12 a 13	TSelLength	UInt	Longitud de TSelector
de 14 a 45	TSel	array [1..32] of Byte	Matriz de caracteres para nombre TSAP
	Selector de transporte local		
de 46 a 47	TSelLength	UInt	Longitud de TSelector
de 48 a 79	TSel	array [1..32] of Byte	Matriz de caracteres para nombre TSAP

TCON_QDN

Tabla 11-15 Estructura de la descripción de la conexión conforme a TCON_QDN

Byte	Parámetro y tipo de datos		Descripción
de 0 a 1	InterfaceId	HW_ANY	CPU S7-1200 con firmware V4.4 o superior: • ID de interfaz de un CP enchufado: – con CP de S7-1200 a partir del firmware V3.2 – con CP de ET 200SP a partir del firmware V2 CPU S7-1200 anterior a V4.4: • parámetro irrelevante
de 2 a 3	ID	CONN_O UC	Referencia a esta conexión (rango de valores: de 1 a 4095). Especifique el valor de este parámetro para la instrucción TCON en ID.
4	ConnectionType	BYTE	Tipo de conexión: • 11: TCP (11 dec = 0x0B hex) • 19: UDP (19 dec = 0x13 hex)
5	ActiveEstablished	BOOL	Identificador del tipo de establecimiento de conexión: • FALSE: establecimiento pasivo de la conexión • TRUE: Establecimiento activo de la conexión (iniciativa local)
de 6 a 261	RemoteQDN	ARRAY of STRING [1..254]	Nombre de dominio plenamente cualificado del punto final del interlocutor, que debe terminar con "." Recuerde que en una red SIMATIC, el nombre y la finalización no deben exceder los 254 caracteres.
de 262 a 263	RemotePort	UINT	Dirección de puerto del interlocutor remoto
de 264 a 265	LocalPort	UINT	Dirección de puerto del interlocutor local

TCON_QDN_SEC

Tabla 11-16 Estructura de la descripción de la conexión conforme a TCON_QDN_SEC

Byte	Parámetro y tipo de datos		Descripción
de 0 a 271	ConnPara	TCON_QDN	Parámetro de conexión
272	ActivateSecureConn	BOOL	Activación de la comunicación segura para esta conexión. Si este parámetro tiene el valor FALSE, los parámetros de seguridad posteriores son irrelevantes. En este caso es posible establecer una conexión TCP o UDP no segura.
273	TLSServerReqClientCert	BOOL	Solo para el lado del servidor: Solicitud de un certificado X.509-V3 del cliente TLS

Byte	Parámetro y tipo de datos		Descripción
de 274 a 275	ExtTLSCapabilities	WORD	• Bit 0: solo para el lado del cliente. Un bit activado significa que el cliente valida el subjectAlternateName en el certificado X.509-V3 del servidor para comprobar la identidad del servidor. Los certificados se comprueban cuando se establece la conexión. • Bits 1 a 15: reservados para actualizaciones futuras
de 276 a 279	TLSServerCertRef	UDINT	• Lado del servidor: ID de su propio certificado X.509-V3 • Lado del cliente: ID del certificado X.509-V3 (normalmente un certificado CA) que utiliza el cliente TLS para validar la autenticación del servidor TLS. Si este parámetro es 0, el cliente TLS utiliza todos los certificados (CA) cargados actualmente en la memoria de certificados del cliente para validar la autenticación del servidor.
de 280 a 283	TLSClientCertRef	UDINT	• Lado del cliente: ID de su propio certificado X.509-V3 • Lado del servidor: ID del certificado X.509-V3 (o un grupo de certificados X.509-V3) que utiliza el servidor TLS para validar la autenticación del cliente TLS. Si este parámetro es 0, el servidor TLS utiliza todos los certificados (CA) cargados actualmente en la memoria de certificados del servidor para validar la autenticación del cliente.

Soporte de la versión TLS

TLS (Transport Layer Security) es la seguridad de la capa de transporte en la capa de aplicación de la comunicación de datos. TLS aumenta la seguridad y privacidad de la comunicación entre la CPU S7-1200 y otros dispositivos. La versión de TLS depende de la versión de firmware de la CPU S7-1200 y de la versión de la CPU en la configuración de dispositivos del proyecto STEP 7. Para encontrar la versión de CPU en el proyecto STEP 7, proceda del siguiente modo:

1. Haga clic en la CPU en la configuración de dispositivos.
2. Observe la sección General de la pestaña Propiedades en la ventana de inspección.
3. La versión de firmware figura en el área de información del catálogo.

Para utilizar la última versión de TLS, configure la versión de la CPU en STEP 7 conforme a la versión de firmware real de la CPU. El interlocutor de la Open User Communication puede utilizar la última versión de TLS soportada para alcanzar el máximo nivel de seguridad.

Tabla 11-17 Soporte de la versión de TLS basada en la CPU y la versión de firmware del proyecto de STEP 7

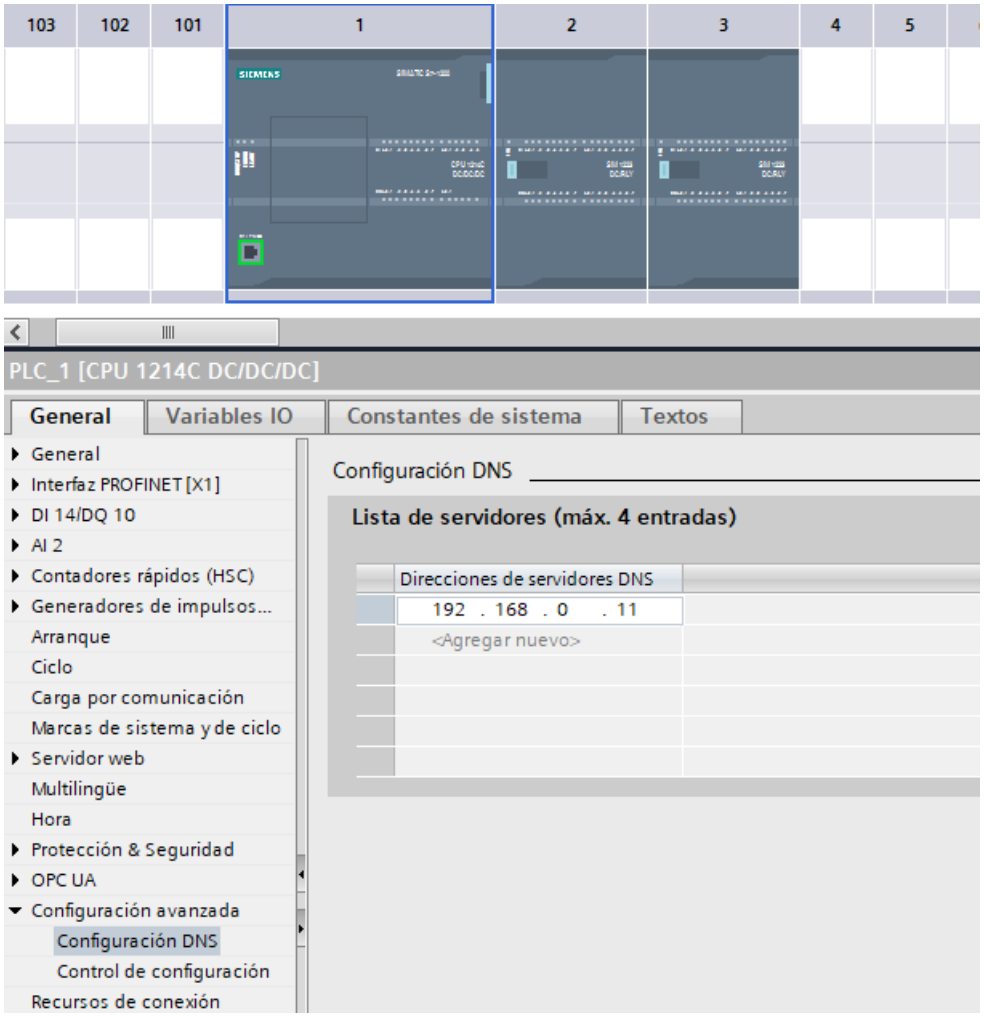
Versión de firmware configurada en el proyecto STEP 7	Versión TLS soportada
V4.5	TLS 1.2, TLS 1.3
V4.4	TLS 1.2
V4.3	TLS 1.2

11.5.8.7 Configuración de DNS

Hay que configurar un Domain Name System (DNS) para utilizar secure OUC. Debe haber por lo menos un servidor DNS en la red y hay que configurar por lo menos un servidor DNS para la CPU S7-1200.

Un servidor DNS se configura del siguiente modo:

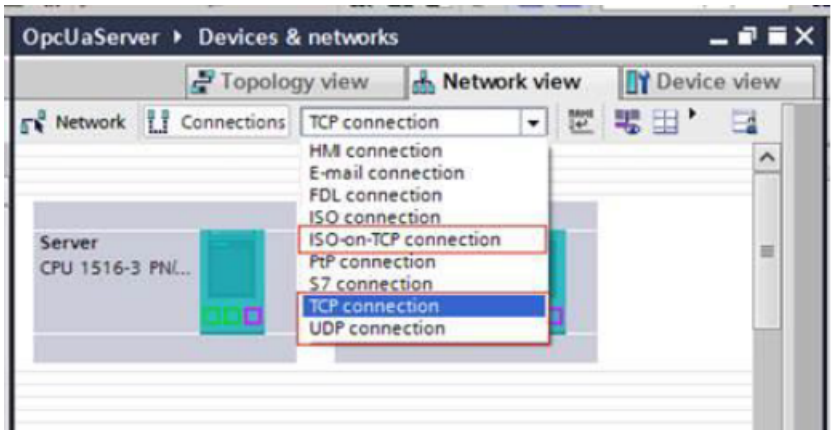
1. Navegue hasta la vista de dispositivos de la CPU S7-1200.
2. Vaya a la página de propiedades y a la ficha General.
3. Haga clic en la configuración DNS para visualizar la página de configuración.
4. En la tabla Lista de servidores, en la primera fila situada debajo de las direcciones de servidores DNS, haga clic en <Agregar nuevo> e introduzca la dirección IP del servidor DNS.



11.5.8.8 Configurar una conexión OUC en TIA Portal V17

En TIA Portal V17 existe la posibilidad de seleccionar las siguientes conexiones de Open User Communication (como se muestra a continuación) para trazarlas hacia o desde las CPU S7-1200 o S7-1500.

- Conexión ISO on TCP
- Conexión TCP
- Conexión UDP



Al trazar una línea entre dispositivos, se configura una conexión para que se compile y descargue en el dispositivo. Esta configuración de la conexión permite al firmware del S7-1200 establecer una conexión con el interlocutor cuando la CPU cambie a RUN. No hay necesidad de ejecutar una instrucción TCON con una conexión configurada. Asimismo, no hay necesidad de ejecutar una instrucción T_DISCON para una conexión configurada.

Para trazar una línea para estas conexiones, ambas interfaces de red para la CPU (o CP) deben encontrarse en la misma subred. TIA Portal no le impide trazar una conexión a dispositivos de diferentes redes. No obstante, TIA Portal generará un error cuando efectúe una compilación o una descarga en el dispositivo.

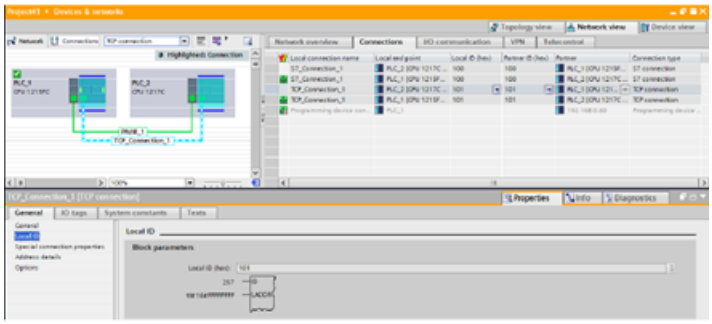
Ahora es posible trazar una conexión OUC entre las CPU S7-1200 o S7-1500, descargar la configuración y establecer automáticamente la conexión entre las CPU (si la conexión es físicamente posible).

Opciones de configuración para conexiones OUC configuradas

Configure las siguientes propiedades de la conexión como se muestra a continuación:

- ID de la conexión
- Nombre de la conexión

- Qué interlocutor inicia el establecimiento de la conexión
- Detalles del puerto en el menú Propiedades de la vista de redes seleccionando un punto final de la conexión

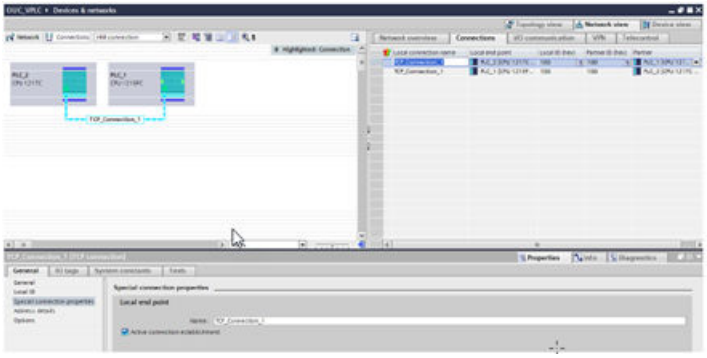


Cuando se traza una conexión OUC en TIA Portal, un "ID local" y un "ID del interlocutor" se asignan en el rango válido a un ID de conexión OUC. Puede modificar el valor asignado en la tabla de conexiones o en el ID local. El valor introducido en cada ID debe estar dentro del rango definido por las instrucciones TBlock (ver TSEND).

Nota

Rango asignado para el ID de la conexión

El rango asignado para el ID de la conexión está en el mismo rango asignado para una conexión S7. Si un ID de conexión S7 se suministra a un TSEND, esto generaría un error desde la instrucción. El error sería 16#80A1 dado que no se establecería ninguna conexión OUC.



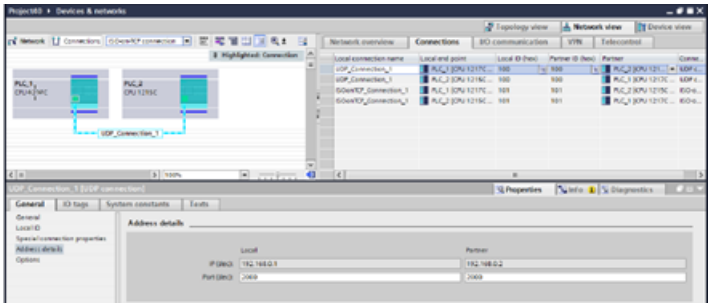
En el menú "Propiedades especiales de la conexión", el campo "Establecimiento activo de la conexión" define el dispositivo que emite los mensajes de conexión tras descargar la configuración en las CPU. Si la casilla de verificación "Establecimiento activo de la conexión" está desactivada, el dispositivo descargado espera mensajes de conexión de su interlocutor. TIA Portal actualiza automáticamente el interlocutor si la casilla de verificación "Establecimiento activo de la conexión" está activada. Solo un lado de la conexión puede ajustarse como activo.

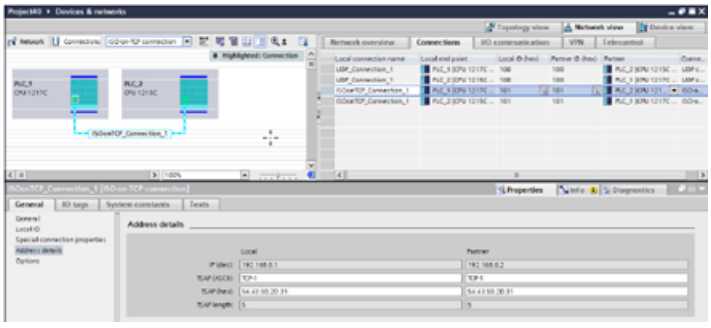
Nota

Conexiones UDP

Para las conexiones UDP, el "Establecimiento activo de la conexión" no está disponible.

El menú de propiedades "Detalles de direcciones" muestra la configuración de las direcciones que se utilizarán para la comunicación de la conexión de los tipos TCP e ISO on TCP. Al usar TUSEND y TURCV en una conexión UDP, la dirección para comunicación se reemplaza por un parámetro en las instrucciones. Además, para los tipos de conexión ISO on TCP es posible modificar los TSAP en el menú de propiedades "Detalles de direcciones" como se muestra a continuación en la segunda pantalla.





Funcionamiento de las instrucciones TBlock existentes con conexiones configuradas

Una vez establecida una conexión, una conexión configurada funciona del mismo modo que una conexión programada. Otras conexiones OUC funcionan del mismo modo que si fuesen ejecutadas en una conexión programada. A continuación figura una descripción de las instrucciones OUC que funcionan con conexiones configuradas.

Instrucción OUC	Descripción con conexión configurada
TSEND_C	Debe tener un SDT TCON_Configured transferido al parámetro Connect. Si se usa una conexión configurada, el campo de ID del TCON_Configured debe contener el ID de la conexión configurada y el campo de tipo de conexión debe estar ajustado en 254. Al usar una conexión configurada, el mecanismo de conexión de la instrucción debe omitirse dado que la conexión ya está establecida.
TRCV_C	Debe tener un SDT TCON_Configured transferido al parámetro Connect. Si se usa una conexión configurada, el campo de ID del TCON_Configured debe contener el ID de la conexión configurada y el campo de tipo de conexión debe estar ajustado en 254. Al usar una conexión configurada, el mecanismo de conexión de la instrucción debe omitirse dado que la conexión ya está establecida.
TMAIL_C	No funciona con una conexión configurada.
TCON	Devuelve un error (0x8085) si se proporciona el identificador de una conexión configurada porque la conexión ya está abierta
TDISCON	Devuelve un error (0x80A3).
TCONSettings[1]	Devuelve un error (0x8085)
TSEND	Funciona del mismo modo.

Instrucción OUC	Descripción con conexión configurada
TRCV	Funciona del mismo modo.
TUSEND	Funciona del mismo modo.
TURCV	Funciona del mismo modo.
T_RESET	Funciona del mismo modo. Desconecta y reconecta la conexión. Los valores de SentBytes y ReceivedBytes se restablecen a 0.
T_DIAG	Funciona del mismo modo. El campo "Tipo" de la variante TDIAG_Status debe indicar una conexión configurada.
T_CONFIG	Funciona del mismo modo. (No relacionado directamente a la comunicación de protocolo OUC).
MB_CLIENT	Debe tener un SDT TCON_Configured transferido al parámetro Connect. Si se usa una conexión configurada, el campo de ID del TCON_Configured debe contener el ID de la conexión configurada y el campo de tipo de conexión debe estar ajustado en 254. Al usar una conexión configurada, el mecanismo de conexión de la instrucción debe omitirse dado que la conexión ya está establecida.
MB_SERVER	Debe tener un SDT TCON_Configured transferido al parámetro Connect. Si se usa una conexión configurada, el campo de ID del TCON_Configured debe contener el ID de la conexión configurada y el campo de tipo de conexión debe estar ajustado en 254. Al usar una conexión configurada, el mecanismo de conexión de la instrucción debe omitirse dado que la conexión ya está establecida.

[1] TCONSettings no puede aplicarse a conexiones configuradas en V4.5. En TCONSettings V4.5, la función de escritura no funciona con conexiones configuradas porque la opción de apagado seguro no se utiliza con conexiones configuradas. La operación de lectura siempre devuelve FALSE para la opción de apagado seguro.

11.5.8.9 Instrucciones TSEND_C y TRCV_C

A partir de la versión V4.1 o posterior de la CPU S7-1200 junto con STEP 7 V13 SP1 o posterior, la CPU amplía la capacidad de las instrucciones TSEND_C y TRCV_C para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4 y TCON_IP_RFC.

A partir de la versión V4.3 o posterior de la CPU S7-1200 junto con STEP 7 V15.1 o posterior, la CPU amplía la capacidad de las instrucciones TSEND_C y TRCV_C para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC y TCON_IP_RFC.

A partir de la versión V4.4 o posterior de la CPU S7-1200 junto con STEP 7 V16 o posterior, la CPU amplía la capacidad de las instrucciones TSEND_C y TRCV_C para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN y TCON_QDN_SEC.

Por este motivo, S7-1200 soporta dos juegos de instrucciones TSEND_C y TRCV_C:

- Instrucciones Legacy TSEND_C y TRCV_C (Página 644): Estas instrucciones TSEND_C y TRCV_C ya existían antes de la versión V4.1 del S7-1200 y solo funcionan con parámetros de conexión que tengan estructuras conforme a TCON_Param.
- Instrucciones TSEND_C y TRCV_C (Página 631): Estas instrucciones TSEND_C y TRCV_C ofrecen toda la funcionalidad de las instrucciones Legacy más la capacidad de usar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN, and TCON_QDN_SEC.

Seleccionar la versión de las instrucciones TSEND_C y TRCV_C

En STEP 7 están disponibles dos versiones de las instrucciones TSEND_C y TRCV_C:

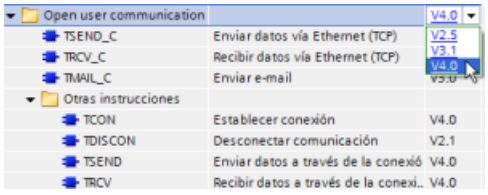
- Las versiones 2.5 y 3.1 estaban disponibles en STEP 7 Basic/Professional V13 y versiones anteriores.
- La versión 4.0 está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No utilice diferentes versiones de la instrucción en el mismo programa de la CPU.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



Para cambiar la versión de las instrucciones TSEND_C y TRCV_C, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción TSEND_C o TRCV_C en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción TSEND_C o TRCV_C seleccionada. La nueva instancia del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción TSEND_C o TRCV_C en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione un árbol del proyecto TSEND_C o TRCV_C o instancia del FB o la FC, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción TSEND_C o TRCV_C.

TSEND_C y TRCV_C (enviar y recibir datos vía Ethernet)

La instrucción TSEND_C combina las funciones de las instrucciones TCON, TDISCON y TSEND . La instrucción TRCV_C combina las funciones de las instrucciones TCON, TDISCON y TRCV. (Véase "TCON, TDISCON, TSEND y TRCV (Página 652)" para más información sobre estas instrucciones.)

El tamaño mínimo de los datos que pueden transmitirse (TSEND_C) o recibirse (TRCV_C) es de un byte; el tamaño máximo es de 8192 bytes. TSEND_C no admite la transmisión de datos desde posiciones booleanas y TRCV_C no recibe datos en posiciones booleanas. Encontrará más

información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 187).

Nota

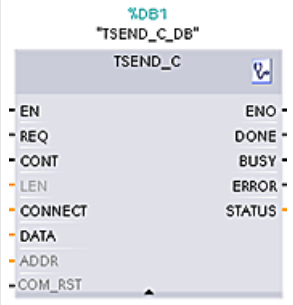
Inicializar los parámetros de comunicación

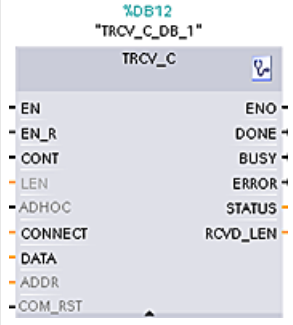
Después de introducir la instrucción TSEND_C o TRCV_C, utilice las "Propiedades" de la instrucción (Página 596) para configurar los parámetros (Página 618) de comunicación. Cuando se introducen los parámetros de los interlocutores en la ventana de inspección, STEP 7 introduce los datos correspondientes en el DB de la instrucción.

Si desea utilizar un DB multiinstancia, debe configurar manualmente el DB en ambas CPUs.

Tabla 11-18 Instrucciones TSEND_C y TRCV_C

KOP / FUP	SCL	Descripción

	`"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 establece una conexión de comunicación TCP o ISO-on-TCP con un interlocutor, envía datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

	`"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 establece una conexión de comunicación TCP o ISO-on-TCP con una CPU interlocutora, recibe datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-19 Tipos de datos TSEND_C y TRCV_C para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ (TSEND_C)	IN	Bool	Inicia la tarea de transmisión cuando se produce un flanco ascendente
EN_R (TRCV_C)	IN	Bool	Habilitación de la recepción

Parámetro y tipo		Tipo de datos	Descripción
CONT	IN	Bool	Controla la conexión de comunicación: • 0: Deshacer la conexión de comunicación después de transmitir los datos. • 1: Establecer y mantener la conexión de comunicación. Cuando se envían (TSEND_C) (flanco ascendente en el parámetro REQ) o se reciben datos (TRCV_C) (flanco ascendente en el parámetro EN_R), el parámetro CONT debe tener el valor TRUE con el fin de establecer o mantener una conexión.
LEN	IN	UDInt	Parámetro opcional (oculto) Número máximo de bytes que deben transmitirse (TSEND_C) o recibirse (TRCV_C) con la tarea. Si usa valores puramente simbólicos en el parámetro DATA, el parámetro LEN debe tener el valor "0".
ADHOC (TRCV_C)	IN	Bool	Parámetro opcional (oculto) solicitud ad hoc para el tipo de conexión TCP.
CONNECT	IN_OUT	Variant	Puntero a la descripción de la conexión: • En el caso de TCP o UDP, use la estructura TCON_IP_v4 o TCON_QDN. Para obtener una descripción, véase: "Parámetros de la conexión PROFINET (Página 618)". • Para TCP con comunicación segura, use la estructura TCON_IP_V4_SEC o TCON_QDN_SEC. Para obtener una descripción, véase: "Parámetros de la conexión PROFINET (Página 618)". • Para ISO-on-TCP, use la estructura TCON_IP_RFC. Para obtener una descripción, véase: "Parámetros de la conexión PROFINET (Página 618)". • Para conexiones ISO del CP 1543-1 / CP 1545-1, use la estructura TCON_ISOnative. Para obtener una descripción, consulte la Ayuda en pantalla de TIA Portal: "Estructura de la descripción de la conexión conforme a TCON_ISOnative". • Para conexiones con clientes SMS, use el tipo de datos de sistema TCON_PHONE. Para obtener una descripción, consulte la Ayuda en pantalla de TIA Portal: "Parámetros de conexión con TCON_Phone". • Para conexiones FDL del CM 1542-5, use el tipo de datos de sistema TCON_FDL; consulte la Ayuda en pantalla de TIA Portal: "Parámetros de conexión con TCON_FDL".
DATA	IN_OUT	Variant	Puntero al área de transmisión que incluye lo siguiente: • Dirección y longitud de los datos que se van a transmitir (TSEND_C) • Dirección y longitud máxima de los datos recibidos (TRCV_C)
ADDR	IN_OUT	Variant	Parámetro opcional (oculto) Puntero a la dirección del destinatario con el tipo de conexión UDP. La información de dirección se mapea en la estructura TADDR_Param ###.

Parámetro y tipo		Tipo de datos	Descripción
COM_RST	IN_OUT	Bool	Parámetro opcional (oculto) Reinicia la instrucción: • 0: Irrelevante • 1: reinicia la instrucción por completo; la conexión existente se deshace o se inicializa y se establece de nuevo según CONT. El parámetro COM_RST se inicializa después de la evaluación mediante la instrucción TSEND_C o TRCV_C y, por tanto, no se debe cambiar estáticamente.
DONE	OUT	Bool	Parámetro de estado con los valores siguientes: • 0: tarea de transmisión no iniciada o aún en ejecución. • 1: tarea de transmisión ejecutada sin errores. Este estado solo se muestra para un ciclo.
BUSY	OUT	Bool	Parámetro de estado con los valores siguientes: • 0: tarea de transmisión no iniciada aún o ya finalizada. • 1: tarea de transmisión no finalizada aún. No se puede iniciar una tarea de transmisión nueva.
ERROR	OUT	Bool	Parámetros de estado con los valores siguientes: • 0: No hay error • 1: Se ha producido un error durante el establecimiento de la conexión, la transmisión de los datos o la interrupción de la conexión.
STATUS	OUT	Word	Estado de instrucción (véase la descripción de los parámetros ERROR y STATUS).
RCVD_LEN (TRCV_C)	OUT	Int	Cantidad de datos (en bytes) recibida realmente.

Nota

La instrucción TSEND_C requiere una transición de bajo a alto en el parámetro de entrada REQ para iniciar una tarea de transmisión. El parámetro BUSY se pone a 1 durante el procesamiento. La finalización de la tarea de transmisión se indica poniendo a 1 los parámetros DONE o ERROR durante un ciclo. Durante este tiempo, se ignora cualquier transición de bajo a alto en el parámetro de entrada REQ.

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Se recomienda que los datos transmitidos por la instrucción TSEND_C tengan el mismo tamaño que el parámetro DATA de la instrucción TRCV_C.

Si se utiliza el ajuste predeterminado del parámetro LEN y hay que enviar los datos en segmentos más pequeños que el tamaño del parámetro DATA, se aplica lo siguiente. Si el tamaño de los datos transmitidos desde TSEND_C no es el mismo que el tamaño del parámetro TRCV_C DATA, TRCV_C permanece en estado ocupado (código de estado: 7006) hasta que el tamaño total de los datos transmitidos desde TSEND_C sea el mismo que el tamaño del parámetro TRCV_C DATA.

El búfer del parámetro TRCV_C DATA no muestra los nuevos datos recibidos hasta que el tamaño de los datos sea el mismo que el tamaño del búfer del parámetro DATA.

Operaciones TSEND_C

La instrucción TSEND_C se ejecuta de forma asíncrona e implementa las funciones siguientes en secuencia:

1. Configuración y establecimiento de una conexión de comunicación:
   TSEND_C configura una conexión de comunicación y establece dicha conexión si se detecta un flanco ascendente en el parámetro REQ y aún no existe una conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente. La descripción de la conexión especificada en el parámetro CONNECT se usa para configurar la conexión de comunicación. Se pueden usar los tipos de conexión siguientes:
   - Estructura TCON_Param para los protocolos TCP, ISO on TCP y UDP
   - Con V4.1 o posterior, TCP/UDP: descripción de la conexión mediante la estructura TCON_IP_V4 en el parámetro CONNECT.
   - Con V4.1 o posterior, ISO-on-TCP: descripción de la conexión mediante la estructura TCON_IP_RFC en el parámetro CONNECT.
   - Con V4.3 o posterior, TCP: descripción de la conexión mediante la estructura TCON_IP_V4_SEC en el parámetro CONNECT.
   - Con V4.4 o posterior, TCP: Descripción de la conexión mediante las estructuras TCON_QDN y TCON_QDN_SEC en el parámetro CONNECT.

   Una conexión existente se deshace y la conexión que se ha configurado se elimina cuando la CPU pasa al estado operativo STOP. Para configurar y establecer la conexión de nuevo, debe volver a ejecutar TSEND_C. Para obtener información sobre el número de conexiones de comunicación posibles, consulte los datos técnicos de la CPU.
2. Transmisión de datos mediante una conexión de comunicación existente:
   La transmisión de datos se ejecuta cuando se detecta un flanco ascendente en el parámetro REQ. Tal como se ha descrito anteriormente, la conexión de comunicación se establece en primer lugar. El área de transmisión se especifica con el parámetro DATA. Incluye la dirección y la longitud de los datos que se van a transmitir. No utilice un área de datos con el tipo de datos BOOL o Array of BOOL en el parámetro DATA. Con el parámetro LEN se especifica el número máximo de bytes transmitidos con una tarea de transmisión. Si se usa un nombre simbólico en el parámetro DATA, el parámetro LEN debe tener el valor "0".
   Los datos que se transmitirán no se deben editar hasta que finalice la tarea de transmisión.
3. Interrupción de la conexión de comunicación:
   La conexión de comunicación se deshace una vez que se han transmitido los datos si el parámetro CONT tenía el valor "0" cuando se produjo el flanco ascendente en el parámetro REQ. De lo contrario, se mantendrá la conexión de comunicación.

Si la tarea de transmisión se ejecuta correctamente, el parámetro DONE se pondrá a "1". La conexión de comunicación se deshará antes que de esto se produzca (consulte la descripción de la dependencia anterior en el parámetro CONT). El estado lógico "1" en el parámetro DONE no es una confirmación de que el interlocutor de la comunicación ya ha leído los datos.

TSEND_C se inicializa cuando el parámetro COM_RST se pone a "1". Se pueden perder datos si se transfieren datos en este momento.

Se pueden producir las situaciones siguientes en función del parámetro CONT:

- CONT = "0":
  Se establece una conexión de comunicación existente.
- CONT = "1" y se ha establecido la conexión de comunicación:
  Se inicializa y se vuelve a establecer una conexión de comunicación existente.
- CONT = "1" y no se ha establecido la conexión de comunicación.
  No se establece ninguna conexión de comunicación.

El parámetro COM_RST se ha inicializado después de que la instrucción T_SEND lo haya evaluado. Para habilitar TSEND_C otra vez después de la ejecución (DONE = 1), llame la instrucción una vez con REQ = 0.

Operaciones TRCV_C

La instrucción TRCV_C se ejecuta de forma asíncrona e implementa las funciones siguientes en secuencia:

1. Configuración y establecimiento de una conexión de comunicación:
   TRCV_C configura una conexión de comunicación y establece dicha conexión si el parámetro EN_R = "1" y no hay conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.
   La descripción de la conexión especificada en el parámetro CONNECT se usa para configurar la conexión de comunicación. Se pueden usar los tipos de conexión siguientes:
   - Estructura TCON_Param para los protocolos TCP, ISO on TCP y UDP
   - Con V4.1 y posterior, TCP/UDP: descripción de la conexión mediante la estructura TCON_IP_V4 en el parámetro CONNECT.
   - Con V4.1 y posterior, ISO on TCP: descripción de la conexión mediante la estructura TCON_IP_RFC en el parámetro CONNECT.
   - Con V4.3 y posterior, TCP: descripción de la conexión mediante la estructura TCON_IP_V4_SEC en el parámetro CONNECT.
   - Con V4.4 o posterior, TCP: Descripción de la conexión mediante las estructuras TCON_QDN y TCON_QDN_SEC.

   Una conexión existente se deshace y la conexión que se ha configurado se elimina cuando la CPU pasa al estado operativo STOP. Para configurar y establecer la conexión de nuevo, debe volver a ejecutar TRCV_C con EN_R = "1".
   Si EN_R se pone a "0" antes de establecer la conexión de comunicación, se establecerá la conexión y se mantendrá activa incluso si CONT = "0". Sin embargo, no se recibirán datos (DONE seguirá siendo "0").
   Para obtener información sobre el número de conexiones de comunicación posibles, consulte los datos técnicos de la CPU.
2. Recepción de datos mediante una conexión de comunicación existente:
   La recepción de datos se habilita cuando el parámetro EN_R se pone al valor "1". Tal como se ha descrito anteriormente, la conexión de comunicación se establece en primer lugar. Los datos recibidos se introducen en un área de recepción. La longitud del área se especifica con el parámetro LEN (si LEN <> 0) o con la información de longitud del parámetro DATA (si LEN = 0), en función de la variante de protocolo que se use. Si usa valores puramente simbólicos en el parámetro DATA, el parámetro LEN debe tener el valor "0".
   Si EN_R se pone a "0" antes de que se reciban datos por primera vez, la conexión de comunicación se mantendrá activa incluso si CONT = 0. Sin embargo, no se recibirán datos (DONE seguirá siendo "0").
3. Interrupción de la conexión de comunicación:
   La conexión de comunicación se deshará una vez que se reciban los datos si el parámetro CONT tenía el valor "0" cuando se inició la conexión establecida. De lo contrario, se mantendrá la conexión de comunicación.

Si la tarea de recepción se ejecuta correctamente, el parámetro DONE se pondrá a "1". La conexión de comunicación se deshará antes que de esto se produzca (consulte la descripción de la dependencia anterior en el parámetro CONT).

TRCV_C se inicializa cuando se ha establecido el parámetro COM_RST. Si se siguen recibiendo datos cuando se vuelve a ejecutar, se podrían perder datos. Se pueden producir las situaciones siguientes en función del parámetro CONT:

- CONT = "0":
  Se establece una conexión de comunicación existente.
- CONT = "1" y se ha establecido la conexión de comunicación:
  Se inicializa y se vuelve a establecer una conexión de comunicación existente.
- CONT = "1" y no se ha establecido la conexión de comunicación:
  No se establece ninguna conexión de comunicación.

El parámetro COM_RST se ha inicializado después de que la instrucción TRCV_ lo haya evaluado.

TRCV_C utiliza los mismos modos de recepción que la instrucción TRCV. En la tabla siguiente se muestra cómo se introducen los datos en el área de recepción:

Variante de protocolo	Disponibilidad de datos en el área de recepción	Parámetro Connection_type de la descripción de la conexión	Parámetro LEN	Parámetro RCVD_LEN
TCP (modo ad hoc)	Los datos están inmediatamente disponibles.	B#16#11	Se seleccionan con la entrada ADHOC de la instrucción TRCV_C.	De 1 a 1472
TCP (recepción de datos con la longitud especificada)	Los datos estarán disponibles en cuanto se haya recibido totalmente la longitud de datos especificada en el parámetro LEN.	B#16#11	De 1 a 8192	Idéntico al valor del parámetro LEN
ISO on TCP (transferencia de datos controlada por protocolo)	Los datos estarán disponibles en cuanto se haya recibido totalmente la longitud de datos especificada en el parámetro LEN.	B#16#12	De 1 a 8192	Idéntico al valor del parámetro LEN

Nota

Modo ad hoc

El "modo ad hoc" solo está disponible con la variante de protocolo TCP. Para configurar la instrucción TRCV_C para el modo ad hoc, ajuste el parámetro de entrada de la instrucción ADHOC. El puntero define la longitud del área de recepción en el parámetro DATA. La longitud de datos que se ha recibido realmente se emite en el parámetro RCVD_LEN. Se pueden recibir 1460 bytes como máximo.

Nota

Importar al S7-1200 proyectos S7-300/400 de STEP 7 que contienen el modo "ad hoc"

En los proyectos S7-300/400 de STEP 7, el modo "ad hoc" se selecciona asignando el valor "0" al parámetro LEN. En S7-1200, la instrucción TRCV_C se configura para el modo ad hoc ajustando el parámetro de entrada de la instrucción ADHOC.

Si se importa al S7-1200 un proyecto S7-300/400 de STEP 7 que contiene el modo "ad hoc", el parámetro LEN debe modificarse a "65535".

Nota

TCP (recepción de datos con la longitud especificada)

El valor del parámetro LEN se utiliza para especificar la longitud de la recepción de datos. Los datos especificados en el parámetro DATA están disponibles en el área de recepción en cuanto se recibe totalmente la longitud especificada en el parámetro LEN.

Nota

ISO on TCP (transferencia de datos controlada por protocolo)

En el caso de la variante de protocolo ISO on TCP, los datos se transfieren de forma controlada por el protocolo. Los parámetros LEN y DATA definen el área de recepción.

Parámetros BUSY, DONE y ERROR

Nota

Debido al procesamiento asíncrono de TSEND_C, es preciso conservar la coherencia de los datos en el área de emisión hasta que el parámetro DONE o ERROR adopte el valor TRUE.

Para TSEND_C, un estado TRUE en el parámetro DONE significa que los datos se han enviado correctamente. Sin embargo, no significa que la CPU interlocutora haya leído realmente el búfer de recepción.

Debido al procesamiento asíncrono de TRCV_C, los datos en el área de recepción solo son coherentes si el parámetro DONE = 1.

Tabla 11-20 Instrucciones TSEND_C y TRCV_C parámetros BUSY, DONE y ERROR

BUSY	DONE	ERROR	Descripción
1	0	0	La tarea de transmisión se está procesando.
0	1	0	La tarea de transmisión se ha ejecutado correctamente.
0	0	1	El establecimiento de la conexión o la tarea de transmisión se han ejecutado con un error. La causa del error se especifica en el parámetro STATUS.
0	0	0	No se ha asignado una nueva tarea de transmisión.

Puede comprobar el estado de la ejecución con los parámetros BUSY, DONE, ERROR y STATUS. El parámetro BUSY indica el estado de procesamiento. Con el parámetro DONE se comprueba si se ha ejecutado o no correctamente una tarea de transmisión. El parámetro ERROR se establece cuando se producen errores durante la ejecución de TSEND_C o TRCV_C. La información de error se genera en el parámetro STATUS.

Parámetros Error y Status

Tabla 11-21 Códigos de condición TSEND_C y TRCV_C para ERROR y STATUS

ERROR	STATUS * (W#16#...)	Descripción
0	0000	Tarea de transmisión (TSEND_C) o recepción (TRCV_C) ejecutada sin errores.
0	0001	Conexión de comunicación establecida.
0	0003	Conexión de comunicación cerrada.
0	7000	No hay ninguna ejecución de tarea de transmisión activa; no se ha establecido ninguna conexión de comunicación.
0	7001	• Inicia la ejecución de una tarea de transmisión (TSEND_C) o recepción (TRCV_C). • Establece la conexión. • Espera al interlocutor de la conexión.
0	7002	Tarea en ejecución (REQ irrelevante)
0	7003	La instrucción está finalizando la conexión de comunicación.
0	7004	Conexión de comunicación establecida y vigilada; no hay ninguna ejecución de tarea de transmisión (TSEND_C) o recepción (TRCV_C) activa.
0	7005	TSEND_C: Transferencia de datos en curso.
0	7006	TRCV_C: La instrucción está recibiendo datos.
1	8085	• El parámetro LEN es mayor que el valor máximo permitido. • La instrucción ha modificado el valor en el parámetro LEN o DATA después de la primera llamada.
1	8086	El parámetro ID dentro del parámetro CONNECT está fuera del rango permitido.
1	8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.
1	8088	El valor en el parámetro LEN no corresponde al área de recepción establecida en el parámetro DATA.
1	8089	• El parámetro CONNECT no apunta a un bloque de datos. • El parámetro CONNECT no apunta a una descripción de la conexión. • La descripción de la conexión creada manualmente tiene una estructura incorrecta para el tipo de conexión seleccionado.
1	8091	Se ha excedido la profundidad de anidamiento máxima.
1	809A	El parámetro CONNECT apunta a un campo que no corresponde a la longitud de la descripción de la conexión.
1	809B	La InterfaceId en la descripción de la conexión no corresponde a la CPU o al CP.
1	80A1	• La conexión o el puerto ya se utiliza. • Error de comunicación: – La conexión especificada no se ha establecido todavía. – La conexión especificada se está deshaciendo. No se pueden transferir datos a través de esta conexión. – La interfaz se está reinicializando.
1	80A2	El sistema está utilizando el puerto local o remoto. Encontrará más información en "Instrucciones TCON y TDISCON" (Página 652), "Códigos de condición ERROR y STATUS".

ERROR	STATUS * (W#16#...)	Descripción
1	80A3	• Se está intentando restablecer una conexión existente. • Se está intentando deshacer una conexión no existente. • La instrucción T_DIAG anidada notifica que la instrucción ha cerrado la conexión.
1	80A4	La dirección IP del punto final remoto de la conexión no es válida, lo que significa que corresponde a la dirección IP del interlocutor local.
1	80A7	Error de comunicación: ha llamado la instrucción con COM_RST = 1 antes de que se ejecutara totalmente la tarea de transmisión.
1	80AA	Otro bloque está estableciendo una conexión con la misma ID de conexión. Repita la tarea con un nuevo flanco ascendente en el parámetro REQ.
1	80B3	• Cuando se utiliza la variante de protocolo UDP, el parámetro ADDR no contiene datos. • Error en la descripción de la conexión • Otra descripción de la conexión ya utiliza el puerto local.
1	80B4	Se han incumplido una o las dos condiciones siguientes por lo que se refiere al establecimiento pasivo de la conexión (ActiveEstablished = FALSE) cuando se usa la variante de protocolo ISO-on-TCP (ConnectionType = B#16#12): • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0
1	80B5	Solo se permite el establecimiento de conexión pasiva para el tipo de conexión 13 = UDP.
1	80B6	Error de asignación de parámetro en el parámetro ConnectionType del bloque de datos para la descripción de la conexión.
1	80B7	• Para el tipo de datos de sistema TCON_Param: Error en uno de los parámetros siguientes del bloque de datos para la descripción de la conexión: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len. • Para los tipos de datos de sistema TCON_IP_V4, TCON_IP_RFC, TCON_IP_V4_SEC, TCON_QDN, y TCON_QDN_SEC : La instrucción ha establecido la dirección IP del punto final del interlocutor en 0.0.0.0.
1	80C3	• Se están utilizando todos los recursos de conexión. • Ya se está procesando un bloque con esta ID en un grupo de prioridad diferente.
1	80C4	Error de comunicación temporal: • La instrucción no puede establecer la conexión en este momento. • La instrucción no puede establecer la conexión porque los cortafuegos de la ruta de conexión no están abiertos para los puertos necesarios. • La interfaz está recibiendo nuevos parámetros o la instrucción está estableciendo la conexión. • La instrucción "TDISCON (Página 652)" está eliminando la conexión configurada. • Una llamada con COM_RST = 1 está deshaciendo la conexión empleada. • Temporalmente no hay recursos de recepción disponibles en el interlocutor de la conexión. El interlocutor de la conexión no está listo para la recepción.
1	80C5	• Conexión finalizada por el interlocutor de la comunicación. • El interlocutor de conexión remoto no ha habilitado el LSAP.
1	80C6	Error de red: • El dispositivo local no puede acceder al interlocutor remoto. • Interrupción física en PROFIBUS

ERROR	STATUS * (W#16#...)	Descripción
1	8722	Error en el parámetro CONNECT: área de origen no válida (área no declarada en el bloque de datos).
1	873A	Error en el parámetro CONNECT: no se puede obtener acceso a la descripción de la conexión (no se puede obtener acceso al bloque de datos).
1	877F	Error en el parámetro CONNECT: error interno
1	8822	TSEND_C:Parámetro DATA: área de origen no válida, el área no existe en el DB.
1	8824	TSEND_C:Parámetro DATA: error de área en el puntero VARIANT.
1	8832	TSEND_C:Parámetro DATA: el número de DB es demasiado alto.
1	883A	TSEND_C:Parámetro CONNECT: no se puede obtener acceso a los datos de conexión especificados (por ejemplo, porque el DB no existe).
1	887F	TSEND_C:Parámetro DATA: error interno (por ejemplo, referencia de VARIANT no válida).
1	893A	TSEND_C:Parámetro DATA: no se puede obtener acceso al área de transmisión (por ejemplo, porque el DB no existe).
1	8922	TRCV_C:Parámetro DATA: área de destino no válida, el área no existe en el DB.
1	8924	TRCV_C:Parámetro DATA: error de área en el puntero VARIANT.
1	8932	TRCV_C:Parámetro DATA: el número de DB es demasiado alto.
1	893A	TRCV_C:Parámetro CONNECT: no se puede obtener acceso a los datos de conexión especificados (por ejemplo, porque el DB no existe).
1	897F	TRCV_C:Parámetro DATA: error interno (por ejemplo, referencia de VARIANT no válida).
1	8A3A	TRCV_C:Parámetro DATA: no se puede obtener acceso al área de datos (por ejemplo, porque el bloque de datos no existe).
* Los códigos de error se pueden mostrar como valores enteros o hexadecimales en el editor de programas.		

Nota

Mensajes de error de las instrucciones TCON, TSEND, TRCV y TDISCON

Internamente, la instrucción TSEND_C usa las instrucciones TCON, TSEND y TDISCON; y la instrucción TRCV_C usa las instrucciones TCON, TRCV y TDISCON. Consulte "TCON, TDISCON, TSEND Y TRCV (Página 652)" para obtener más información sobre mensajes de error de estas instrucciones.

Protocolos de conexión Ethernet

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Todas las instrucciones TSEND_C y TRCV_C y TSEND y TRCV soportan los protocolos Ethernet TCP e ISO on TCP.

Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

Consulte también

ID de conexión para las instrucciones de Open User Communication (Página 616)

11.5.8.10 Instrucciones Legacy TSEND_C y TRCV_C

Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones TSEND_C y TRCV_C solo podían trabajar con parámetros de conexión con estructuras conforme a "TCON_Param". Los conceptos generales son válidos para ambos juegos de instrucciones. Consulte las instrucciones Legacy individuales TSEND_C y TRCV_C para obtener información sobre la programación.

Seleccionar la versión de las instrucciones TSEND_C y TRCV_C

En STEP 7 están disponibles dos versiones de las instrucciones TSEND_C y TRCV_C:

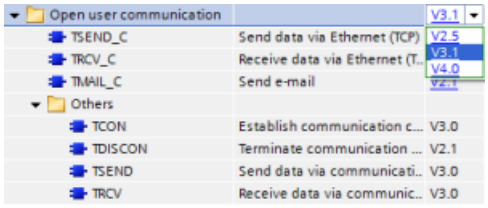
- Las versiones 2.5 y 3.1 estaban disponibles en STEP 7 Basic/Professional V13 y versiones anteriores.
- La versión 4.0 está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No utilice diferentes versiones de la instrucción en el mismo programa de la CPU.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



Para cambiar la versión de las instrucciones TSEND_C y TRCV_C, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción TSEND_C o TRCV_C en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción TSEND_C o TRCV_C seleccionada. La nueva instancia del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción TSEND_C o TRCV_C en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione un árbol del proyecto TSEND_C o TRCV_C o instancia del FB o la FC, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción TSEND_C o TRCV_C.

TSEND_C y TRCV_C antiguas (enviar y recibir datos vía Ethernet)

La instrucción Legacy TSEND_C combina las funciones de las instrucciones Legacy TCON, TDISCON y TSEND . La instrucción TRCV_C combina las funciones de las instrucciones TCON, TDISCON y TRCV. (Consulte las "Instrucciones Legacy TCON, TDISCON, TSEND y TRCV (comunicación TCP) (Página 668)" para más información sobre estas instrucciones.)

El tamaño mínimo de los datos que pueden transmitirse (TSEND_C) o recibirse (TRCV_C) es de un byte; el tamaño máximo es de 8192 bytes. TSEND_C no admite la transmisión de datos desde

posiciones booleanas y TRCV_C no recibe datos en posiciones booleanas. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 187).

Nota

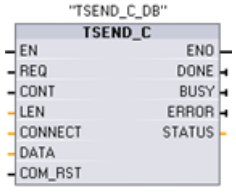
Inicializar los parámetros de comunicación

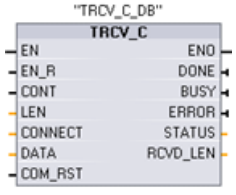
Después de introducir la instrucción TSEND_C o TRCV_C, utilice las "Propiedades" de la instrucción (Página 596) para configurar los parámetros (Página 618) de comunicación. Cuando se introducen los parámetros de los interlocutores en la ventana de inspección, STEP 7 introduce los datos correspondientes en el DB de la instrucción.

Si desea utilizar un DB multiinstancia, debe configurar manualmente el DB en ambas CPUs.

Tabla 11-22 Instrucciones TSEND_C y TRCV_C

KOP / FUP	SCL	Descripción

	`"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 establece una conexión de comunicación TCP o ISO-on-TCP con un interlocutor, envía datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

	`"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 establece una conexión de comunicación TCP o ISO-on-TCP con una CPU interlocutora, recibe datos y puede deshacer la conexión. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-23 Tipos de datos TSEND_C y TRCV_C para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ (TSEND_C)	IN	Bool	REQ = 1 inicia la tarea de transmisión TSEND_C en un flanco ascendente con la conexión descrita en el parámetro CONNECT. (CONT = 1 también se requiere para establecer y mantener la conexión de comunicación.)
EN_R (TRCV_C)	IN	Bool	Si EN_R = 1, TRCV_C está preparado para recibir. La tarea de recepción se procesa. (CONT = 1 también se requiere para establecer y mantener la conexión de comunicación.)
CONT	IN	Bool	Controla la conexión: • 0: desconectar la conexión • 1: establecer y mantener la conexión Cuando se envían datos (TSEND_C) (flanco ascendente en el parámetro REQ), el parámetro CONT debe tener el valor TRUE con el fin de establecer o mantener una conexión. Cuando se reciben datos (TRCV_C) (flanco ascendente en el parámetro EN_R), el parámetro CONT debe tener el valor TRUE con el fin de establecer o mantener una conexión.
LEN	IN	UInt	Número máximo de bytes que deben enviarse (TSEND_C) o recibirse (TRCV_C): • Predeterminado = 0: el parámetro DATA determina la longitud de los datos que deben enviarse (TSEND_C) o recibirse (TRCV_C). • Modo ad hoc = 65535: una longitud de datos variable está configurada para la recepción (TRCV_C).
CONNECT	IN_OUT	TCON_Param	Puntero hacia la descripción de la conexión (Página 618)
DATA	IN_OUT	Variant	• Contiene la dirección y longitud de los datos que se van a enviar (TSEND_C). • Contiene la dirección de inicio y la longitud máxima de los datos recibidos (TRCV_C).
COM_RST	IN_OUT	Bool	Permite reiniciar la instrucción: • 0: Irrelevante • 1: Reinicio completo del bloque de función; se deshace la conexión existente.
DONE	OUT	Bool	• 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
BUSY	OUT	Bool	• 0: Tarea finalizada. • 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	OUT	Bool	Parámetros de estado con los valores siguientes: • 0: No hay error • 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de estado, incluida la información de error. (Véase la tabla "Parámetros de error y estado" a continuación.)
RCVD_LEN (TRCV_C)	OUT	Int	Cantidad de datos (en bytes) recibida realmente

Nota

La instrucción TSEND_C requiere una transición de bajo a alto en el parámetro de entrada REQ para iniciar una tarea de transmisión. El parámetro BUSY se pone a 1 durante el procesamiento. La finalización de la tarea de transmisión se indica poniendo a 1 los parámetros DONE o ERROR durante un ciclo. Durante este tiempo, se ignora cualquier transición de bajo a alto en el parámetro de entrada REQ.

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Se recomienda que los datos transmitidos por la instrucción TSEND_C tengan el mismo tamaño que el parámetro DATA de la instrucción TRCV_C.

Si se utiliza el ajuste predeterminado del parámetro LEN y hay que enviar los datos en segmentos más pequeños que el tamaño del parámetro DATA, se aplica lo siguiente. Si el tamaño de los datos transmitidos desde TSEND_C no es el mismo que el tamaño del parámetro TRCV_C DATA, TRCV_C permanece en estado ocupado (código de estado: 7006) hasta que el tamaño total de los datos transmitidos desde TSEND_C sea el mismo que el tamaño del parámetro TRCV_C DATA.

El búfer del parámetro TRCV_C DATA no muestra los nuevos datos recibidos hasta que el tamaño de los datos sea el mismo que el tamaño del búfer del parámetro DATA.

Operaciones TSEND_C

Las funciones siguientes describen el funcionamiento de la instrucción TSEND_C:

- Para establecer una conexión, ejecute TSEND_C con CONT = 1.
- Una vez establecida correctamente la conexión, TSEND_C activa el parámetro DONE durante un ciclo.
- Para deshacer la conexión, ejecute TSEND_C con CONT = 0. La conexión se interrumpirá inmediatamente. Esto afecta también la estación receptora. La conexión se cierra allí y pueden perderse los datos del búfer de recepción.
- Para enviar datos a través de una conexión establecida, ejecute TSEND_C cuando se produzca un flanco ascendente en REQ. Tras una operación de envío correcta, TSEND_C activa el parámetro DONE durante un ciclo.
- Para establecer una conexión y enviar datos, ejecute TSEND_C con CONT = 1 y REQ = 1. Tras una operación de envío correcta, TSEND_C activa el parámetro DONE durante un ciclo.

Operaciones TRCV_C

Las funciones siguientes describen el funcionamiento de la instrucción TRCV_C:

- Para establecer una conexión, ejecute TRCV_C con el parámetro CONT = 1.
- Para recibir datos, ejecute TRCV_C con el parámetro EN_R = 1. TRCV_C recibe los datos continuamente si los parámetros EN_R = 1 y CONT = 1.
- Para deshacer la conexión, ejecute TRCV_C con el parámetro CONT = 0. La conexión se deshace inmediatamente y pueden perderse datos.

TRCV_C utiliza los mismos modos de recepción que la instrucción TRCV. En la tabla siguiente se muestra cómo se introducen los datos en el área de recepción:

Tabla 11-24 Entrada de datos en el área de recepción

Variante de protocolo	Entrada de datos en el área de recepción	Parámetro "connection_type"	Valor del parámetro LEN	Valor del parámetro RCVD_LEN (bytes)
TCP	Modo ad hoc	B#16#11	65535	1 a 1472
TCP	Recepción de datos con la longitud especificada	B#16#11	0 (recomendado) o 1 a 8192, excepto 65535	1 a 8192
ISO on TCP	Modo ad hoc	B#16#12	65535	1 a 1472
ISO on TCP	Controlado por protocolo	B#16#12	0 (recomendado) o 1 a 8192, excepto 65535	1 a 8192

Nota

Modo ad hoc

El modo "ad hoc" solo está disponible con las variantes de protocolo TCP e ISO on TCP. El modo "ad hoc" se establece asignando el valor "65535" al parámetro LEN. El área de recepción es idéntica al área especificada en el parámetro DATA. La longitud de los datos recibidos será transferida al parámetro RCVD_LEN.

Si se guardan los datos en un DB "optimizado" (solo simbólico), únicamente es posible recibir datos en matrices de tipos de datos Byte, Char, USInt y SInt.

Nota

Importar al S7-1200 proyectos S7-300/400 de STEP 7 que contienen el modo "ad hoc"

En los proyectos S7-300/400 de STEP 7, el modo "ad hoc" se selecciona asignando el valor "0" al parámetro LEN. En el S7-1200, el modo "ad hoc" se establece asignando el valor "65535" al parámetro LEN.

Si se importa al S7-1200 un proyecto S7-300/400 de STEP 7 que contiene el modo "ad hoc", el parámetro LEN debe modificarse a "65535".

Nota

Es preciso conservar la coherencia de los datos en el área de transmisión hasta que el parámetro DONE o ERROR adopten el valor TRUE

Debido al procesamiento asíncrono de TSEND_C, es preciso conservar la coherencia de los datos en el área de emisión hasta que el parámetro DONE o ERROR adopte el valor TRUE.

Para TSEND_C, un estado TRUE en el parámetro DONE significa que los datos se han enviado correctamente. Sin embargo, no significa que la CPU interlocutora haya leído realmente el búfer de recepción.

Debido al procesamiento asíncrono de TRCV_C, los datos en el área de recepción solo son coherentes si el parámetro DONE = 1.

Tabla 11-25 Instrucciones TSEND_C y TRCV_C parámetros BUSY, DONEy ERROR

BUSY	DONE	ERROR	Descripción
TRUE	Irrelevante	Irrelevante	La tarea se está procesando.
FALSE	TRUE	FALSE	La tarea se ha completado correctamente.
FALSE	FALSE	TRUE	La tarea se ha finalizado con un error. La causa del error se indica en el parámetro STATUS.
FALSE	FALSE	FALSE	No se ha asignado ninguna tarea nueva.

Parámetros Error y Status

Tabla 11-26 Códigos de condición TSEND_C y TRCV_C para ERROR y STATUS

ERROR	STATUS	Descripción
0	0000	Tarea ejecutada sin errores
0	7000	No se está procesando ninguna tarea
0	7001	Iniciar procesamiento de la tarea, estableciendo la conexión, esperando al interlocutor
0	7002	Enviando o recibiendo datos
0	7003	Deshaciendo la conexión
0	7004	Conexión establecida y vigilada, no se está procesando ninguna tarea
1	8085	El parámetro LEN es mayor que el valor máximo admisible.
1	8086	El parámetro CONNECT está fuera del rango admisible.
1	8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.
1	8088	El parámetro LEN no es válido para el área de memoria especificada en DATA.
1	8089	El parámetro CONNECT no apunta a un bloque de datos.
1	8091	Se ha excedido la profundidad de anidamiento máxima.
1	809A	El parámetro CONNECT apunta a un campo que no concuerda con la longitud de la descripción de la conexión.
1	809B	La "local_device_id" indicada en la descripción de la conexión no concuerda con la CPU.
1	80A1	Error de comunicación: • La conexión indicada no se ha establecido todavía • La conexión indicada se está deshaciendo; no es posible transferir a través de esta conexión • La interfaz se está reinicializando
1	80A3	Se está intentando deshacer una conexión no existente
1	80A4	La dirección IP del interlocutor remoto de la conexión no es válida. Por ejemplo, la dirección IP del interlocutor remoto es igual a la dirección IP del interlocutor local.
1	80A5	ID de conexión (Página 616) en uso.
1	80A7	Error de comunicación: TDISCON se ha llamado antes de finalizar TSEND_C.
1	80B2	El parámetro CONNECT apunta a un bloque de datos generado con la palabra clave UNLINKED

ERROR	STATUS	Descripción
1	80B3	Parámetros incoherentes: • Error en la descripción de la conexión • El puerto local (parámetro local_tsap_id) ya existe en una descripción de conexión diferente. • La ID indicada en la descripción de la conexión no concuerda con la ID especificada como parámetro
1	80B4	Si se utiliza ISO on TCP (connection_type = B#16#12) para establecer una conexión pasiva, el código de condición 80B4 advierte de que el TSAP introducido no cumple uno de los siguientes requisitos para la dirección: • Si la longitud del TSAP local es 2 y el valor del ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 o 01. • Si la longitud del TSAP local es 3 o superior y el valor del ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 o 01 y todos los demás bytes deberán ser caracteres ASCII válidos. • Si la longitud del TSAP local es 3 o superior y el valor del primer byte del ID TSAP no es E0 ni E1 (hexadecimal), todos los demás bytes del ID TSAP deberán ser caracteres ASCII válidos. Los caracteres ASCII válidos son valores de byte comprendidos entre 20 y 7E (hexadecimal).
1	80B7	El tipo de datos y/o la longitud de los datos transmitidos no cabe en el área de la CPU interlocutora en la que deben escribirse.
1	80C3	Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: • La conexión no se puede establecer en estos momentos • La interfaz está recibiendo nuevos parámetros • La instrucción TDISCON está deshaciendo la conexión configurada.
1	8722	Parámetro CONNECT: Área de origen no válida: el área no existe en el DB.
1	873A	Parámetro CONNECT: Imposible acceder a la descripción de la conexión (p. ej. porque el DB no está disponible).
1	877F	Parámetro CONNECT: Error interno, p. ej. referencia no válida a ANY
1	893A	El parámetro contiene el número de un DB que no está cargado.

Protocolos de conexión Ethernet

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Todas las instrucciones TSEND_C y TRCV_C y TSEND y TRCV soportan los protocolos Ethernet TCP e ISO on TCP.

Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

11.5.8.11 Instrucciones TCON, TDISCON, TSEND y TRCV

A partir de la versión V4.1 o posterior de la CPU S7-1200 junto con STEP 7 V13 SP1 o posterior, la CPU S7-1200 amplía la capacidad de la instrucción TCON para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4 y TCON_IP_RFC. La CPU S7-1200 también amplía la capacidad de las instrucciones TSEND y TRCV para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4 y TCON_IP_RFC.

A partir de la versión V4.3 o posterior de la CPU S7-1200 junto con STEP 7 V15.1 o posterior, la CPU S7-1200 amplía la capacidad de la instrucción TCON para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC y TCON_IP_RFC.

A partir de la versión V4.4 o posterior de la CPU S7-1200 junto con STEP 7 V16 o posterior, la CPU amplía la capacidad de la instrucción TCON para utilizar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN y TCON_QDN_SEC.

Por este motivo, S7-1200 soporta dos juegos de instrucciones TCON, TDISCON, TSEND y TRCV:

- Instrucciones Legacy TCON, TDISCON, TSEND y TRCV (Página 668): Estas instrucciones TCON, TDISCON, TSEND y TRCV ya existían antes de la versión V4.1 del S7-1200 y solo funcionan con parámetros de conexión que tengan estructuras conforme a TCON_Param.
- Instrucciones TCON, TDISCON, TSEND y TRCV (Página 652): Estas instrucciones TCON, TDISCON, TSEND y TRCV ofrecen toda la funcionalidad de las instrucciones Legacy más la capacidad de usar parámetros de conexión con estructuras conforme a TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN y TCON_QDN_SEC.

Selección de la versión de las instrucciones TCON, TDISCON, TSEND y TRCV

Existen dos versiones de las instrucciones TCON, TDISCON, TSEND o TRCV disponibles en STEP 7:

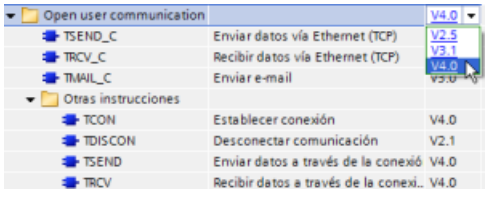
- Las versiones 2.5 y 3.1 estaban disponibles en STEP 7 Basic/Professional V13 y versiones anteriores.
- La versión 4.0 está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No utilice diferentes versiones de la instrucción en el mismo programa de la CPU.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



Para cambiar la versión de las instrucciones TCON, TDISCON, TSEND o TRCV, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción TCON, TDISCON, TSEND y TRCV en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción TCON, TDISCON, TSEND o TRCV seleccionada. La nueva instancia del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción TCON, TDISCON, TSEND o TRCV en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione un árbol del proyecto TCON, TDISCON, TSEND o TRCV o instancia del FB o la FC, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción TCON, TDISCON, TSEND o TRCV.

Instrucciones TCON, TDISCON, TSEND y TRCV (comunicación TCP)

Comunicación Ethernet con los protocolos TCP e ISO on TCP

Nota

Instrucciones TSEND_C y TRCV_C

Para ayudar a simplificar la programación de la comunicación PROFINET/Ethernet, las instrucciones TSEND_C y TRCV_C combinan las funciones de las instrucciones TCON, TDISCON, TSEND y TRCV:

- TSEND_C combina las instrucciones TCON, TDISCON y TSEND.
- TRCV_C combina las instrucciones TCON, TDISCON y TRCV.

Las instrucciones siguientes controlan el proceso de comunicación:

- TCON establece la conexión TCP/IP entre el cliente y el servidor (CPU) PC.
- TSEND y TRCV permiten enviar y recibir datos.
- TDISCON deshace la conexión.

El tamaño mínimo de los datos que pueden transmitirse (TSEND) o recibirse (TRCV) es de un byte; el tamaño máximo es de 8192 bytes. TSEND no admite la transmisión de datos desde posiciones booleanas y TRCV no recibe datos en posiciones booleanas. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 187).

TCON, TDISCON, TSEND y TRCV funcionan de forma asíncrona, con lo que el procesamiento de la tarea abarca varias ejecuciones de la instrucción. Por ejemplo, una tarea para configurar y establecer una conexión se inicia ejecutando una instrucción TCON con el parámetro REQ = 1. Después se utilizan ejecuciones adicionales de TCON para vigilar la tarea y comprobar si ha finalizado con el parámetro DONE.

La tabla siguiente muestra la relación entre los parámetros BUSY, DONE y ERROR. Utilice la tabla para determinar el estado actual de la tarea:

Tabla 11-27 Interacción entre los parámetros BUSY, DONE y ERROR

BUSY	DONE	ERROR	Descripción
1	0	0	La tarea se está procesando.
0	1	0	La tarea se ha ejecutado correctamente.
0	0	1	La tarea ha finalizado con un error. La causa del error se genera en el parámetro STATUS.
0	0	0	No se ha asignado una nueva tarea.

TCON y TDISCON

Nota

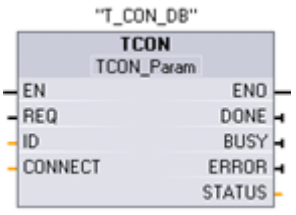
Inicializar los parámetros de comunicación

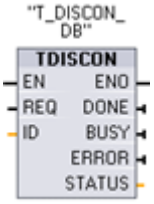
Después de introducir la instrucción TCON, utilice las "Propiedades" de la instrucción (Página 596) para configurar los parámetros de comunicación (Página 618). Cuando se introducen los parámetros de los interlocutores en la ventana de inspección, STEP 7 introduce los datos correspondientes en el DB de instancia de la instrucción.

Si desea utilizar un DB multiinstancia, debe configurar manualmente el DB en ambas CPUs.

Tabla 11-28 Instrucciones TCON y TDISCON

KOP / FUP		Descripción

	`"TCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_undef_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_);`
		TCP e ISO on TCP: TCON inicia una conexión entre la CPU y un interlocutor.

	`"TDISCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		TCP e ISO on TCP: TDISCON deshace una conexión entre la CPU y un interlocutor.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-29 Tipos de datos para los parámetros de TCON y TDISCON

Parámetro	Declaración	Tipo de datos	Descripción
REQ	IN	Bool	Inicia la tarea de establecer la conexión especificada en la ID cuando se produce un flanco ascendente.
ID	IN	CONN_OUC (Word)	Referencia a la conexión asignada. Rango de valores: W#16#0001 a W#16#0FFF

Parámetro	Declaración	Tipo de datos	Descripción
CONNECT (TCON)	IN_OUT	VARIANT	Puntero a la descripción de la conexión: • En el caso de TCP o UDP, use la estructura TCON_IP_v4 o TCON_QDN. Para obtener una descripción, véase: "Parámetros de la conexión PROFINET (Página 618)". • Para TCP con comunicación segura, use la estructura TCON_IP_V4_SEC o TCON_QDN_SEC. Para obtener una descripción, véase: "Parámetros de la conexión (Página 618) PROFINET (Página 618)". • Para ISO-on-TCP, use la estructura TCON_IP_RFC. Para obtener una descripción, véase: "Parámetros de la conexión PROFINET (Página 618)". • Para conexiones ISO del CP 1543-1 / CP 1545-1, use la estructura TCON_ISOnative. Para obtener una descripción, véase: Ayuda en pantalla de TIA Portal: "Estructura de la descripción de la conexión conforme a TCON_ISOnative". • Para conexiones con clientes SMS, use el tipo de datos de sistema TCON_PHONE. Para obtener una descripción, consulte la Ayuda en pantalla de TIA Portal: "Parámetros de conexión con TCON_Phone". • Para conexiones FDL del CM 1542-5, use el tipo de datos de sistema TCON_FDL; consulte la Ayuda en pantalla de TIA Portal: "Parámetros de conexión con TCON_FDL".
DONE	OUT	Bool	Parámetro de estado con los valores siguientes: • 0: Tarea no iniciada aún o en proceso. • 1: Tarea ejecutada sin errores.
BUSY	OUT	Bool	Parámetro de estado con los valores siguientes: • 0: Tarea no iniciada aún o ya finalizada. • 1: tarea no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	OUT	Bool	Parámetro de estado ERROR: • 0: No hay error • 1: se ha producido un error
STATUS	OUT	Word	Estado de la instrucción

Ambos interlocutores ejecutan la instrucción TCON para configurar y establecer la conexión. El punto final activo y el punto final pasivo de la comunicación se especifican mediante parámetros. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

Si la conexión se deshace p. ej. debido a una interrupción de la línea o por el interlocutor remoto, el interlocutor activo intenta restablecer la conexión configurada. No es necesario volver a ejecutar TCON.

Una conexión existente se deshace y la conexión configurada se elimina cuando se ejecuta la instrucción TDISCON o cuando la CPU pasa a estado operativo STOP. Para configurar y restablecer la conexión es preciso ejecutar TCON de nuevo.

Tabla 11-30 Códigos de condición ERROR y STATUS para TCON y TDISCON

ERROR	STATUS * (W#16#...)	Explicación
0	0000	La conexión se ha establecido correctamente.
0	7000	No se está procesando ninguna tarea
0	7001	Inicia la ejecución de la tarea, establece la conexión (TCON) o deshace la conexión (TDISCON).
0	7002	La instrucción está estableciendo una conexión (REQ es irrelevante); establece la conexión (TCON) o deshace la conexión (TDISCON).
1	8085	TCON: Se está utilizando la ID de la conexión.
1	8086	TCON: El parámetro ID está fuera del rango válido.
1	8087	TCON: Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.
1	8089	TCON: El parámetro CONNECT no apunta a una descripción de la conexión o bien esta se ha creado manualmente.
1	809A	TCON: La instrucción no soporta la estructura del parámetro CONNECT o la longitud no es válida.
1	809B	TCON: • El elemento InterfaceId en la descripción de la conexión no corresponde a la CPU o al CP, o bien es "0". • El elemento InterfaceId dentro de la estructura TCON_xxx no hace referencia a un identificador de hardware de una interfaz de CPU o de CM/CP.
1	80A1	TCON: Para TCP/UDP: la conexión o el puerto se están utilizando.
1	80A2	TCON: el sistema está utilizando el puerto local o remoto. Encontrará más información en "Parámetros comunes para instrucciones" (Página 723), "TSAP y números de puerto restringidos para comunicación ISO y TCP pasiva".
1	80A3	TCON: Una conexión (TCON) creada por el programa de usuario utiliza el valor en el parámetro ID. La conexión utiliza la misma ID y la misma configuración de conexión en el parámetro CONNECT.
1	80A4	TCON: La dirección IP del punto final remoto de la conexión no es válida o corresponde a la dirección IP del interlocutor local.
1	80A7	TCON: Error de comunicación: ha ejecutado "TDISCON" antes de que finalizara "TCON".
1	80B3	Parametrización incoherente
1	80B4	TCON: Solo con TCON_IP_RFC: se ha producido una de las opciones siguientes: • La instrucción no ha asignado el selector T local. • El primer byte no contiene el valor 0x0E. • El selector T local empieza por "SIMATIC-".
1	80B5	TCON: La instrucción solo permite el establecimiento pasivo de la conexión para el tipo de conexión 13 = UDP (el parámetro ActiveEstablished de la estructura TCON_xxx tiene el valor TRUE).
1	80B6	TCON: Error de parametrización en el parámetro ConnectionType del bloque de datos para la descripción de la conexión: • Solo válido con TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: 0x11, 0x0B y 0x13. • Solo válido con TCON_IP_RFC: 0x0C y 0x12.

ERROR	STATUS * (W#16#...)	Explicación
1	80B7	TCON: Con TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: • TCP (iniciativa local): el puerto remoto es "0". • TCP (establecimiento de conexión pasiva): el puerto local es "0". • UDP: el puerto local es "0". • La instrucción ha establecido la dirección IP del punto final del interlocutor en 0.0.0.0. TCON: Con TCON_IP_RFC: • La instrucción ha asignado el selector T local (local_tselector) o remoto (remote_tselector) con una longitud de más de 32 bytes. • Para TSelLength del selector T (local o remoto), la instrucción ha asignado una longitud mayor que 32. • Error en la longitud de la dirección IP del interlocutor de conexión específico. • La instrucción ha establecido la dirección IP del punto final del interlocutor en 0.0.0.0.
1	80B8	TCON: El parámetro ID de la descripción de la conexión local (estructura en el parámetro CONNECT) y el parámetro ID de la instrucción son diferentes.
1	80C3	TCON: Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: • La instrucción no puede establecer la conexión en este momento (TCON). • La instrucción no puede establecer la conexión porque los cortafuegos de la ruta de conexión no están abiertos para los puertos necesarios (TCON). • La interfaz está recibiendo nuevos parámetros (TCON y TDISCON). • La instrucción "TDISCON" está eliminando la conexión configurada (TCON).
1	80C5	TCON: El interlocutor remoto ha realizado una de las opciones siguientes: • Ha rechazado establecer la conexión • Ha deshecho la conexión • Ha finalizado activamente la conexión
1	80C6	TCON: La instrucción no puede acceder al interlocutor remoto (error de red).
1	80C7	TCON: Tiempo de ejecución excedido.
1	80C8	TCON: Una conexión (TCON) creada por el programa de usuario utiliza el valor en el parámetro ID. La conexión utiliza la misma ID, pero una configuración de conexión distinta en el parámetro CONNECT.
1	80C9	TCON: Error al validar el interlocutor remoto. El interlocutor remoto que desea establecer la conexión no coincide con el interlocutor definido de la estructura en el parámetro CONNECT.
1	80CE	TCON: La dirección IP de la interfaz local es 0.0.0.0.
1	80E0	TCON: La instrucción ha recibido un mensaje inadecuado o deficiente.
1	80E1	TCON: Error durante el handshake. Causas posibles: • Cancelación por parte del usuario. • Seguridad insuficiente. • La instrucción no soporta la negociación renovada. • La instrucción no soporta la versión de SSL/TLS. • Validación fallida del nombre de host.

ERROR	STATUS * (W#16#...)	Explicación
1	80E2	Certificado no soportado / certificado no válido / no hay certificado Causa posible: para el módulo en cuestión, la CPU no ha ajustado la hora del día o no ha sincronizado el módulo. Ejemplo: el ajuste predeterminado para la fecha del módulo es 1/1/2012 y la CPU no ha ajustado la fecha durante la puesta en marcha. El período de validez del certificado empieza el 20 de agosto de 2016 y finaliza el 20 de agosto de 2024. En este caso, la fecha del módulo está fuera del período de validez del certificado, por lo que este no es válido para el módulo.
1	80E3	Certificado rechazado.
1	80E4	No se ha encontrado ninguna autoridad de certificación válida.
1	80E5	Certificado caducado.
1	80E6	Errores de integridad en el protocolo de seguridad para la capa de transporte (TLSP)
1	80E7	Complemento no soportado en el certificado X.509-V3
1	80E9	La instrucción no soporta un servidor TLS sin un certificado de servidor.
1	80EA	La instrucción no soporta el protocolo DTLS (UDP).
1	80EB	Un cliente no puede solicitar un certificado de cliente.
1	80EC	El servidor no puede realizar la validación basándose en el subjectAlternateName (solo pueden hacerlo los clientes).
1	80ED	TLSServerCertRef_m-ID no válida
* Los códigos de error en el editor de programas se pueden mostrar como valores enteros o hexadecimales.		

TSEND y TRCV

Nota

Si se utiliza la comunicación Open User de PROFINET y se ejecuta una instrucción TSEND sin que el dispositivo remoto ejecute la instrucción TRCV correspondiente, es posible que la instrucción TSEND permanezca indefinidamente en un estado ocupado mientras espera que la instrucción TRCV reciba los datos. En este estado, la salida "Busy" de la instrucción TSEND está activada y la salida "Status" tiene un valor de "0x7002". Esto puede suceder si se transfieren más de 4096 bytes de datos. Este problema se soluciona la próxima vez que se ejecuta la instrucción TRCV.

Tabla 11-31 Instrucciones TSEND y TRCV

KOP / FUP	SCL	Descripción

	`"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 e ISO on TCP: TSEND envía datos mediante una conexión entre la CPU y un interlocutor.

	`"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 e ISO on TCP: TRCV recibe datos mediante una conexión entre un interlocutor y la CPU.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-32 Tipos de datos para los parámetros de TSEND y TRCV

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	TSEND: Inicia la tarea de transmisión cuando se produce un flanco ascendente. Los datos se transfieren desde el área que indican los parámetros DATA y LEN.
EN_R	IN	Bool	TRCV: Habilita la CPU para recibir, con EN_R = 1, TRCV está listo para recibir. La tarea de recepción se procesa.
ID	IN	CONN_OUC (Word)	Referencia a la conexión asociada. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
LEN	IN	UDInt	Número máximo de bytes que deben enviarse (TSEND) o recibirse (TRCV): • Predeterminado = 0: el parámetro DATA determina la longitud de los datos que deben enviarse (TSEND) o recibirse (TRCV). • Modo ad hoc = 65535: una longitud de datos variable está configurada para la recepción (TRCV).
ADHOC	IN	Bool	TRCV: Parámetro opcional (oculto) solicitud ad hoc para el tipo de conexión TCP.
DATA	IN_OUT	Variant	Puntero para enviar (TSEND) o recibir (TRCV) un área de datos, el área de datos contiene la dirección y la longitud. La dirección hace referencia a la memoria I, Q, M o a un DB.
DONE	OUT	Bool	TSEND: • 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.

Parámetro y tipo		Tipo de datos	Descripción
NDR	OUT	Bool	TRCV: • NDR = 0: Tarea no iniciada aún o en proceso. • NDR = 1: La tarea se ha ejecutado correctamente.
BUSY	OUT	Bool	• BUSY = 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva. • BUSY = 0: Tarea finalizada.
ERROR	OUT	Bool	ERROR = 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de estado, incluida la información de error. (Consulte los códigos de condición de error y estado en la tabla siguiente.)
RCVD_LEN	OUT	UDInt	TRCV: Cantidad de datos recibida realmente en bytes

Nota

La instrucción TSEND requiere una transición de bajo a alto en el parámetro de entrada REQ para iniciar una tarea de transmisión. El parámetro BUSY se pone a 1 durante el procesamiento. La finalización de la tarea de transmisión se indica poniendo a 1 los parámetros DONE o ERROR durante un ciclo. Durante este tiempo, se ignora cualquier transición de bajo a alto en el parámetro de entrada REQ.

Operaciones TRCV

La instrucción TRCV escribe los datos recibidos en un área de recepción especificada por las dos variables siguientes:

- Puntero al comienzo del área
- Longitud del área, o bien el valor suministrado en la entrada LEN si no es 0

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Se recomienda que los datos transmitidos por la instrucción TSEND tengan el mismo tamaño que el parámetro DATA de la instrucción TRCV.

Si se utiliza el ajuste predeterminado del parámetro LEN y hay que enviar los datos en segmentos más pequeños que el tamaño del parámetro DATA, se aplica lo siguiente. Se recomienda mantener el bit EN_R en "high" hasta que el TSEND correspondiente transfiera la cantidad adecuada de datos para rellenar el parámetro TRCV DATA. Si el tamaño de los datos transmitidos desde TSEND no es el mismo que el tamaño del parámetro TRCV DATA, TRCV permanece en estado ocupado (código de estado: 7002) mientras el bit EN_R esté en "high", hasta que el tamaño total de los datos transmitidos desde TSEND sea el mismo que el tamaño del parámetro TRCV DATA. Si el bit EN_R de TRCV se impulsa, tiene que impulsarse el mismo número de veces que se ejecuta TSEND para recibir los datos.

El búfer del parámetro TRCV DATA no muestra los nuevos datos recibidos hasta que el tamaño de los datos sea el mismo que el tamaño del búfer del parámetro DATA.

Tan pronto como se hayan recibido todos los datos de la tarea, TRCV los transferirá al área de recepción y pondrá NDR a 1.

Tabla 11-33 Entrada de datos en el área de recepción

Variante de protocolo	Entrada de datos en el área de recepción	Parámetro "connec-tion_type"	Valor del parámetro LEN	Valor del parámetro RCVD_LEN (bytes)
TCP	Modo ad hoc	B#16#11	Se seleccionan con la entrada ADHOC de la instrucción TRCV.	De 1 a 1472
TCP	Recepción de datos con la longitud especificada	B#16#11	0 (recomendado) o 1 a 8192, excepto 65535	De 1 a 8192
ISO on TCP	Modo ad hoc	B#16#12	65535	De 1 a 1472
ISO on TCP	Controlado por protoco-lo	B#16#12	0 (recomendado) o 1 a 8192, excepto 65535	De 1 a 8192

Nota

Modo ad hoc

El modo "ad hoc" solo está disponible con las variantes de protocolo TCP e ISO on TCP. Para configurar la instrucción TRCV para el modo ad hoc, ajuste el parámetro de entrada de la instrucción ADHOC. El área de recepción es idéntica al área especificada en el parámetro DATA. La longitud de los datos recibidos será transferida al parámetro RCVD_LEN. Inmediatamente después de recibir un bloque de datos, TRCV introduce los datos en el área de recepción y pone NDR a 1.

Si se guardan los datos en un DB "optimizado" (solo simbólico), únicamente es posible recibir datos en matrices de tipos de datos Byte, Char, USInt y SInt.

Nota

Importar al S7-1200 proyectos S7-300/400 de STEP 7 que contienen el modo "ad hoc"

En los proyectos S7-300/400 de STEP 7, el modo "ad hoc" se selecciona asignando el valor "0" al parámetro LEN. En S7-1200, configure la instrucción TRCV para el modo ad hoc ajustando el parámetro de entrada de la instrucción ADHOC.

Si se importa al S7-1200 un proyecto S7-300/400 de STEP 7 que contiene el modo "ad hoc", el parámetro LEN debe modificarse a "65535".

Tabla 11-34 Códigos de condición ERROR y STATUS para TSEND y TRCV

ERROR	STATUS	Descripción
0	0000	• Tarea de transmisión finalizada sin errores (TSEND) • Se han aceptado datos nuevos: La longitud actual de los datos recibidos se muestra en RCVD_LEN (TRCV).
0	7000	• No se está procesando ninguna tarea (TSEND) • El bloque no está listo para la recepción (TRCV)

ERROR	STATUS	Descripción
0	7001	• Inicio del procesamiento de la tarea, enviando los datos: Durante este procesamiento, el sistema operativo accede a los datos del área de emisión DATA (TSEND). • El bloque está listo para la recepción, se ha activado la tarea de recepción (TRCV).
0	7002	• Ejecución intermedia de la instrucción (REQ es irrelevante), procesando la tarea: El sistema operativo accede a los datos del área de emisión DATA durante este procesamiento (TSEND). • Ejecución intermedia de la instrucción, procesando la tarea de recepción: Los datos se escriben en el área de recepción durante este procesamiento. Por este motivo, un error puede generar datos incoherentes en el área de recepción (TRCV).
1	8085	• El parámetro LEN es mayor que el valor máximo admisible (TSEND) y (TRCV). • El parámetro LEN o DATA ha sido modificado desde la primera ejecución de la instrucción (TRCV).
1	8086	El parámetro ID está fuera del rango de direcciones admisible.
1	8088	El parámetro LEN excede el área de memoria especificada en DATA.
1	80A1	Error de comunicación: • La conexión indicada no se ha establecido todavía (TSEND y TRCV). • La conexión indicada se está deshaciendo. La transmisión o la tarea de recepción no es posible a través de esta conexión (TSEND y TRCV). • La interfaz se está reinicializando (TSEND). • La interfaz está recibiendo nuevos parámetros (TRCV).
1	80C3	Falta interna de recursos de conexión (Página 616): Ya se está procesando un bloque con esta ID en una clase de prioridad diferente.
1	80C4	Error de comunicación temporal: • La conexión con el interlocutor no se puede establecer en estos momentos. • La interfaz está recibiendo nuevos ajustes de parámetros o la conexión se está estableciendo.

Protocolos de conexión Ethernet

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Las instrucciones TSEND_C, TRCV_C, TSEND y TRCV soportan los protocolos Ethernet para TCP e ISO on TCP.

Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

11.5.8.12 TCONSettings

Igual que con V4.5 y TIA Portal V17, la instrucción "TCONSettings" se puede utilizar para ejecutar las siguientes funciones:

- Solicitar un ID de conexión para una nueva conexión OUC
- Solicitar un ID de conexión para una nueva conexión OUC y, al mismo tiempo, especificar una propiedad para esta conexión

- Leer una propiedad de una conexión OUC preparada o ya existente
- Especificar una propiedad para una conexión OUC preparada o ya existente

Con la instrucción "TCONSettings" se pueden leer o especificar las siguientes propiedades de conexión:

- Cómo terminar una conexión TCP

La instrucción "TCONSettings" es una instrucción asíncrona. Su ejecución se extiende a lo largo de varias llamadas. Se comienza a ejecutar con un flanco ascendente en el parámetro "REQ".

Los parámetros "Busy" y "Done" indican el estado de la tarea.

Si se produce un error durante la ejecución, los parámetros "Error" y "Status" lo indican.

Tabla 11-35 Tipos de datos TCONSettings para los parámetros

Parámetros y tipo		**Tipo de datos**	**Descripción**
REQ	Entrada	Bool	Petición de parámetro de control Activa la tarea con un flanco ascendente.
MODE	Entrada	USInt	Utilice el parámetro "Mode" para seleccionar la información que desee leer de la CPU: • 0: solicitar un ID de conexión para una nueva conexión OUC y, en caso necesario, especificar una propiedad para la conexión asociada (si en el parámetro OPTION hay presente un valor válido para una propiedad) • 1: leer una propiedad de la conexión OUC referenciada por ID • 2: especificar una propiedad de la conexión OUC referenciada por ID • de 3 a 255: reservado
DONE	Salida	Bool	Parámetro de estado con los valores siguientes: • 0: tarea no iniciada aún o en proceso. • 1: tarea finalizada sin errores. Este estado solo se muestra para un llamada.
BUSY	Salida	Bool	Parámetro de estado con los valores siguientes: • 0: tarea no iniciada aún o ya finalizada. • 1: tarea no finalizada aún. No se puede iniciar una tarea nueva con esta instancia.
ERROR	Salida	Bool	Parámetro de estado con los valores siguientes: • 0: no ha ocurrido ningún error. • 1: ha ocurrido un error durante el procesamiento. STATUS ofrece información detallada sobre el tipo de error. Este estado solo se muestra para un llamada.
STATUS	Salida	Word	Valor de retorno o información de error de la instrucción "TCONSettings".
ID	Entrada/salida	CONN_OUC	Referencia a la conexión: Nota: si MODE=0, ID es un parámetro de salida, si MODE=1 o MODE=2, ID es un parámetro de entrada.
OPTION	Entrada/salida	Variant	Puntero hacia la descripción de la propiedad de conexión que debe leerse o especificarse: • TCON_TCPTermination: cómo terminar la conexión TCP.

Parámetros BUSY, DONE y ERROR

Puede comprobar el estado de la tarea con los parámetros BUSY, DONE, ERROR y STATUS. El parámetro BUSY indica el estado de procesamiento. Con el parámetro DONE se comprueba si una tarea se ha ejecutado correctamente o no. El parámetro ERROR se activa si se producen errores durante la ejecución de "TCONSettings". La información de error se deposita en el parámetro STATUS.

En la tabla siguiente se muestra la relación entre los parámetros BUSY, DONE y ERROR:

BUSY	DONE	ERROR	Descripción
1	0	0	La tarea se está ejecutando.
0	1	0	La tarea se ha ejecutado correctamente.
0	0	1	La tarea ha finalizado con un error. La causa del error se deposita en el parámetro STATUS.
0	0	0	No se ha asignado ninguna tarea nueva.

Tabla 11-36 Códigos de condición TCONSettings para STATUS

STATUS (W#16#...)	Explicación
0000	"TCONSettings" se ha completado sin errores.
7000	No se está procesando ninguna tarea.
7001	Inicio de la ejecución de la tarea.
7002	Llamada intermedia (REQ irrelevante):
8086	El ID está fuera del rango permitido.
8087	Se ha alcanzado el número máximo de conexiones OUC; no es posible establecer más conexiones.
8089	OPTION no apunta a un tipo de datos válido o bien OPTION está vacío.
8090	OPTION apunta a una propiedad de conexión que no debe cambiarse para la conexión referenciada por ID.
8091	Valor no válido para MODE.
8092	Un valor del bloque de datos referenciado por OPTION no está permitido.
8093	Si MODE tiene el valor 0, ID también debe tener el valor 0.
809A	OPTION apunta a un tipo de datos que no está permitido para "TCONSettings".
80A3	ID apunta a un punto final de comunicación que no existe.
80B1	El parámetro OPTION ha sido modificado antes de que se completara la ejecución de "TCONSettings". No está permitido cambiar OPTION mientras "TCONSettings" se está ejecutando.
80C3	Se ha alcanzado el número máximo de tareas activas al mismo tiempo.

Número máximo de tareas activas al mismo tiempo.

El número máximo de tareas activas al mismo tiempo es idéntico al número máximo de conexiones OUC de la CPU

Consulte también

Nuevas funciones (Página 35)

Reservar un recurso de conexión

Llame TCONSettings con MODE=0. Asigne los parámetros relevantes del siguiente modo:

- Introduzca el valor NULL en el parámetro ID.
- Si no quiere especificar una propiedad para la conexión asociada, deje el parámetro OPTION vacío.
  Si quiere especificar una propiedad para la conexión asociada, asigne un valor válido al parámetro OPTION (Página 661).

Una vez que el parámetro DONE de TCONSettings sea TRUE, habrá un ID de conexión disponible para una nueva conexión OUC en el parámetro ID. Si especifica una propiedad en el parámetro OPTION, la conexión utilizará esa propiedad para la conexión. La instrucción TCONSettings utiliza un recurso de conexión OUC para el ID y crea los objetos de diagnóstico correspondientes. La instrucción TCONSettings ha preparado la conexión, pero los interlocutores externos aún no conocen la conexión.

No ha especificado ningún detalle para la conexión: ni el interlocutor, ni el protocolo, ni la interfaz, ni el DB con la descripción de la conexión.

Nota

Establecimiento de la conexión

TCONSettings no establece una conexión.

Establecer la conexión asociada

Si desea establecer la conexión correspondiente tras la ejecución correcta de "TCONSettings", proceda del siguiente modo:

1. Guarde el ID de conexión proporcionado por "TCONSettings".
2. Llame la instrucción "TCON" exactamente con ese ID.

El número de conexiones OUC disponibles no cambia porque la instrucción TCONSettings ya ha consumido la conexión.

Habilitar el ID de conexión y el recurso de conexión correspondiente

Si desea volver a activar el ID de conexión proporcionado por "TCONSettings" y el recurso de conexión correspondiente, llame la instrucción "TDISCON" exactamente con ese ID.

Cambios de la CPU al estado operativo STOP

Cuando la CPU cambia al estado operativo STOP, se vuelven a habilitar todos los ID de conexión proporcionados por "TCONSettings" y los recursos de conexión correspondientes.

Leer una propiedad de una conexión preparada o existente

Llame "TCONSettings" con MODE=1. Asigne los parámetros relevantes del siguiente modo:

- En el parámetro ID especifique la referencia a la conexión que desee.
- En el parámetro OPTION especifique la propiedad de conexión que desee leer.

Una vez que el parámetro DONE ha asumido el valor TRUE, los valores actuales de la propiedad deseada estarán disponibles en el área de datos especificada por OPTION.

Especificar una propiedad de una conexión preparada o existente

Llame "TCONSettings" con MODE=2. Asigne los parámetros relevantes del siguiente modo:

- En el parámetro ID especifique la referencia a la conexión a la que desea asignar una propiedad.
- En el parámetro OPTION indique qué propiedad de conexión desea especificar.

Una vez que el parámetro DONE ha adoptado el valor TRUE, a la conexión se le habrá asignado la propiedad deseada.

Conexiones creadas por instrucciones OUC y Modbus

Las instrucciones de la librería OUC que acaban con "_C" y las instrucciones de la librería MODBUS-TCP establecen conexiones llamando la instrucción "TCON" internamente. Este tipo de conexiones se pueden modificar con "TCONSettings" de igual modo que las conexiones creadas explícitamente llamando "TCON".

Propiedades de conexión que generalmente pueden leerse y especificarse

Las siguientes propiedades de conexión pueden leerse y especificarse con la instrucción "TCONSettings":

- Cómo terminar una conexión TCP.

Relación entre el protocolo o la interfaz y las propiedades de conexión que realmente pueden leerse o especificarse

No todos los protocolos o interfaces pueden leer o especificar todas las propiedades de conexión indicadas anteriormente. La tabla siguiente muestra qué propiedades de conexión son posibles para cada uno de los protocolos o interfaces.

Protocolo/interfaz	Terminar una conexión
Conexión programada:	Sí
Conexión configurada:	No [1)]
TCP	Sí
UDP	No [3)]
ISO on TCP	No
Interfaz de CPU	Sí
Interfaz de CP	No

Protocolo/interfaz	Terminar una conexión
Interfaz de CP virtual	Sí
[1] "TDISCON" no se puede llamar para una conexión configurada. Por ese motivo no existe ninguna forma de terminar la conexión de manera ordenada. [2] UDP carece de conexiones en el nivel de protocolo, por lo que no es necesaria ninguna terminación.	

Conflictos en la especificación de propiedades de conexión

Cada propiedad de conexión predefinible está permitida únicamente para protocolos o interfaces específicos. Por esa razón pueden existir conflictos entre la especificación de una propiedad de conexión por parte del usuario y el protocolo o interfaz deseados. En ese caso "TCONSettings" devuelve el valor W#16#8090 en el parámetro STATUS.

¿Cómo se termina una conexión TCP?

Una conexión TCP existente se puede terminar de las dos formas siguientes:

- Con un TCP-Reset (predeterminado)
  La conexión se cierra después de que se haya enviado la trama con el bit RST activado en el encabezado. Los recursos asociados se borran y habilitan inmediatamente. Los datos restantes ni se envían ni se transfieren al programa de usuario.

  Nota

  Terminar una conexión TCP en CPU S7-1500 con versión de firmware < V2.9 y CPU S7-1200 con versión de firmware < V4.5

  En las CPU S7-1500 con versión de firmware < V2.9 y en las CPU S7-1200 con versión de firmware < V4.5, una conexión TCP siempre se termina con un TCP-Reset.

- Con un TCP-Finish
  Si se ha configurado TCP-Finish como forma de terminar una conexión y se llama la instrucción "TDISCON", desde el punto de vista del usuario la conexión se cierra tras la terminación de "TDISCON" cuando DONE=TRUE, es decir, cuando el ID de la conexión vuelve a estar disponible. Sin embargo, en los niveles inferiores de la pila TCP/IP del módulo, los recursos aún continúan asignados durante un tiempo, igual que los objetos de diagnóstico pertenecientes a la conexión.
  Si se eliminan muchas conexiones con TCP-Finish y se reservan (con "TCONSettings") o establecen (con "TCON") conexiones antes de que concluya el temporizador para la habilitación de recursos, puede producirse una escasez de recursos.

Condiciones para TCP-Finish

Para terminar una conexión de forma ordenada utilizando TCP-Finish deben cumplirse las siguientes condiciones:

- El protocolo utilizado es TCP.
- La interfaz asociada está ubicada en la CPU.
- La razón para la terminación de la conexión es la llamada de la instrucción "TDISCON".

Nota

Terminar una conexión TCP durante la transición a STOP

Durante la transición a STOP las conexiones TCP siempre terminan con un TCP-Reset.

SDT para la terminación de conexiones: TCON_TCPTermination

El SDT para la terminación de conexiones tiene la siguiente estructura:

Parámetros	Tipo de datos	Valor inicial	Descripción
GracefulShut-down	Bool	FALSE	• FALSE: utilice un TCP-Reset para terminar la conexión. • TRUE: utilice un TCP-Finish para terminar la conexión.

11.5.8.13 Instrucciones Legacy TCON, TDISCON, TSEND y TRCV

Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones TCON, TDISCON, TSEND y TRCV solo podían trabajar con parámetros de conexión con estructuras conforme a "TCON_Param". Los conceptos generales son válidos para ambos juegos de instrucciones. Consulte las instrucciones Legacy individuales TCON, TDISCON, TSEND, y TRCV para obtener información sobre la programación.

Selección de la versión de las instrucciones TCON, TDISCON, TSEND y TRCV

Existen dos versiones de las instrucciones TCON, TDISCON, TSEND o TRCV disponibles en STEP 7:

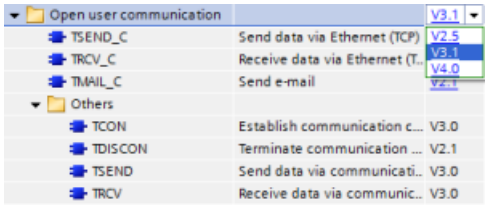
- Las versiones 2.5 y 3.1 estaban disponibles en STEP 7 Basic/Professional V13 y versiones anteriores.
- La versión 4.0 está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No utilice diferentes versiones de la instrucción en el mismo programa de la CPU.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



Para cambiar la versión de las instrucciones TCON, TDISCON, TSEND o TRCV, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción TCON, TDISCON, TSEND y TRCV en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción TCON, TDISCON, TSEND o TRCV seleccionada. La nueva instancia

del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción TCON, TDISCON, TSEND o TRCV en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione un árbol del proyecto TCON, TDISCON, TSEND o TRCV o instancia del FB o la FC, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción TCON, TDISCON, TSEND o TRCV.

Instrucciones Legacy TCON, TDISCON, TSEND y TRCV (comunicación TCP)

Comunicación Ethernet con los protocolos TCP e ISO on TCP

Nota

Instrucciones TSEND_C y TRCV_C

Para ayudar a simplificar la programación de la comunicación PROFINET/Ethernet, las instrucciones TSEND_C y TRCV_C combinan las funciones de las instrucciones TCON, TDISCON, TSEND y TRCV:

- TSEND_C combina las instrucciones TCON, TDISCON y TSEND.
- TRCV_C combina las instrucciones TCON, TDISCON y TRCV.

Las instrucciones siguientes controlan el proceso de comunicación:

- TCON establece la conexión TCP/IP entre el cliente y el servidor (CPU) PC.
- TSEND y TRCV permiten enviar y recibir datos.
- TDISCON deshace la conexión.

El tamaño mínimo de los datos que pueden transmitirse (TSEND) o recibirse (TRCV) es de un byte; el tamaño máximo es de 8192 bytes. TSEND no admite la transmisión de datos desde posiciones booleanas y TRCV no recibe datos en posiciones booleanas. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 187).

TCON, TDISCON, TSEND y TRCV funcionan de forma asíncrona, con lo que el procesamiento de la tarea abarca varias ejecuciones de la instrucción. Por ejemplo, una tarea para configurar y establecer una conexión se inicia ejecutando una instrucción TCON con el parámetro REQ = 1. Después se utilizan ejecuciones adicionales de TCON para vigilar la tarea y comprobar si ha finalizado con el parámetro DONE.

La tabla siguiente muestra la relación entre los parámetros BUSY, DONE y ERROR. Utilice la tabla para determinar el estado actual de la tarea:

Tabla 11-37 Interacción entre los parámetros BUSY, DONE y ERROR

BUSY	DONE	ERROR	Descripción
TRUE	Irrelevante	Irrelevante	La tarea se está procesando.
FALSE	TRUE	FALSE	La tarea se ha ejecutado correctamente.

BUSY	DONE	ERROR	Descripción
FALSE	FALSE	TRUE	La tarea se ha finalizado con un error. La causa del error se indica en el parámetro STATUS.
FALSE	FALSE	FALSE	No se ha asignado ninguna tarea nueva.

TCON y TDISCON

Nota

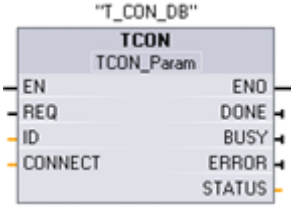
Inicializar los parámetros de comunicación

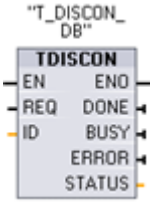
Después de introducir la instrucción TCON, utilice las "Propiedades" de la instrucción (Página 596) para configurar los parámetros de comunicación (Página 618). Cuando se introducen los parámetros de los interlocutores en la ventana de inspección, STEP 7 introduce los datos correspondientes en el DB de instancia de la instrucción.

Si desea utilizar un DB multiinstancia, debe configurar manualmente el DB en ambas CPUs.

Tabla 11-38 Instrucciones TCON y TDISCON

KOP / FUP		Descripción

	`"TCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_undef_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`connect:=_struct_inout_);`
		TCP e ISO on TCP: TCON inicia una conexión entre la CPU y un interlocutor.

	`"TDISCON_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`busy=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		TCP e ISO on TCP: TDISCON deshace una conexión entre la CPU y un interlocutor.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-39 Tipos de datos para los parámetros de TCON y TDISCON

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	El parámetro de control REQ inicia la tarea estableciendo la conexión que indica ID. La tarea comienza cuando se produce un flanco ascendente.
ID	IN	CONN_OUC (Word)	Referencia a la conexión que debe establecerse (TCON) o deshacerse (TDISCON) con el interlocutor remoto, o bien entre el programa de usuario y el nivel de comunicación del sistema operativo. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
CONNECT (TCON)	IN_OUT	TCON_Param	Puntero hacia la descripción de la conexión (Página 618)
DONE	OUT	Bool	• 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
BUSY	OUT	Bool	• 0: Tarea finalizada. • 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	OUT	Bool	Parámetros de estado con los valores siguientes: • 0: No hay error • 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de estado, incluida la información de error. (Consulte los códigos de condición de error y estado en la tabla siguiente.)

Ambos interlocutores ejecutan la instrucción TCON para configurar y establecer la conexión. El punto final activo y el punto final pasivo de la comunicación se especifican mediante parámetros. Una vez configurada y establecida la conexión, la CPU la mantiene y la vigila automáticamente.

Si la conexión se deshace p. ej. debido a una interrupción de la línea o por el interlocutor remoto, el interlocutor activo intenta restablecer la conexión configurada. No es necesario volver a ejecutar TCON.

Una conexión existente se deshace y la conexión configurada se elimina cuando se ejecuta la instrucción TDISCON o cuando la CPU pasa a estado operativo STOP. Para configurar y restablecer la conexión es preciso ejecutar TCON de nuevo.

Tabla 11-40 Códigos de condición ERROR y STATUS para TCON y TDISCON

ERROR	STATUS	Descripción
0	0000	La conexión se ha establecido correctamente.
0	7000	No se está procesando ninguna tarea
0	7001	Iniciar procesamiento de la tarea, estableciendo la conexión (TCON) o deshaciendo la conexión (TDISCON)
0	7002	Llamada intermedia (REQ es irrelevante), estableciendo la conexión (TCON) o deshaciendo la conexión (TDISCON)
1	8086	El parámetro ID está fuera del rango de direcciones admisible.
1	8087	TCON: Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.

ERROR	STATUS	Descripción
1	809B	TCON: La "local_device_id" indicada en la descripción de la conexión no concuerda con la CPU.
1	80A1	TCON: El usuario ya está utilizando el puerto o la conexión.
1	80A2	TCON: El sistema está utilizando el puerto local o remoto.
1	80A3	Se está intentando restablecer una conexión existente (TCON) o deshacer una conexión no existente (TDISCON).
1	80A4	TCON: La dirección IP del punto final remoto de la conexión no es válida; concuerda con la dirección IP local del interlocutor.
1	80A5	TCON: ID de conexión (Página 616) en uso.
1	80A7	TCON: Error de comunicación: TDISCON se ha ejecutado antes de finalizar TCON. TDISCON debe deshacer primero por completo la conexión referenciada por la ID.
1	80B2	TCON: El parámetro CONNECT apunta a un bloque de datos que se ha generado con el atributo "Depositar solo en la memoria de carga".
1	80B4	TCON: Si se utiliza ISO on TCP (connection_type = B#16#12) para establecer una conexión pasiva, el código de condición 80B4 advierte de que el TSAP introducido no cumple uno de los siguientes requisitos para la dirección: • Si la longitud del TSAP local es 2 y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 o 01. • Si la longitud del TSAP local es 3 o superior y el valor de la ID TSAP es E0 o E1 (hexadecimal) para el primer byte, el segundo byte deberá ser 00 o 01 y todos los demás bytes deberán ser caracteres ASCII válidos. • Si la longitud del TSAP local es 3 o superior y el valor del primer byte de la ID TSAP no es E0 ni E1 (hexadecimal), todos los demás bytes de la ID TSAP deberán ser caracteres ASCII válidos. Los caracteres ASCII válidos son valores de byte comprendidos entre 20 y 7E (hexadecimal).
1	80B5	TCON: El tipo de conexión "13 = UDP" solo permite un establecimiento pasivo de la conexión.
1	80B6	TCON: Error de asignación en el parámetro CONNECTION_TYPE del SDT TCON_Param.
1	80B7	TCON: Error en uno de los parámetros siguientes del bloque de datos para la descripción de la conexión: • block_length • local_tsap_id_len • rem_subnet_id_len • rem_staddr_len • rem_tsap_id_len • next_staddr_len Nota: cuando se utiliza TCON en modo TCP pasivo, LOCAL_TSAP_ID_LEN debe ser "2" y REM_TSAP_ID_LEN debe ser "0".
1	80B8	TCON: Los parámetros de la descripción de la conexión local y el parámetro ID son diferentes.

ERROR	STATUS	Descripción
1	80C3	TCON: Se están utilizando todos los recursos de conexión.
1	80C4	Error de comunicación temporal: • La conexión no se puede establecer en estos momentos (TCON). • La instrucción TDISCON (TCON) está deshaciendo la conexión configurada. • La conexión se está estableciendo (TDISCON). • La interfaz está recibiendo nuevos parámetros (TCON y TDISCON).

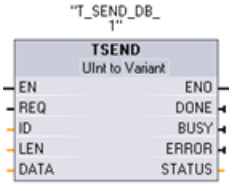
TSEND y TRCV

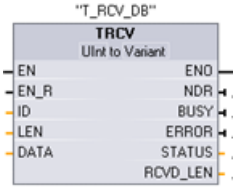
Nota

Si se utiliza la comunicación Open User de PROFINET y se ejecuta una instrucción TSEND sin que el dispositivo remoto ejecute la instrucción TRCV correspondiente, es posible que la instrucción TSEND permanezca indefinidamente en un estado ocupado mientras espera que la instrucción TRCV reciba los datos. En este estado, la salida "Busy" de la instrucción TSEND está activada y la salida "Status" tiene un valor de "0x7002". Esto puede suceder si se transfieren más de 4096 bytes de datos. Este problema se soluciona la próxima vez que se ejecuta la instrucción TRCV.

Tabla 11-41 Instrucciones TSEND y TRCV

KOP / FUP	SCL	Descripción

	`"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 e ISO on TCP: TSEND envía datos mediante una conexión entre la CPU y un interlocutor.

	`"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 e ISO on TCP: TRCV recibe datos mediante una conexión entre un interlocutor y la CPU.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 11-42 Tipos de datos para los parámetros de TSEND y TRCV

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	TSEND: Inicia la tarea de transmisión cuando se produce un flanco ascendente. Los datos se transfieren desde el área que indican los parámetros DATA y LEN.
EN_R	IN	Bool	TRCV: Habilita la CPU para recibir, con EN_R = 1, TRCV está listo para recibir. La tarea de recepción se procesa.
ID	IN	CONN_OUC (Word)	Referencia a la conexión asociada. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF
LEN	IN	UInt	Número máximo de bytes que deben enviarse (TSEND) o recibirse (TRCV): • Predeterminado = 0: el parámetro DATA determina la longitud de los datos que deben enviarse (TSEND) o recibirse (TRCV). • Modo ad hoc = 65535: una longitud de datos variable está configurada para la recepción (TRCV).
DATA	IN_OUT	Variante	Puntero para enviar (TSEND) o recibir (TRCV) un área de datos, el área de datos contiene la dirección y la longitud. La dirección hace referencia a la memoria I, Q, M o a un DB.
DONE	OUT	Bool	TSEND: • 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
NDR	OUT	Bool	TRCV: • NDR = 0: Tarea no iniciada aún o en proceso. • NDR = 1: La tarea se ha ejecutado correctamente.
BUSY	OUT	Bool	• BUSY = 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva. • BUSY = 0: Tarea finalizada.
ERROR	OUT	Bool	ERROR = 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de estado, incluida la información de error. (Consulte los códigos de condición de error y estado en la tabla siguiente.)
RCVD_LEN	OUT	Int	TRCV: Cantidad de datos recibida realmente en bytes

Nota

La instrucción TSEND requiere una transición de bajo a alto en el parámetro de entrada REQ para iniciar una tarea de transmisión. El parámetro BUSY se pone a 1 durante el procesamiento. La finalización de la tarea de transmisión se indica poniendo a 1 los parámetros DONE o ERROR durante un ciclo. Durante este tiempo, se ignora cualquier transición de bajo a alto en el parámetro de entrada REQ.

Operaciones TRCV

La instrucción TRCV escribe los datos recibidos en un área de recepción especificada por las dos variables siguientes:

- Puntero al comienzo del área
- Longitud del área, o bien el valor suministrado en la entrada LEN si no es 0

Nota

El ajuste predeterminado del parámetro LEN (LEN = 0) utiliza el parámetro DATA para determinar la longitud de los datos que se están transmitiendo. Se recomienda que los datos transmitidos por la instrucción TSEND tengan el mismo tamaño que el parámetro DATA de la instrucción TRCV.

Si se utiliza el ajuste predeterminado del parámetro LEN y hay que enviar los datos en segmentos más pequeños que el tamaño del parámetro DATA, se aplica lo siguiente. Se recomienda mantener el bit EN_R en "high" hasta que el TSEND correspondiente transfiera la cantidad adecuada de datos para rellenar el parámetro TRCV DATA. Si el tamaño de los datos transmitidos desde TSEND no es el mismo que el tamaño del parámetro TRCV DATA, TRCV permanece en estado ocupado (código de estado: 7002) mientras el bit EN_R esté en "high", hasta que el tamaño total de los datos transmitidos desde TSEND sea el mismo que el tamaño del parámetro TRCV DATA. Si el bit EN_R de TRCV se impulsa, tiene que impulsarse el mismo número de veces que se ejecuta TSEND para recibir los datos.

El búfer del parámetro TRCV DATA no muestra los nuevos datos recibidos hasta que el tamaño de los datos sea el mismo que el tamaño del búfer del parámetro DATA.

Tan pronto como se hayan recibido todos los datos de la tarea, TRCV los transferirá al área de recepción y pondrá NDR a 1.

Tabla 11-43 Entrada de datos en el área de recepción

Variante de protocolo	Entrada de datos en el área de recepción	Parámetro "connection_type"	Valor del parámetro LEN	Valor del parámetro RCVD_LEN (bytes)
TCP	Modo ad hoc	B#16#11	65535	1 a 1472
TCP	Recepción de datos con la longitud especificada	B#16#11	0 (recomendado) o 1 a 8192, excepto 65535	1 a 8192
ISO on TCP	Modo ad hoc	B#16#12	65535	1 a 1472
ISO on TCP	Controlado por protocolo	B#16#12	0 (recomendado) o 1 a 8192, excepto 65535	1 a 8192

Nota

Modo ad hoc

El modo "ad hoc" solo está disponible con las variantes de protocolo TCP e ISO on TCP. El modo "ad hoc" se establece asignando el valor "65535" al parámetro LEN. El área de recepción es idéntica al área especificada en el parámetro DATA. La longitud de los datos recibidos será transferida al parámetro RCVD_LEN. Inmediatamente después de recibir un bloque de datos, TRCV introduce los datos en el área de recepción y pone NDR a 1.

Si se guardan los datos en un DB "optimizado" (solo simbólico), únicamente es posible recibir datos en matrices de tipos de datos Byte, Char, USInt y SInt.

Nota

Importar al S7-1200 proyectos S7-300/400 de STEP 7 que contienen el modo "ad hoc"

En los proyectos S7-300/400 de STEP 7, el modo "ad hoc" se selecciona asignando el valor "0" al parámetro LEN. En el S7-1200, el modo "ad hoc" se establece asignando el valor "65535" al parámetro LEN.

Si se importa al S7-1200 un proyecto S7-300/400 de STEP 7 que contiene el modo "ad hoc", el parámetro LEN debe modificarse a "65535".

Códigos de condición TSEND , TRCV Error y Status

ERROR	STATUS	Descripción
0	0000	• Tarea de transmisión finalizada sin errores (TSEND) • Se han aceptado datos nuevos: La longitud actual de los datos recibidos se muestra en RCVD_LEN (TRCV).
0	7000	• No se está procesando ninguna tarea (TSEND) • El bloque no está listo para la recepción (TRCV)
0	7001	• Inicio del procesamiento de la tarea, enviando los datos: Durante este procesamiento, el sistema operativo accede a los datos del área de emisión DATA (TSEND). • El bloque está listo para la recepción, se ha activado la tarea de recepción (TRCV).
0	7002	• Ejecución intermedia de la instrucción (REQ es irrelevante), procesando la tarea: El sistema operativo accede a los datos del área de emisión DATA durante este procesamiento (TSEND). • Ejecución intermedia de la instrucción, procesando la tarea de recepción: Los datos se escriben en el área de recepción durante este procesamiento. Por este motivo, un error puede generar datos incoherentes en el área de recepción (TRCV).
1	8085	• El parámetro LEN es mayor que el valor máximo admisible (TSEND) y (TRCV). • El parámetro LEN o DATA ha sido modificado desde la primera ejecución de la instrucción (TRCV).
1	8086	El parámetro ID está fuera del rango de direcciones admisible.
1	8088	El parámetro LEN excede el área de memoria especificada en DATA.

ERROR	STATUS	Descripción
1	80A1	Error de comunicación: • La conexión indicada no se ha establecido todavía (TSEND y TRCV). • La conexión indicada se está deshaciendo. La transmisión o la tarea de recepción no es posible a través de esta conexión (TSEND y TRCV). • La interfaz se está reinicializando (TSEND). • La interfaz está recibiendo nuevos parámetros (TRCV).
1	80C3	Falta interna de recursos: Ya se está procesando un bloque con esta ID en una clase de prioridad diferente.
1	80C4	Error de comunicación temporal: • La conexión con el interlocutor no se puede establecer en estos momentos. • La interfaz está recibiendo nuevos ajustes de parámetros o la conexión se está estableciendo.

Protocolos de conexión Ethernet

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Las instrucciones TSEND_C, TRCV_C, TSEND y TRCV soportan los protocolos Ethernet para TCP e ISO on TCP.

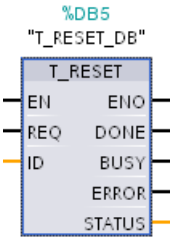
Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

11.5.8.14 Instrucción T_RESET (Finalizar y restablecer una conexión existente)

La instrucción "T_RESET" finaliza y restablece una conexión existente:

Tabla 11-44 Instrucción T_RESET

KOP / FUP	SCL	Descripción

	`"T_RESET_DB"(`
	`req:=_bool_in_,`
	`id:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_);`
		Utilice la instrucción T_RESET para finalizar y restablecer una conexión existente.

Los puntos finales locales de la conexión se conservan. Se generan automáticamente:

- Si se ha configurado y cargado una conexión en la CPU.
- Si el programa de usuario ha generado una conexión, por ejemplo, llamando a la instrucción "TCON (Página 652)".

La instrucción "T_RESET" se puede ejecutar para todos los tipos de conexión al margen de si se ha usado la interfaz local de la CPU o la interfaz de un CM/CP para la conexión, aunque hay que exceptuar las conexiones para transferencia de datos en modo ad hoc con TCP, puesto que estas conexiones no se pueden referenciar con una ID de conexión.

Una vez que se haya llamado la instrucción "T_RESET" con el parámetro REQ, se deshará la conexión especificada con el parámetro ID y, si es necesario, se borrará el búfer de transmisión y recepción de datos. Al cancelar la conexión también se cancela cualquier transferencia de datos en curso. Por tanto, existe el riesgo de que se pierdan datos si la transferencia de datos está en curso. A continuación, la CPU definida como interlocutor de conexión activa intentará automáticamente restaurar la conexión de comunicación interrumpida. Por consiguiente, no es necesario llamar la instrucción "TCON (Página 652)" para restablecer la conexión de comunicación.

Los parámetros de salida DONE, BUSY y STATUS indican el estado de la tarea.

Tipos de datos para los parámetros

En la tabla siguiente se muestran los parámetros de la instrucción T_RESET:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L, T, C o constante	El parámetro de control REQUEST inicia la tarea de deshacer la conexión especificada por ID. La tarea se inicia con un flanco ascendente.
ID	Input	CONN_OUC (WORD)	L, D o constante	Referencia a la conexión del interlocutor pasivo que se va a deshacer. ID debe ser igual a la ID de parámetro correspondiente en la descripción de conexión local. Rango de valores: W#16#0001 a W#16#0FFF
DONE	Output	BOOL	I, Q, M, D, L	Parámetro de estado DONE • 0: tarea no iniciada aún o en ejecución. • 1: tarea ejecutada sin errores.
BUSY	Output	BOOL	I, Q, M, D, L	Parámetro de estado BUSY • 0: tarea finalizada. • 1: tarea no finalizada aún.
ERROR	Output	BOOL	I, Q, M, D, L	Parámetro de estado ERROR • 0: no ha ocurrido ningún error. • 1: ha ocurrido un error durante el procesamiento. El parámetro STATUS ofrece información detallada sobre el tipo de error.
STATUS	Output	WORD	I, Q, M, D, L	Parámetro de estado STATUS Información de error (véase la tabla de parámetros de "STATUS").

Parámetro STATUS

Bit de error	STATUS* (W#16#...)	Descripción
0	0000	Sin errores.
0	0001	No se ha establecido la conexión.
0	7001	Se ha iniciado la interrupción de la conexión.

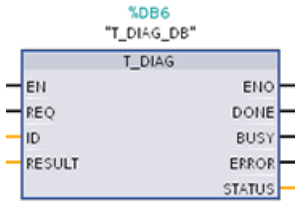
Bit de error	STATUS* (W#16#...)	Descripción
0	7002	Se está deshaciendo la conexión.
1	8081	Conexión desconocida especificada en el parámetro ID.

11.5.8.15 Instrucción T_DIAG (Comprueba el estado de la conexión y lee información)

La instrucción "T_DIAG" comprueba el estado de una conexión y lee información adicional del punto final local de esta conexión:

Tabla 11-45 Instrucción T_DIAG

KOP / FUP	SCL	Descripción

	`"T_DIAG_DB"(`
	`req:=_bool_in_,`
	`id:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	
	`status=>_dword_out_);`
		Use la instrucción T_DIAG para comprobar el estado de una conexión y leer información adicional del punto final local de esta conexión.

La instrucción "T_DIAG" funciona del siguiente modo:

- La conexión está referenciada por el parámetro ID. Puede leer tanto los puntos finales de conexión configurados en el editor de conexión como los puntos finales de conexión programados (por ejemplo con la instrucción "TCON").
  Los puntos finales de conexión temporales (por ejemplo, los puntos finales creados cuando se conecta a una estación de ingeniería) no se pueden diagnosticar, puesto que no se genera ninguna ID de conexión en este proceso.
- La información de la conexión leída se almacena en una estructura referenciada por el parámetro RESULT.
- El parámetro de salida STATUS indica si se ha podido leer la información de conexión. La información de conexión de la estructura en el parámetro RESULT solo es válida si se ha finalizado la instrucción "T_DIAG" con STATUS = W#16#0000 y ERROR = FALSE.
  La información de conexión no se puede evaluar si se produce un error.

Posible información de conexión

La estructura "TDiag_Status" se puede usar para leer la información de conexión en el parámetro RESULT. La estructura TDiag_Status solo contiene la información más importante sobre un punto final de una conexión (por ejemplo, el protocolo usado, el estado de conexión y el número de bytes de datos transmitidos o recibidos).

La estructura y los parámetros de la estructura TDiag_Status se describen más adelante (consulte la tabla de estructuras de "TDIAG_Status").

Tipos de datos para los parámetros

En la tabla siguiente se muestran los parámetros de la instrucción T_DIAG:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L, T, C o constante	Inicia la instrucción para comprobar la conexión especificada en el parámetro ID cuando se produce un flanco ascendente.
ID	Input	CONN_OUC (WORD)	L, D o constante	Referencia a la conexión asignada. Rango de valores: W#16#0001 a W#16#0FFF
RESULT	InOut	VARIANT	D	Puntero a la estructura en la que se almacena la información de conexión. La estructura TDiag_Status se puede usar en el parámetro RESULT (para obtener una descripción, véase la tabla de estructuras de "TDIAG_Status").
DONE	Output	BOOL	I, Q, M, D, L	Parámetro de estado: • 0: instrucción no iniciada aún o en proceso. • 1: instrucción ejecutada sin errores.
BUSY	Output	BOOL	I, Q, M, D, L	Parámetro de estado: • 0: instrucción no iniciada aún o ya finalizada. • 1: instrucción no finalizada aún. No se puede iniciar una tarea nueva.
ERROR	Output	BOOL	I, Q, M, D, L	Parámetro de estado: • 0: sin errores. • 1: se ha producido un error.
STATUS	Output	WORD	I, Q, M, D, L	Estado de la instrucción

Parámetros BUSY, DONE y ERROR

Puede comprobar el estado de la ejecución de instrucción "T_DIAG" con los parámetros BUSY, DONE, ERROR y STATUS. El parámetro BUSY indica el estado de procesamiento. El parámetro DONE se usa para comprobar si se ha ejecutado o no correctamente una instrucción. El parámetro ERROR se establece si se producen errores durante la ejecución de "T_DIAG".

En la tabla siguiente se muestra la relación entre los parámetros BUSY, DONE y ERROR.

BUSY	DONE	ERROR	Descripción
1	-	-	La instrucción se está procesando.
0	1	0	La instrucción se ha ejecutado correctamente. Los datos de la estructura referenciados por RESULT solo son válidos en un caso como este.
0	0	1	Instrucción finalizada con un error. La causa del error se genera en el parámetro STATUS.
0	0	0	No se ha asignado una nueva instrucción.

Parámetro STATUS

En la tabla siguiente se explican los valores en el parámetro STATUS:

Bit de error	STATUS* (W#16#...)	Descripción
0	0000	La instrucción "T_DIAG" se ha ejecutado correctamente. Los datos de la estructura referenciada en el parámetro RESULT se pueden evaluar.
0	7000	No se está procesando ninguna instrucción.
0	7001	Se ha iniciado el procesamiento de la instrucción.
0	7002	Se está leyendo la información de conexión (el parámetro REQ es irrelevante).
1	8086	El valor en el parámetro ID está fuera del rango válido (de W#16#0001 a W#16#0FFF).
1	8089	El parámetro RESULT apunta a un tipo de datos no válido (solo en las estructuras TDIAG_Status y TDIAG_StatusExt).
1	80A3	El parámetro ID hace referencia a un punto final de conexión que no existe. En las conexiones programadas, este error también se produce después de llamar a la instrucción "TDISCON".
1	80C4	Error interno. El acceso al punto final de conexión no está temporalmente disponible.

Estructura TDIAG_Status

En la tabla siguiente se ofrece información sobre la forma de la estructura TDIAG_Status. El valor de cada elemento solo es válido si se ha ejecutado la instrucción sin errores. Si se produce un error, el contenido de los parámetros no cambiará.

Nombre	Tipo de datos	Descripción
Los siguientes parámetros se encuentran en la estructura TDIAG_Status:		
InterfaceID	HW_ANY	ID de interfaz (LADDR) de la CPU o del CM/CP.
ID	CONN_OUC	ID de la conexión diagnosticada. Después de que una llamada se haya realizado correctamente, el valor de este elemento será idéntico a la ID de parámetro de la instrucción "T_DIAG".

Nombre	Tipo de datos	Descripción
ConnectionType	BYTE	Tipo de protocolo usado para la conexión: • 0x01: No utilizado. • ... • 0x0B: Protocolo TCP (IP_v4) • 0x0C: Protocolo ISO on TCP (RFC1006) • 0x0D: Protocolo TCP (DNS) • 0x0E: Protocolo de marcado • 0x0F: Protocolo WDC • 0x10: Protocolo SMTP • 0x11: Protocolo TCP • 0x12: Protocolo TCP e ISO on TCP (RFC1006) • 0x13: Protocolo UDP • 0x14: Reservado • 0x15: Protocolo de acceso a bus PROFIBUS (FDL) • 0x16: Protocolo de transporte ISO 8073 (ISO nativo) • ... • 0x20: Protocolo SMTP o SMTPS basado en IPv4 • 0x21: Protocolo SMTP o SMTPS basado en IPv6 • 0x22: Protocolo SMTP o SMTPS basado en FQDN (**F**ully **Q**ualified **D**omain **N**ame) • ... • 0x70: Conexión S7 • Otros: Reservado
ActiveEstablished	BOOL	• FALSE: local, punto final de la conexión pasiva • TRUE: local, punto final de la conexión activa
State	BYTE	Estado actual del punto final de conexión • 0x00: No utilizado. • 0x01: Conexión deshecha. Estado temporal, por ejemplo, después de llamar a la instrucción "T_RESET". Seguidamente, el sistema intenta volver a establecer la conexión automáticamente. • 0x02: El punto final de conexión activa está intentando establecer una conexión con el interlocutor remoto. • 0x03: El punto final de conexión pasiva está a la espera de establecer la conexión con el interlocutor remoto. • 0x04: Conexión establecida. • 0x05: Se está deshaciendo la conexión. Esto puede deberse a que se ha llamado a la instrucción "T_RESET" o "T_DISCON". Otros posibles motivos son errores de protocolo y saltos de línea. • 0x06..0xFF: No utilizado.

Nombre	Tipo de datos	Descripción
Kind	BYTE	Modo del punto final de conexión: • 0x00: No utilizado. • 0x01: Conexión estática configurada que se ha configurado y cargado en la CPU. • 0x02: Conexión dinámica configurada que se ha configurado y cargado en la CPU (no se admite actualmente). • 0x03: Conexión programada generada en el programa de usuario con la instrucción "TCON"". Una llamada de la instrucción "TDISCON" o una transición al estado de STOP de la CPU ha destruido el punto final de la conexión. • 0x04: Temporal, conexión dinámica establecida por la estación de ingeniería (ES) o la estación de operador (OS), por ejemplo (este tipo de conexión no puede diagnosticarse actualmente porque no hay ninguna ID). • 0x05..0xFF: No utilizado.
SentBytes	UDINT	Número de bytes transmitidos.
ReceivedBytes	UDINT	Número de bytes recibidos.

11.5.8.16 Instrucción TMAIL_C (Envía un correo electrónico utilizando la interfaz Ethernet de la CPU)

Vista general

La instrucción TMAIL_C se utiliza para enviar un correo electrónico a través de la interfaz Ethernet de la CPU S7-1200.

La instrucción TMAIL_C tiene dos funcionalidades:

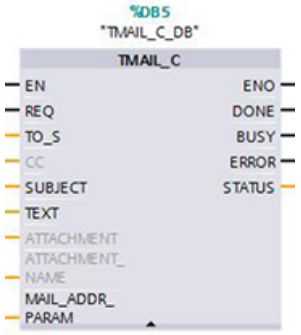
- Correo electrónico a través de la interfaz de la CPU
- Correo electrónico a través de una interfaz del CP

Para utilizar la instrucción TMAIL_C es necesario cumplir los siguientes requisitos:

- El hardware está configurado.
- La infraestructura de red permite la conexión de comunicación con el servidor de correo.

Tabla 11-46 Instrucción TMAIL_C

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción TMAIL_C envía un correo electrónico a través de la interfaz Ethernet de la CPU S7-1200.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

El contenido del correo electrónico y los datos de conexión se definen mediante los parámetros siguientes:

- Las direcciones de los destinatarios con los parámetros TO_S y CC .
- El contenido del correo electrónico con los parámetros SUBJECT y TEXT.
- Anexo opcional con punteros VARIANT en los parámetros ATTACHMENT y ATTACHMENT_NAME.
- Los datos de conexión, el direccionamiento y la autenticación para el servidor de correo en el parámetro MAIL_ADDR_PARAM (Página 687).

A partir de la versión V6.0 o posterior de TMAIL_C y la versión de firmware V4.4 o posterior de la CPU S7-1200 es posible utilizar la instrucción TMAIL_C para enviar un correo electrónico con comunicación segura a través de un puerto Ethernet integrado de la CPU S7-1200. Los datos necesarios para el proceso de envío se definen en el parámetro MAIL_ADDR_PARM (Página 687) con el SDT TMail_V4_SEC o TMail_QDN_SEC.

No es posible enviar un SMS directamente con la instrucción TMAIL_C. Si el servidor de correo puede reenviar o no el correo electrónico como SMS dependerá del proveedor de telecomunicaciones.

Funcionamiento de la instrucción

El envío de un correo electrónico se inicia con un cambio de flanco de "0" a "1" para el parámetro REQ.

La instrucción TMAIL_C indica el estado de la orden con los parámetros de salida "BUSY", "DONE", "ERROR" y "STATUS".

La instrucción TMAIL_C funciona de forma asíncrona, lo que significa que su ejecución abarca varias llamadas. Debe especificar una instancia cuando se invoque la instrucción "TMAIL_C".

En los siguientes casos, se perderá la conexión con el servidor de correo electrónico:

- Si la CPU cambia a STOP mientras la instrucción TMAIL_C está activa.
- Si se producen problemas de comunicación en el bus Industrial Ethernet.
  En este caso, la transferencia del correo electrónico se interrumpirá y no llegará a su destinatario.

La conexión también se cancela una vez que la instrucción se ha ejecutado correctamente y se ha enviado el correo electrónico.

ATENCIÓN
Cambio de los programas de usuario Solo es posible cambiar las partes del programa de usuario que afectan directamente a las llamadas de TMAIL_C si se dan las siguientes condiciones: • La CPU está en modo STOP. • No se está enviando ningún correo electrónico (REQ = 0 y BUSY = 0). Esto tiene que ver, en concreto, con el borrado y la sustitución de los bloques de programa que contienen llamadas de TMAIL_C o llamadas para la instancia de TMAIL_C. Si se ignora esta restricción, los recursos de conexión pueden quedar bloqueados. El sistema de automatización puede cambiar a un estado indefinido con las funciones de comunicación TCP/IP por medio de Industrial Ethernet. Se necesita reiniciar la CPU en frío o en caliente después de transferir los cambios.

Coherencia de datos

La instrucción TMAIL_C aplica los parámetros TO_S, CC, SUBJECT, TEXT, ATTACHMENT y MAIL_ADDR_PARAM mientras está en ejecución, lo que significa que solo se pueden cambiar una vez que se ha ejecutado la orden (BUSY = 0).

Autenticación SMTP

La autenticación hace referencia a un procedimiento de verificación de la identidad, por ejemplo, con una solicitud de contraseña.

Si se utiliza la interfaz de la CPU S7-1200, la instrucción TMAIL_C soporta el procedimiento de autenticación SMTP AUTH-LOGIN, que requieren la mayoría de servidores de correo. Para más

información sobre el procedimiento de autenticación del servidor de correo, consulte el manual del servidor de correo o el sitio web de su proveedor de servicios de Internet.

- Antes de poder usar el procedimiento de autenticación AUTH-LOGIN, la instrucción TMAIL_C necesita el nombre de usuario que va a usar para iniciar sesión en el servidor de correo. Este nombre de usuario corresponde al nombre de usuario que ha usado para configurar una cuenta de correo en el servidor de correo. Se transfiere mediante el parámetro UserName a la estructura del parámetro MAIL_ADDR_PARAM.
  Si no se especifica ningún nombre de usuario en el parámetro MAIL_ADDR_PARAM, no se usará el procedimiento de autenticación AUTH-LOGIN. A continuación, el correo electrónico se envía sin autenticación.
- Para iniciar sesión, la instrucción TMAIL_C también necesita la contraseña asociada. Esta contraseña corresponde a la contraseña que ha especificado cuando ha configurado su cuenta de correo. Se transfiere mediante el parámetro PassWord a la estructura en el parámetro MAIL_ADDR_PARAM.

Tipos de datos para los parámetros

La tabla siguiente muestra los parámetros de la instrucción TMAIL_C:

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
REQ	Input	BOOL	I, Q, M, D, L, T, C o constante	Parámetro de control REQUEST: Activa el envío de un correo electrónico cuando se produce un flanco ascendente.
TO_S (Página 697)	Input	STRING	D	Direcciones de destinatarios STRING con una longitud máxima de 180 caracteres (bytes). Para saber el formato de la dirección de correo electrónico, consulte el ejemplo en la descripción del parámetro.
CC (Página 697)	Input	STRING	D	Direcciones de destinatarios CC (opcional) STRING con una longitud máxima de 180 caracteres (bytes). El mismo formato de dirección de correo electrónico que para el parámetro TO_S. Si aquí se asigna una cadena vacía, la instrucción no envía el correo electrónico al destinatario CC.
SUBJECT	Input	STRING	D	Asunto del correo electrónico STRING con una longitud máxima de 180 caracteres (bytes).
TEXT	Input	STRING	D	Texto del correo electrónico (opcional) STRING con una longitud máxima de 180 caracteres (bytes). Si en este parámetro se asigna una cadena vacía, la instrucción envía el correo electrónico sin texto.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
ATTACHMENT	Input	VARIANT	D	Anexo de correo electrónico (opcional) Referencia a un campo de carácter, byte, palabra, palabra doble o cadena (ArrayOfChar, ArrayOfByte, ArrayOfWord, ArrayOfDWord, or String) de una longitud máxima de 64 KB. Nota: si en el parámetro ATTACHMENT no se asigna ningún valor o se asigna una cadena vacía, la instrucción envía el correo electrónico sin anexo.
ATTACHMENT_NAME	Input	VARIANT	D	Nombre de anexo de correo electrónico (opcional) Referencia una cadena de caracteres con una longitud máxima de 50 caracteres (bytes) para definir el nombre de archivo del anexo. Si en este parámetro se asigna una cadena vacía, el anexo de correo electrónico se enviará con el nombre de archivo "attachment.bin". Mediante el parámetro AttachmentName es posible asignar el nombre de anexo que aparece cuando el interlocutor recibe el correo electrónico. El SDT TMail_FileReference utiliza automáticamente el parámetro FileName para el parámetro AttachmentName. Si se utiliza el SDT TMail_FileReference, el parámetro AttachmentName no se aplica. Déjelo vacío. Si se introducen datos en el parámetro AttachmentName mientras se utiliza el SDT TMail_FileReference, la instrucción TMAIL_C genera un error. Consulte "Códigos de condición de error, parámetro SFB_STATUS del DB de instancia" para más información.
MAIL_ADDR_PARAM (Página 687)	Input	VARIANT	D	Dirección y parámetro de conexión del servidor de correo electrónico Para definir los parámetros de conexión, use el SDT TMail_V4, TMail_FQDN, TMail_V4_SEC o TMail_QDN_SEC (consulte la descripción del parámetro).
DONE (Página 697)	Output	BOOL	I, Q, M, D, L	Parámetro de estado • DONE = 0: orden no iniciada aún o en ejecución. • DONE = 1: orden ejecutada sin errores.

Parámetro	Declaración	Tipo de datos	Área de memoria	Descripción
BUSY (Página 697)	Output	BOOL	I, Q, M, D, L	Parámetro de estado • BUSY= 0: Se ha detenido el procesamiento de TMAIL_C. • BUSY = 1: la transmisión de correo electrónico no ha finalizado aún.
ERROR (Página 697)	Output	BOOL	I, Q, M, D, L	Parámetro de estado • ERROR = 0: no se ha producido ningún error. • ERROR = 1: se ha producido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS (Página 701)	Output	WORD	I, Q, M, D, L	Parámetro de estado Valor de retorno o información de error de la instrucción TMAIL_C (véase la descripción del parámetro).

Nota

Parámetros opcionales

La instrucción envía los parámetros opcionales CC, TEXT y ATTACHMENT solo si los parámetros correspondientes contienen una cadena de una longitud > 0.

Parámetro MAIL_ADDR_PARAM

En el parámetro MAIL_ADDR_PARAM se define la conexión para enviar el correo electrónico y guardar la dirección del servidor de correo electrónico y los detalles de inicio de sesión.

El tipo de datos de sistema (SDT) que se utiliza en el parámetro MAIL_ADDR_PARAM depende del formato en el que debe direccionarse el servidor de correo electrónico:

SDT	Descripción	Soporte de interfaz
TMail_V4	direccionamiento con dirección IP (IPv4)	CPU y CP
TMail_V6	Direccionamiento con dirección IP (IPv6)	CP
TMail_FQDN	Direccionamiento con Fully Qualified Domain Name (FQDN)	CP
TMail_V4_SEC	direccionamiento seguro con dirección IP (IPv4)	CPU y CP
TMail_V6_SEC	Direccionamiento seguro con dirección IP (IPv6)	CP
TMail_QDN_SEC	Direccionamiento seguro con Fully Qualified Domain Name (FQDN)	CPU y CP

TMail_V4: Direccionar el servidor de correo con dirección IP (IPv4)

Parámetro	Tipo de datos	Descripción
TMail_v4	Struct	
InterfaceId	LADDR	Identificador de hardware de la interfaz Ethernet
ID	CONN_OUC	ID de conexión
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#20 como tipo de conexión para IPv4.
ActiveEstablished	BOOL	Establecimiento activo/pasivo de la conexión. La CPU siempre es el cliente SMTP.
CertIndex	BYTE	=0: se ha usado SMTP (**S**imple **M**ail **T**ransfer **P**rotocol). Hay que utilizar SMTP si el correo electrónico se envía a través de la interfaz de una CPU S7-1200. ≠0: SMTPS utilizado para asegurar la conexión (interfaz de CP)
WatchDogTime	TIME	Watchdog de ejecución. Use este parámetro para definir el tiempo de ejecución máximo para la transmisión. Este valor determina el tiempo durante el cual la instrucción TMAIL_C puede permanecer en ejecución antes de alcanzar un timeout y de que se detenga la ejecución de la instrucción TMAIL_C. A partir de la versión V6.0 de TMAIL_C no se permite un valor cero de WatchDogTime, que indica a la instrucción TMAIL_C que desactive el temporizador durante la ejecución. Sigue siendo posible introducir un valor diferente de cero para WatchDogTime con el fin de que la ejecución de la instrucción TMAIL_C sea más determinista. Nota: si la conexión es lenta, esta tardará más en establecerse (aproximadamente un minuto). A la hora de especificar el parámetro WatchDogTime recuerde prever tiempo suficiente para que se establezca la conexión.
MailServerAddress	IP_v4	Dirección IP del servidor de correo. IPv4 en el formato siguiente: XXX.XXX.XXX.XXX (decimal). Ejemplo: 192.142.131.237
UserName	STRING[254]	Nombre de inicio de sesión del servidor de correo
PassWord	STRING[254]	Contraseña del servidor de correo electrónico
From	EMAIL_ADDR	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del emisor, incluido el símbolo @. Ejemplo: "myname@".
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado FQDN) del servidor de correo. Ejemplo: "mymailserver.com".

Si se utiliza la interfaz de la CPU S7-1200 con una versión de firmware V4.3 o anterior, hay que usar el SDT TMail_V4. En este caso, el correo electrónico solo se puede enviar con SMTP. Los mensajes de correo electrónico no son seguros.

TMail_V6: Direccionar el servidor de correo con dirección IP a IPv6

Parámetro	Tipo de datos	Descripción
TMail_V6	Struct	
InterfaceId	LADDR	Identificador de hardware de la interfaz
ID	CONN_OUC	ID de conexión
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#21 como tipo de conexión para IPv6.
ActiveEstablished	BOOL	Bit de estado. Se pone a "1" una vez que la conexión se ha establecido.
CertIndex	BYTE	=0: se ha usado SMTP (**S**imple **M**ail **T**ransfer **P**rotocol). Hay que utilizar SMTP si el correo electrónico se envía a través de la interfaz de una CPU S7-1500. ≠0: se ha usado SMTPS para asegurar la conexión antes de que esté establecida (con CP/CM). El parámetro CertIndex se utiliza para especificar el certificado que debe usarse (véase "Árbol del proyecto" > "Ajustes globales de seguridad" > "Administrador de certificados").
WatchDogTime	TIME	Watchdog de ejecución. Utilice este parámetro para definir el tiempo de ejecución máximo para la operación de transmisión. Nota: el establecimiento de la conexión puede requerir más tiempo (aprox. un minuto) si la conexión es lenta. A la hora de especificar el parámetro WATCH_DOG_TIME, recuerde prever tiempo suficiente para que se establezca la conexión. La conexión se deshace cuando transcurre el tiempo especificado.
MailServerAddress	IP_V6	Dirección IP del servidor de correo (IPv6) en el formato siguiente: XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX (hexadecimal). La dirección se divide en 8 bloques de 2 bytes cada uno (16 bytes en total). Ejemplo: 2001:db8:1f11:08d3:290:27ff:0370:2093
UserName	STRING[254]	Nombre de inicio de sesión en el servidor de correo
PassWord	STRING[254]	Contraseña del servidor de correo
From	EMAIL_ADDR	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del remitente, incluido el signo @. Ejemplo: "myname@"
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado como FQDN) del servidor de correo. Ejemplo: "mymailserver.com"

TMail_FQDN : Direccionar el servidor de correo con FQDN

Parámetro	Tipo de datos	Descripción
TMail_FQDN	Struct	
InterfaceId	LADDR	Identificador de hardware de la interfaz Ethernet
ID	CONN_OUC	ID de conexión
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#22 como tipo de conexión para FQDN.
ActiveEstablished	BOOL	Establecimiento activo/pasivo de la conexión. Un procesador de comunicaciones (CP) siempre es el cliente SMTP.
CertIndex	BYTE	=0: se ha usado SMTP (**S**imple **M**ail **T**ransfer **P**rotocol). ≠0: se ha usado SMTPS para asegurar la conexión antes de que esté establecida (interfaz del CP). CertIndex especifica el certificado que debe usarse.
WatchDogTime	TIME	Watchdog de ejecución. Use este parámetro para definir el tiempo de ejecución máximo para la transmisión. Nota: si la conexión es lenta, esta tardará más en establecerse (aproximadamente un minuto). A la hora de especificar el parámetro WatchDogTime recuerde prever tiempo suficiente para que se establezca la conexión.
MailServerAddress	STRING[254]	FQDN (**F**ully **Q**ualified **D**omain **N**ame) del servidor de correo. El servidor de correo se direcciona mediante el FQDN. Ejemplo: "www.mymailserver.com."
UserName	STRING[254]	Nombre de inicio de sesión del servidor de correo
PassWord	STRING[254]	Contraseña del servidor de correo electrónico
From	Struct	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del remitente, incluido el signo @. Ejemplo: "myname@".
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado FQDN) del servidor de correo. Ejemplo: "mymailserver.com".

TMail_V4_SEC: Direccionar el servidor de correo con dirección IP (IPv4)

Parámetro	Tipo de datos	Descripción
TMail_V4_SEC	Struct	
InterfaceId	LADDR	Identificador de hardware de la interfaz Ethernet. Rango de valores: • 0 (nuevo): el sistema operativo selecciona un puerto integrado adecuado por cuenta propia. • Identificador de hardware del puerto integrado que se utilizará.
ID	CONN_OUC	Referencia de la conexión. Rango de valores: • 0 (nuevo): el sistema operativo selecciona un ID de conexión libre del rango interno. • de 1 a 4095: ID de conexión que se utilizará
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#20 como tipo de conexión para IPv4.
ActiveEstablishment	BOOL	Establecimiento activo/pasivo de la conexión. La CPU siempre es el cliente SMTP.
WatchDogTime	TIME	Watchdog de ejecución. Utilice este parámetro para definir el tiempo de ejecución máximo para la operación de transmisión. Este valor determina el tiempo durante el cual la instrucción TMAIL_C puede permanecer en ejecución antes de alcanzar un timeout y de que se detenga la ejecución de la instrucción TMAIL_C. A partir de la versión V6.0 de TMAIL_C no se permite un valor cero de WatchDogTime, que indica a la instrucción TMAIL_C que desactive el temporizador durante la ejecución. Sigue siendo posible introducir un valor diferente de cero para WatchDogTime con el fin de que la ejecución de la instrucción TMAIL_C sea más determinista. Nota: el establecimiento de la conexión puede requerir más tiempo (aprox. un minuto) si la conexión es lenta. A la hora de especificar el parámetro WatchDogTime recuerde prever tiempo suficiente para que se establezca la conexión.
MailServerAddress	IP_V4	Dirección IP del servidor de correo intrínseco en formato IPv4: XXX.XXX.XXX.XXX (decimales) Ejemplo: 192.142.131.237
UserName	STRING[254]	Nombre de usuario Hay que usar el "nombre de usuario" para acceder a la carpeta de entrada del correo electrónico; esto sirve para identificarse como propietario de la carpeta de entrada ante el proveedor de correo electrónico.
PassWord	STRING[254]	Contraseña de usuario Hay que usar la "contraseña" para acceder a la carpeta de entrada del correo electrónico; esto sirve para identificarse como propietario de la carpeta de entrada ante el proveedor de correo electrónico.
From	EMAIL_ADDR	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com"

Parámetro		Tipo de datos	Descripción
	LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del remitente, incluido el signo @. Ejemplo: "myname@"
	FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado FQDN) del servidor de correo. Ejemplo: "mymailserver.com"
RemotePort		UINT	Puerto TCP del servidor de correo
ActivateSecureConn		BOOL	0: conexión SMTP (no segura). En este caso, los parámetros siguientes son irrelevantes. 1: conexión SMTP segura
ExtTLSCapabilities		BYTE	No se utiliza actualmente
TLSServerCertRef		UDINT	Referencia al certificado X.509 V3 (CA) del servidor de correo que el cliente TLS utiliza para validar la autenticación del servidor TLS

TMail_V6_SEC: Direccionamiento del servidor de correo electrónico a través de la dirección IP en formato IPv6

Parámetro	Tipo de datos	Descripción
TMail_V6_SEC	Struct	
InterfaceId	LADDR	Identificador de hardware de la interfaz Ethernet
ID	CONN_OUC	ID de conexión
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#21 como tipo de conexión para IPv6.
ActiveEstablishment	BOOL	Establecimiento activo/pasivo de la conexión. Dado que el CP siempre es el cliente SMTP, este parámetro debe ponerse a "1".
WatchDogTime	TIME	Watchdog de ejecución. Utilice este parámetro para definir el tiempo de ejecución máximo para la operación de transmisión. Nota: el establecimiento de la conexión puede requerir más tiempo (aprox. un minuto) si la conexión es lenta. A la hora de especificar el parámetro WatchDogTime, recuerde prever tiempo suficiente para que se establezca la conexión. La conexión se deshace cuando transcurre el tiempo especificado.
MailServerAddress	IP_V6	Dirección IP del servidor de correo en formato IPv6: XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX (hexadecimal). La dirección se divide en 8 bloques de 2 bytes cada uno (16 bytes en total). Ejemplo: 2001:db8:1f11:08d3:290:27ff:0370:2093
UserName	STRING[254]	Nombre de usuario. Es necesario si los usuarios desean acceder a su buzón de entrada del correo electrónico para identificarse a sí mismos como propietarios del buzón ante el proveedor de correo electrónico.
PassWord	STRING[254]	Contraseña de usuario. Es necesario si los usuarios desean acceder a su buzón de entrada del correo electrónico para identificarse a sí mismos como propietarios del buzón ante el proveedor de correo electrónico.
From	EMAIL_ADDR	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del remitente, incluido el signo @. Ejemplo: "myname@"

Parámetro		Tipo de datos	Descripción
	FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado como FQDN) del servidor de correo. Ejemplo: "mymailserver.com"
RemotePort		UINT	Puerto TCP del servidor de correo
ActivateSecureConn		BOOL	0: conexión SMTP (no segura). En este caso, los parámetros siguientes son irrelevantes. 1: conexión SMTP segura
ExtTLSCapabilities		BYTE	No se utiliza actualmente
TLSServerCertRef		UDINT	Referencia al certificado X.509 V3 (CA) del servidor de correo que el cliente TLS utiliza para validar la autenticación del servidor TLS.

TMail_QDN_SEC: Direccionar el servidor de correo con FQDN

Parámetro	Tipo de datos	Descripción
TMail_QDN_SEC	Struct	

Parámetro	Tipo de datos	Descripción
InterfaceId	LADDR	Identificador de hardware de la interfaz Ethernet. Rango de valores: • 0 (nuevo): el sistema operativo selecciona un puerto integrado adecuado por cuenta propia. • Identificador de hardware del puerto integrado que se utilizará.
ID	CONN_OUC	Referencia de la conexión. Rango de valores: • 0 (nuevo): El sistema operativo selecciona un ID de conexión libre del rango interno. • de 1 a 4095: el ID de conexión que se utilizará
ConnectionType	BYTE	Tipo de conexión. Seleccione 16#22 como tipo de conexión para FQDN.
ActiveEstablishment	BOOL	Establecimiento activo/pasivo de la conexión. La CPU siempre es el cliente SMTP.
WatchDogTime	TIME	Watchdog de ejecución. Utilice este parámetro para definir el tiempo de ejecución máximo para la operación de transmisión. Este valor determina el tiempo durante el cual la instrucción TMAIL_C puede permanecer en ejecución antes de alcanzar un timeout y de que se detenga la ejecución de la instrucción TMAIL_C. A partir de la versión V6.0 de TMAIL_C no se permite un valor cero de WatchDogTime, que indica a la instrucción TMAIL_C que desactive el temporizador durante la ejecución. Sigue siendo posible introducir un valor diferente de cero para WatchDogTime con el fin de que la ejecución de la instrucción TMAIL_C sea más determinista. Nota: el establecimiento de la conexión puede requerir más tiempo (aprox. un minuto) si la conexión es lenta. A la hora de especificar el parámetro WatchDogTime recuerde prever tiempo suficiente para que se establezca la conexión.
MailServerQDN	STRING[254]	FQDN (**F**ully **Q**ualified **D**omain **N**ame) del servidor de correo. El servidor de correo se direcciona mediante el nombre de dominio plenamente cualificado, que de terminar con "." Ejemplo: "www.mymailserver.com."
UserName	STRING[254]	Nombre de usuario Hay que usar el "nombre de usuario" para acceder a la carpeta de entrada del correo electrónico; esto sirve para identificarse como propietario de la carpeta de entrada ante el proveedor de correo electrónico.
PassWord	STRING[254]	Contraseña de usuario Hay que usar la "contraseña" para acceder a la carpeta de entrada del correo electrónico; esto sirve para identificarse como propietario de la carpeta de entrada ante el proveedor de correo electrónico.
From	EMAIL_ADDR	Dirección del remitente del correo electrónico, que se define mediante los dos parámetros STRING siguientes. Por ejemplo: "myname@mymailserver.com"

Parámetro		Tipo de datos	Descripción
	LocalPartPlusAtSign	STRING[64]	Parte local de la dirección del emisor, incluido el símbolo @. Ejemplo: "myname@"
	FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abreviado FQDN) del servidor de correo. Ejemplo: "mymailserver.com"
RemotePort		UINT	Puerto TCP del servidor de correo
ActivateSecureConn		BOOL	0: conexión SMTP (no segura). En este caso, los parámetros siguientes son irrelevantes. 1: conexión SMTP segura
ExtTLSCapabilities		BYTE	No se utiliza actualmente
TLSServerCertRef		UDINT	Referencia del certificado X.509 V3 (CA) del servidor de correo, que el cliente TLS utiliza para validar la autenticación del servidor TLS

Parámetros TO_S y CC

Para instrucciones TMAIL_C anteriores a la versión 6.0 y CPU S7-1200 con firmware anterior a V4.4 se aplican las reglas siguientes a la hora de introducir los parámetros TO_S y CC:

- Introduzca un espacio y un paréntesis angular de apertura "<" delante de cada dirección.
- Introduzca un paréntesis angular de cierre ">" detrás de cada dirección.
- Introduzca una coma entre las direcciones en TO y CC.

A continuación se muestran algunos ejemplos de estas cadenas de los parámetros TO_S y CC:

- <wenna@mydomain.com>, <ruby@mydomain.com>
- <admin@mydomain.com>, <judy@mydomain.com>

A partir de la versión 6.0 de la instrucción TMAIL_C y CPU S7-1200 con firmware V4.4 o posterior solo se aplica la regla siguiente a la hora de introducir los parámetros:

- Introduzca una coma o un punto y coma entre las direcciones en TO y CC.

A continuación se muestran algunos ejemplos de estas cadenas de los parámetros TO_S y CC:

- wenna@mydomain.com, ruby@mydomain.com
- admin@mydomain.com, judy@mydomain.com

Por motivos de tiempos de ejecución y espacio en memoria, la instrucción TMAIL_C no comprueba la sintaxis del parámetro TO_S o CC.

Parámetros DONE, BUSY y ERROR

Los parámetros de salida DONE, BUSY y ERROR se muestran solo durante un ciclo si el estado del parámetro de salida BUSY cambia de "1" a "0".

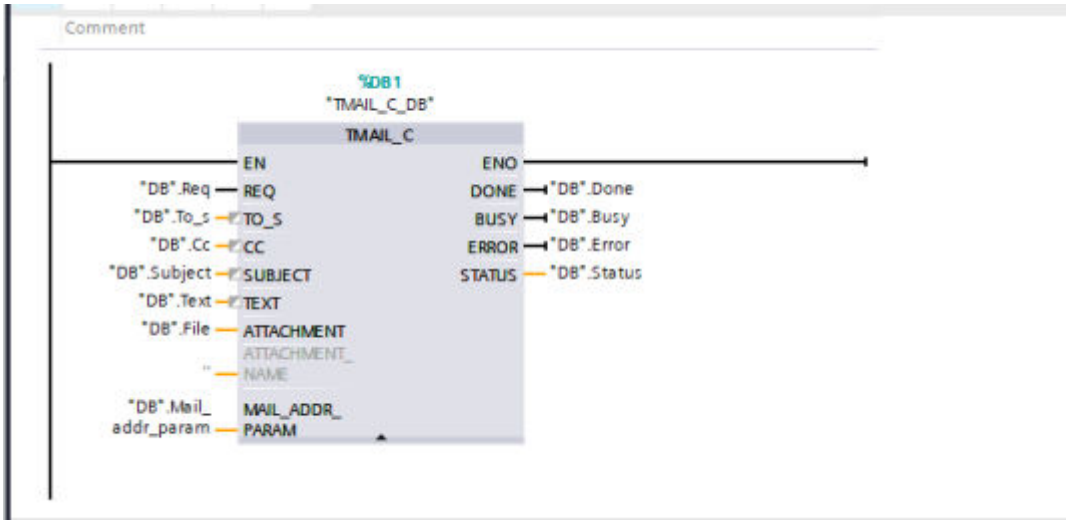
En la tabla siguiente se muestra la relación entre los parámetros DONE, BUSY y ERROR. Con esta tabla es posible determinar el estado actual de la instrucción TMAIL_C y saber cuándo ha finalizado el envío del correo electrónico.

DONE	BUSY	ERROR	Descripción
0	1	0	La tarea se está procesando.
1	0	0	Tarea finalizada correctamente.
0	0	1	La tarea ha finalizado con un error. La causa del error se indica en el parámetro STATUS (Página 701).
0	0	0	No se ha asignado ninguna orden (nueva) a la instrucción TMAIL_C.

Enviar registros de datos, recetas y archivos de usuario mediante anexos de correo electrónico

A partir de la versión V6.0 o posterior de TMAIL_C y la versión de firmware V4.4 o posterior de la CPU S7-1200 es posible agregar el SDT TMail_FileReference y acceder a él mediante el parámetro Attachment de la instrucción TMAIL_C. Seguidamente, es posible direccionar una ruta de archivo en la SIMATIC Memory Card (SMC). Si no hay ninguna Memory Card es posible acceder a los directorios de recetas y registros de datos en la memoria de carga interna del PLC.

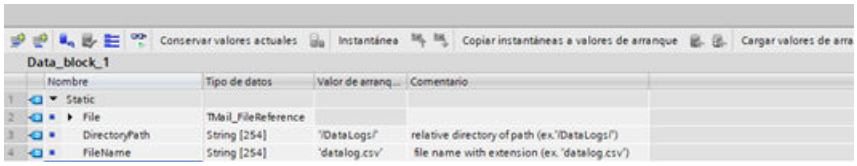
El SDT TMail_FileReference utiliza automáticamente el parámetro FileName para el parámetro AttachmentName.



SDT TMail_FileReference

El SDT TMail_FileReference consta de dos parámetros que son ambos cadenas SIMATIC:

- En el parámetro DirectoryPath se direcciona el directorio del archivo deseado.
- El parámetro FileName indica el nombre del archivo y la extensión (en su caso) del archivo al que debe accederse en el directorio, que se especifica en el parámetro anterior.



La instrucción TMAIL_C le restringe a los directorios DataLogs, Recipes o UserFiles al direccionar el parámetro DirectoryPath. También es posible direccionar subdirectorios dentro de estos directorios.

Más allá de la restricción anterior del directorio básico, la sección de reglas de direccionamiento de archivos prescribe las reglas que deben respetarse para direccionar los subdirectorios y los nombres de archivo.

El tamaño del archivo que puede enviarse no está limitado por la instrucción TMAIL_C. Recuerde esto al crear el programa.

Reglas de direccionamiento de archivos

Hay determinadas reglas que deben respetarse para direccionar correctamente un archivo utilizando el SDT TMail_FileReference con la instrucción TMAIL_C. Los siguientes subapartados describen los detalles de DirectoryPath y FileName, respectivamente. En general, lo siguiente es válido para ambos parámetros del SDT TMail_FileReference y, si no se siguen estas reglas, la instrucción TMAIL_C genera un estado de error:

- No es posible utilizar una cadena vacía como subdirectorio o nombre de archivo.
- No es posible utilizar caracteres de control ASCII (rango Hex: de 0x00 a 0x1F) en ninguna cadena del parámetro.
- No es posible utilizar los siguientes caracteres reservados en ninguna cadena del parámetro:
  - < (menor)
  - > (mayor)
  - : (dos puntos)
  - " (comillas dobles)
  - / (barra inclinada) (este carácter está permitido en DirectoryPath como separador)
  - \ (barra invertida)
  - | (barra recta)
  - ? (signo de interrogación)
  - * (asterisco)
- Los subdirectorios o nombres de archivo no pueden terminar con un espacio o punto.

DirectoryPath

Recuerde esto al introducir el directorio deseado en el parámetro DirectoryPath del SDT. El directorio raíz es deducido por la lógica de firmware del PLC y el usuario no tiene que conocerlo. Opcionalmente es posible introducir una barra inclinada al principio y al final (/). Si el usuario no introduce las barras, el firmware las agregará automáticamente a la ruta. Por consiguiente, son válidas todas las entradas siguientes para DirectoryPath:

- /DataLogs/
- /DataLogs
- DataLogs/
- DataLogs

También es posible acceder a una ruta más profunda que el directorio básico utilizando la forma siguiente: '/DataLogs/dir1/', siendo cada (/) un directorio nuevo. La profundidad máxima es de ocho incluyendo el directorio raíz.

Además de las reglas expuestas en el apartado Reglas de direccionamiento de archivos, recuerde que el uso de rutas relativas está estrictamente prohibido. Así, '/DataLogs/' es un nombre de ruta no válido. También está prohibido el uso de un punto en cualquier componente del subdirectorio para representar el directorio actual (por ejemplo, '/DataLogs/').

FileName

Recuerde las reglas indicadas en el apartado Reglas de direccionamiento de archivos cuando utilice el parámetro FileName del SDT. Asimismo, tenga en cuenta que el sistema operativo del PLC limita el nombre de archivo a menos de 60 caracteres. Si intenta direccionar un nombre de archivo de 60 caracteres o más, la instrucción TMAIL_C cancela la operación y genera un error.

Aparte de estas excepciones, es posible anexar cualquier archivo independientemente de su tamaño o extensión. El archivo direccionado puede incluir una extensión o no.

Códigos de condición de error

Parámetro STATUS

En la tabla siguiente se muestran los valores de retorno de TMAIL_C en el parámetro STATUS:

Valor de retorno STATUS* (W#16#...):	Explicación	Notas
0000	El procesamiento de TMAIL_C ha finalizado sin errores.	La finalización de TMAIL_C sin errores no significa que necesariamente que el correo electrónico enviado llegue a su destino. Si se introducen las direcciones de destinatario incorrectamente, no se generará un estado de error de la instrucción TMAIL_C. En ese caso, no se podrá garantizar que el correo electrónico se envíe a otros destinatarios, aunque estos se hayan introducido correctamente.
7001	TMAIL_C está activa (BUSY = 1).	Primera llamada: tarea disparada.
7002	TMAIL_C está activa (BUSY = 1).	Llamada intermedia: tarea ya activa.
8xxx	El procesamiento de TMAIL_C ha finalizado con un código de error de las instrucciones de comunicación llamadas internamente.	Para obtener información detallada, consulte las descripciones del parámetro STATUS para las instrucciones de comunicación TCON, TDISCON, TSEND y TRCV (Página 652).
8009	Error interno de función	Una función interna ha devuelto un error. Encontrará información detallada en el parámetro SFB_STATUS del DB de instancia. Los valores posibles se describen más adelante.
8010	Error durante el establecimiento de conexión	Encontrará más información sobre la evaluación en el parámetro SFB_STATUS del bloque de datos de instancia. El código de error que se muestra en el parámetro SFB_STATUS se explica en la descripción del parámetro STATUS para la instrucción TCON (Página 652).
8011	Error al transmitir datos	Encontrará más información sobre la evaluación en el parámetro SFB_STATUS del bloque de datos de instancia. El código de error que se muestra en el parámetro SFB_STATUS se explica en la descripción del parámetro STATUS para la instrucción TSEND (Página 652).
8012	Error al recibir datos	Encontrará más información sobre la evaluación en el parámetro SFB_STATUS del bloque de datos de instancia. El código de error que se muestra en el parámetro SFB_STATUS se explica en la descripción del parámetro STATUS para la instrucción TRCV (Página 652).

Valor de retorno STATUS* (W#16#...):	Explicación	Notas
8013	Error durante el establecimiento de conexión	Encontrará más información sobre la evaluación en el parámetro SFB_STATUS del bloque de datos de instancia. El código de error que se muestra en el parámetro SFB_STATUS se explica en la descripción del parámetro STATUS para las instrucciones TCON (Página 652) y TDISCON (Página 652).
8014	No es posible establecer una conexión.	Puede que haya introducido una dirección IP de servidor de correo electrónico (MailServerAddress (Página 687)) o un intervalo de tiempo demasiado corto (WatchDogTime (Página 687)) para establecer la conexión. Asimismo, es posible que la CPU no tenga conexión con la red o que su configuración sea incorrecta.
8015	Tipo de datos incorrecto para MAIL_ADDR_PARAM	Los únicos tipos de datos válidos son los tipos de datos de sistema (estructuras) Tmail_v4 y TMail_FQDN.
8016	Tipo de datos incorrecto para el parámetro ATTACHMENT	Los tipos de datos válidos se encuentran en la lista siguiente: • ArrayOfChar • ArrayOfByte • ArrayOfWord • ArrayOfDWord • String Nota: Los tipos de datos ArrayOfChar y String solo son válidos con la instrucción TMAIL_C en versión V5.0 o posterior.
8017	Longitud de datos incorrecta para el parámetro ATTACHMENT	La longitud de datos debe ser <= 65534 bytes.
82xx, 84xx, o 85xx	El mensaje de error se genera en el servidor de correo electrónico y corresponde, excepto en el caso del "8", al número de error del protocolo SMTP. En las siguientes líneas se enumeran varios códigos de error que se pueden generar.	Encontrará más información sobre el código de error SMTP y otros códigos de error del protocolo SMTP en Internet o en la documentación de errores del servidor de correo electrónico. Además, puede ver el mensaje de error más reciente del servidor de correo en su DB de instancia en el parámetro BUFFER1. Encontrará los últimos datos transmitidos por la instrucción TMAIL_C en DATEN en el DB de instancia.
8450	Acción no ejecutada: buzón de correo no disponible/no accesible.	Inténtelo más tarde.
8451	Acción cancelada: error de procesamiento local.	Inténtelo más tarde.

Valor de retorno STATUS* (W#16#...):	Explicación	Notas
8500	Error de sintaxis: error no reconocido. También incluye el error cuando una cadena de comandos es demasiado larga. Esto puede producirse cuando el servidor de correo electrónico no admite el proceso de autenticación LOGIN.	Compruebe los parámetros de TMAIL_C. Intente enviar un correo electrónico sin autenticación. Para ello, sustituya el contenido del parámetro UserName por un parámetro con una cadena vacía. Si no se especifica ningún nombre de usuario, no se usará el procedimiento de autenticación LOGIN.
8501	Error de sintaxis: entrada incorrecta en un parámetro.	Causa posible: dirección incorrecta en el parámetro TO_S o CC (consulte también: parámetros TO_S y CC (Página 697)).
8502	Comando desconocido o no implementado.	Verifique las entradas, especialmente el parámetro FROM. Puede estar incompleto y puede que haya olvidado "@" o bien "." (ver también: parámetros TO_S y CC (Página 697)).
8535	Autenticación SMTP incompleta	Posiblemente ha introducido un nombre de usuario incorrecto o una contraseña incorrecta.
8550	No se puede acceder al servidor de correo electrónico. No dispone de derechos de acceso.	Puede que haya introducido un nombre de usuario incorrecto o una contraseña incorrecta o que el servidor de correo electrónico no admita sus datos de inicio de sesión. Otra causa del error puede ser un error en el nombre de dominio después de "@" en el parámetro TO_S o CC (consulte también: parámetros TO_S y CC (Página 697)).
8552	Acción cancelada: se ha superado el tamaño de memoria asignado.	Inténtelo más tarde.
8554	Error en la transferencia	Inténtelo más tarde.
* Es posible visualizar códigos de error en el editor de programas como valores enteros o hexadecimales.		

Parámetro SFB_STATUS del DB de instancia

Para la versión de firmware V4.4 o posterior de la CPU S7-1200, con la versión V6.0 o posterior de la instrucción TMAIL_C, son posibles los siguientes valores de retorno en el parámetro SFB_STATUS de la instancia DB:

Valores de retorno en el parámetro SFB_STATUS de la instancia DB (W#16#...)	Explicación
8085	El ID de conexión (parámetro ID) ya está siendo utilizado por una conexión configurada.
8086	El parámetro ID está fuera del rango válido.
8087	Se ha alcanzado el número máximo de conexiones; no es posible establecer más conexiones.
8088 *	El archivo no existe o no está disponible actualmente.
8089 *	No es posible abrir el archivo porque el número de archivos abiertos simultáneamente excede el límite del sistema. El límite por sistema de archivos es de 26 en el S7-1200.

Valores de retorno en el parámetro SFB_STATUS de la instancia DB (W#16#...)	Explicación
808A *	DirectoryPath contiene un directorio distinto de DataLogs, Recipes o UserFiles, o bien uno de los subdirectorios direccionados infringe las reglas de direccionamiento mencionadas anteriormente. Encontrará más información en DirectoryPath (Página 698).
808B *	FileName contiene una secuencia de caracteres no válida o se ha dejado en blanco. Encontrará más información en FileName (Página 698).
808C *	El parámetro AttachmentName debe estar vacío cuando se direcciona una ruta de archivo como anexo.
8092	Los parámetros TO_S y CC están vacíos o el subparámetro De está vacío o incompleto.
8093	El parámetro MAIL_ADDR_PARAM necesita que la conexión pase a ser una conexión segura, pero el servidor de correo no soporta el comando STARTTLS.
8095	Respuesta no válida del servidor de correo. Es posible que el servidor de correo no se conforme a RFC.
809A	La estructura del SDT en el parámetro MAIL_ADDR_PARAM no se soporta en una interfaz integrada.
809B	ID de interfaz no válido en el parámetro MAIL_ADDR_PARAM del SDT
80A1	La conexión especificada o el puerto remoto ya se está utilizando.
80A3	el ID es utilizado por una conexión que ha creado el programa de usuario.
80A4	La dirección IP del punto final remoto de la conexión no es válida o corresponde a la dirección IP del interlocutor local.
80A7	Error de comunicación: ha ejecutado TDISCON antes de que finalizara TMAIL_C.
80B7	El puerto remoto es 0 o la dirección IP del punto final del interlocutor se ha puesto a 0.0.0.0.
80C3	Escasez de recursos en la CPU
80C4	Error de comunicación temporal: • La conexión no se puede establecer en estos momentos. • La conexión no puede establecerse porque los cortafuegos de la ruta de conexión no están abiertos para los puertos necesarios. • La interfaz está recibiendo nuevos parámetros. • La instrucción TDISCON está eliminando la conexión configurada.
80C5	El servidor de correo se niega a establecer la conexión, la ha cancelado o la está finalizando activamente.
80C6	No es posible acceder al interlocutor de la conexión (error de red).
80C7	Tiempo de ejecución excedido.
80C8	Se está intentando restablecer una conexión existente.

Valores de retorno en el parámetro SFB_STATUS de la instancia DB (W#16#...)	Explicación
80C9	Error al validar el interlocutor de la conexión. El servidor de correo no se corresponde con el interlocutor definido en el parámetro MailServerAddress.
80CE	La dirección IP de la interfaz local es 0.0.0.0.
80D0	El parámetro MailServerAddress contiene una cadena vacía porque se ha intentado utilizar DNS.
80D1	El parámetro MailServerAddress no es un nombre de dominio plenamente cualificado. Es posible que falte el punto al final.
80D2	No se ha configurado una dirección de servidor DNS.
80D3	No se ha podido resolver el Fully Qualified Domain Name (FQDN). Causas posibles: • No es posible acceder al servidor DNS (por ejemplo, el servidor DNS se ha apagado o no se puede acceder al puerto remoto). • Se ha producido un error durante la comunicación con el servidor DNS. • El servidor DNS ha devuelto una respuesta DNS válida; sin embargo, la respuesta no incluía ninguna dirección IPv4.
80E0	La comunicación con el servidor de correo ha fallado debido a un mensaje con errores. Causas posibles: • Código de autenticación no válido del mensaje • Error al descodificar el mensaje • Error al descomprimir un mensaje • Desbordamiento de la capacidad interna
80E1	Error durante el handshake. Causas posibles: • Cancelación por parte del usuario • Seguridad insuficiente • La negociación renovada no se soporta • La versión de SSL/TLS no se soporta • Error de descifrado
80E2	Certificado no soportado / no válido Causa posible: La hora del módulo en cuestión no está ajustada o el módulo no está sincronizado. Ejemplo: el ajuste predeterminado para la fecha del módulo es 1/1/2012 y no se ha ajustado la fecha durante la puesta en marcha. El período de validez del certificado empieza el 20 de agosto de 2016 y finaliza el 20 de agosto de 2024. En este caso, la fecha del módulo está fuera del período de validez del certificado, por lo que este no es válido para el módulo.
80E3	El certificado del servidor de correo ha sido rechazado.
80E4	No se ha encontrado ninguna entidad de certificación válida para el certificado del servidor de correo.
80E5	El certificado del servidor de correo ha caducado.

Valores de retorno en el parámetro SFB_STATUS de la instancia DB (W#16#...)	Explicación
80E6	Errores de integridad en el protocolo de seguridad para la capa de transporte
80E7	Extensión no soportada en el certificado del servidor de correo
80E9	No se soporta el servidor TLS sin certificado de servidor

* Estos códigos de error se agregan a la instrucción TMAIL_C con el fin de ayudar a diagnosticar un direccionamiento incorrecto de la ruta de archivo.

11.5.8.17 UDP

UDP es un protocolo estándar descrito por RFC 768: User Datagram Protocol. UDP ofrece un mecanismo que le permite a una aplicación enviar un datagrama a otra; no obstante, el suministro de datos no está garantizado. Este protocolo tiene las características siguientes:

- Protocolo de comunicación rápido
- Adecuado para cantidades de datos pequeñas y medianas (hasta 1472 bytes)
- UDP es un protocolo de control de transporte más sencillo que TCP, con una capa delgada que tiene un nivel bajo de overheads
- Puede utilizarse muy flexiblemente con distintos sistemas de terceros
- Apto para routing
- Utiliza números de puerto para direccionar los datagramas
- Los avisos no se acusan: esta aplicación se requiere para asumir la responsabilidad en cuanto a la recuperación de errores y a la seguridad
- Es necesario programar la gestión de datos debido a la interfaz de programación SEND / RECEIVE

UDP soporta la comunicación Broadcast. Para utilizar Broadcast es necesario configurar la parte de la dirección IP correspondiente a la configuración de ADDR. Ejemplo: una CPU que tenga una dirección IP 192.168.2.10 y una máscara de subred 255.255.255.0 utilizaría una dirección Broadcast de 192.168.2.255.

11.5.8.18 TUSEND y TURCV

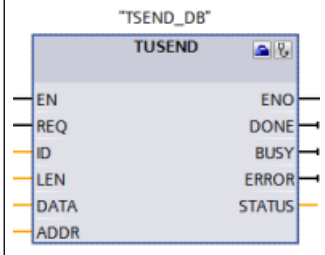
Las instrucciones siguientes controlan el proceso de comunicación UDP:

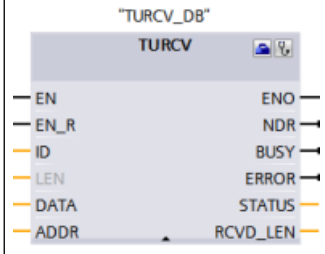
- TCON establece la comunicación entre el PC cliente y servidor (CPU).
- TUSEND y TURCV permiten transmitir y recibir datos.
- TDISCON desconecta la comunicación entre el cliente y servidor.

Consulte TCON, TDISCON, TSEND y TRCV (Página 652) en la sección "TCP e ISO on TCP" para obtener más información sobre las instrucciones de comunicación TCON y TDISCON.

Tabla 11-47 Instrucciones TUSEND y TURCV

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción TUSEND envía datos a través de UDP al interlocutor remoto especificado en el parámetro ADDR.
		Para iniciar la tarea de transmisión de datos llame la instrucción TUSEND con 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_);`
		La instrucción TURCV recibe datos a través de UDP. El parámetro ADDR muestra la dirección del emisor. Tras ejecutar correctamente TURCV, el parámetro ADDR contiene la dirección del interlocutor remoto (emisor).
		TURCV no soporta el modo ad hoc.
		Para iniciar la tarea de recepción de datos llame la instrucción TURCV con EN_R = 1.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

TCON, TDISCON, TUSEND y TURCV funcionan de forma asíncrona, con lo que el procesamiento de la tarea abarca varias ejecuciones de la instrucción.

Tabla 11-48 Tipos de datos TUSEND y TURCV para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ (TUSEND)	IN	Bool	Inicia la tarea de transmisión cuando se produce un flanco ascendente. Los datos se transfieren desde el área que indican los parámetros DATA y LEN.
EN_R (TURCV)	IN	Bool	• 0: La CPU no está habilitada para recibir. • 1: Habilita la CPU para recibir. La instrucción TURCV está lista para recibir y la tarea de recepción se procesa.
ID	IN	Word	Referencia a la conexión asociada entre el programa de usuario y el nivel de comunicación del sistema operativo. La ID tiene que ser idéntica al parámetro asociado ID en la descripción de la conexión local. Rango de valores: W#16#0001 a W#16#0FFF.
LEN	IN	UDInt	Número de bytes que deben enviarse (TUSEND) o recibirse (TURCV): • Predeterminado = 0. El parámetro DATA determina la longitud de los datos que deben enviarse o recibirse. • En otro caso, rango de valores: 1 a 1472

Parámetro y tipo		Tipo de datos	Descripción
DONE (TUSEND)	IN	Bool	Parámetro de estado DONE (TUSEND): • 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
NDR (TURCV)	OUT	Bool	Parámetro de estado NDR (TURCV): • 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada correctamente.
BUSY	OUT	Bool	• 1: Tarea no finalizada aún. No se puede iniciar una tarea nueva. • 0: Tarea finalizada.
ERROR	OUT	Bool	Parámetros de estado con los valores siguientes: • 0: No hay error • 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	OUT	Word	Información de estado, incluida la información de error. (Consulte los códigos de condición de error y estado en la tabla siguiente.)
RCVD_LEN	OUT	UDInt	Número de bytes recibido (TURCV).
DATA	IN_OUT	Variant	Dirección del área de emisión (TUSEND) o del área de recepción (TURCV): • La memoria imagen de proceso de las entradas • La memoria imagen de proceso de las salidas • Un bit de marcas • Un bloque de datos
ADDR	IN_OUT	Variant	Puntero a la dirección del receptor (para TUSEND) o del emisor (para TURCV) (por ejemplo, P#DB100.DBX0.0 byte 8). El puntero puede apuntar a cualquier área de memoria. Se requiere una estructura de 8 bytes como se indica a continuación: • Los primeros 4 bytes contienen la dirección IP remota. • Los 2 bytes siguientes especifican el número de puerto remoto. • Los últimos 2 bytes están reservados.

El estado de la tarea se indica en los parámetros de salida BUSY y STATUS. STATUS equivale al parámetro de salida RET_VAL de las instrucciones que operan de forma asíncrona.

La tabla siguiente muestra la relación entre los parámetros BUSY, DONE (TUSEND), NDR (TURCV) y ERROR. Utilizando esta tabla se puede determinar el estado actual de la instrucción (TUSEND o TURCV) o bien el instante en que finaliza el proceso de envío (transmisión) / recepción.

Tabla 11-49 Estado de los parámetros BUSY, DONE (TUSEND) / NDR (TURCV) y ERROR

BUSY	DONE / NDR	ERROR	Descripción
TRUE	Irrelevante	Irrelevante	La tarea se está procesando.
FALSE	TRUE	FALSE	La tarea se ha ejecutado correctamente.

BUSY	DONE / NDR	ERROR	Descripción
FALSE	FALSE	TRUE	La tarea se ha finalizado con un error. La causa del error se indica en el parámetro STATUS.
FALSE	FALSE	FALSE	A la instrucción no se le ha asignado una tarea (nueva).

[1] Debido al funcionamiento asíncrono de las instrucciones: Para TUSEND, es preciso conservar la coherencia de los datos en el área de emisión hasta que el parámetro DONE o ERROR adopta el valor TRUE. Para TURCV, los datos en el área de recepción solo son coherentes si el parámetro NDR adopta el valor TRUE.

Tabla 11-50 Códigos de condición TUSEND y TURCV para ERROR y STATUS

ERROR	STATUS	Descripción
0	0000	• Tarea de transmisión finalizada sin errores (TUSEND). • Se han aceptado datos nuevos. La longitud actual de los datos recibidos se muestra en RCVD_LEN (TURCV).
0	7000	• No se está procesando ninguna tarea (TUSEND) • El bloque no está listo para la recepción (TURCV)
0	7001	• Inicio del procesamiento de la tarea, enviando los datos (TUSEND): Durante este procesamiento, el sistema operativo accede a los datos del área de emisión DATA. • El bloque está listo para la recepción, se ha activado la tarea de recepción (TURCV).
0	7002	• Ejecución intermedia de la instrucción (REQ es irrelevante), procesando la tarea (TUSEND): Durante este procesamiento, el sistema operativo accede a los datos del área de emisión DATA. • Ejecución intermedia de la instrucción, procesando la tarea: Durante este procesamiento, la instrucción TURCV escribe datos en el área de recepción. Por este motivo, un error puede generar datos incoherentes en el área de recepción.
1	8085	El parámetro LEN excede el valor máximo admisible, tiene el valor 0 (TUSEND) o bien se ha cambiado el valor del parámetro LEN o DATA desde la primera ejecución de la instrucción (TURCV).
1	8086	El parámetro ID está fuera del rango de direcciones admisible.
1	8088	• El parámetro LEN excede el área de memoria (TUSEND) o de recepción (TURCV) especificada en DATA. • El área de recepción es demasiado pequeña (TURCV).
1	8089	El parámetro ADDR no apunta a un bloque de datos.
1	80A1	Error de comunicación: • Aún no se ha establecido la conexión especificada entre el programa de usuario y la capa de comunicación del sistema operativo. • Se está deshaciendo la conexión especificada entre el programa de usuario y la capa de comunicación del sistema operativo. La transmisión (TUSEND) o la tarea de recepción (TURCV) no es posible a través de esta conexión. • La interfaz se está reinicializando.
1	80A4	La dirección IP del punto final remoto de la conexión no es válida; puede ser que concuerde con la dirección IP local (TUSEND).
1	80B3	• El protocolo definido (parámetro connection_type en la descripción de la conexión) no es UDP. Utilice la instrucción TSEND o TRCV. • Parámetro ADDR: Ajustes no válidos para el número de puerto (TUSEND)

ERROR	STATUS	Descripción
1	80C3	• Ya se está procesando un bloque con esta ID en una clase de prioridad diferente. • Falta interna de recursos
1	80C4	Error de comunicación temporal: • La conexión entre el programa de usuario y la capa de comunicación del sistema operativo no se puede establecer en estos momentos (TUSEND). • La interfaz está recibiendo nuevos parámetros (TUSEND). • La conexión se está reiniciando (TURCV).

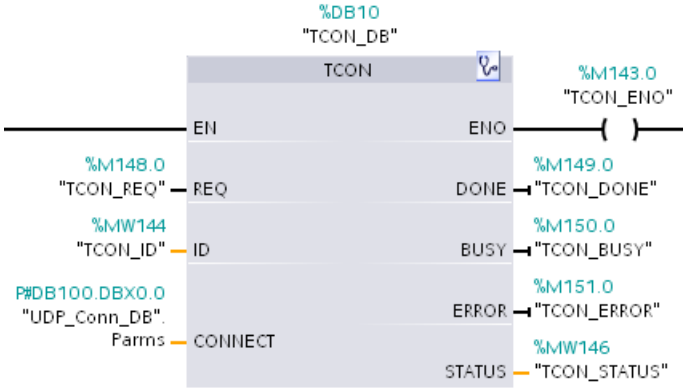
Protocolos de conexión Ethernet

Toda CPU incorpora un puerto PROFINET que soporta la comunicación PROFINET estándar. Las instrucciones TUSEND y TURCV soportan el protocolo Ethernet para UDP.

Encontrará más información en "Configurar la vía de conexión local/interlocutor" (Página 596), en el capítulo "Configuración de dispositivos".

Operaciones

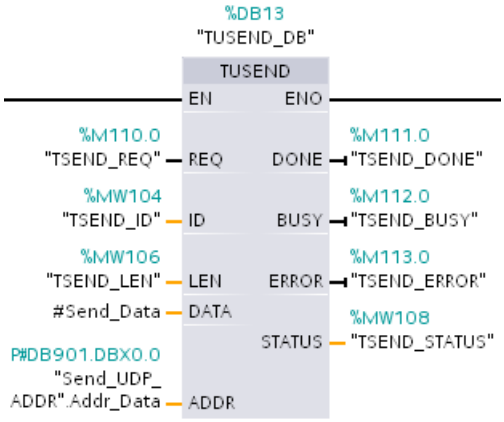
Los dos interlocutores son pasivos en la comunicación UDP. En las figuras siguientes se muestran los valores de arranque típicos de los parámetros para el tipo de datos "TCON_Param". Los números de puerto (LOCAL_TSAP_ID) están escritos en un formato de 2 bytes. Se permiten todos los puertos excepto 161, 34962, 34963 y 34964.



UDP_Conn_DB

	Nombre	Tipo de datos	Offset	Valor de arranque	Comentario
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_LEN	USInt	7.0	2	byte length of local TSAP id/port number
9	REM_SUBNET_ID_LEN	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_ID[1]	Byte		B#16#07	
15	LOCAL_TSAP_ID[2]	Byte		B#16#D0	

La instrucción TUSEND envía datos a través de UDP al interlocutor remoto especificado en el tipo de datos "TADDR_Param". La instrucción TURCV recibe datos a través de UDP. Tras ejecutar correctamente la instrucción TURCV, el tipo de datos "TADDR_Param" muestra la dirección del interlocutor remoto (emisor), como se muestra en las figuras que aparecen a continuación.



Send_UDP_ADDR

	Nome	Tipo di dati	Offset	Valore di avvio	Commento
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		0	
5	REM_IP_ADDR[2]	USint		0	
6	REM_IP_ADDR[3]	USint		0	
7	REM_IP_ADDR[4]	USint		0	
8	REM_PORT_NR	UInt	4.0	0	remote port number
9	RESERVED	Word	6.0	0	ununsed; has to be 0

11.5.8.19 T_CONFIG

La instrucción T_CONFIG puede modificar la dirección Ethernet, el nombre de dispositivo PROFINET o las direcciones IP de los servidores NTP para la sincronización horaria desde el programa de usuario. Las funciones siguientes pueden ajustarse de forma permanente o temporal:

- Dirección IP
- Máscara de subred
- Dirección del router
- Nombre de la estación
- Direcciones IP de un máximo de cuatro servidores NTP

Nota

El botón de opción "Permitir ajustar la dirección IP directamente en el dispositivo" (Página 719), situado en la página "Dirección Ethernet" de las "Propiedades" de la CPU, permite cambiar la dirección IP online o mediante la instrucción "T_CONFIG" después de descargar el programa.

El botón de opción "Permitir ajustar el nombre de dispositivo PROFINET directamente en el dispositivo" (Página 721), situado en la página "Dirección Ethernet" de las "Propiedades" de la CPU, permite cambiar el nombre del dispositivo PROFINET online o mediante la instrucción "T_CONFIG" después de descargar el programa.

El cuadro "Activar sincronización horaria vía servidor NTP" (Página 722), situado en la página "Sincronización horaria" de las "Propiedades" de la CPU, permite cambiar las direcciones IP de un máximo de cuatro servidores NTP.

Nota

No es posible ejecutar más de una instrucción T_CONFIG al mismo tiempo.

Nota

Los cambios en la dirección IP o el nombre de estación de la CPU pueden ser temporales o permanentes. Los cambios en las direcciones IP de los servidores NTP solo son temporales:

- Un cambio permanente indica que el cambio es remanente, es decir, que se conservará aunque se produzca un corte de la alimentación eléctrica.
- Un cambio temporal indica que el cambio es volátil y que se recuperará el valor original tras un corte de alimentación.

Tabla 11-51 Instrucción T_CONFIG

KOP / FUP	SCL	Descripción

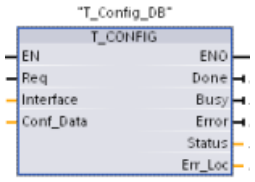
	`"T_CONFIG_DB"(`
	`Req:=_bool_in_,`
	`Interface:=_uint_in_,`
	`Conf_Data:=_variant_in_,`
	`Done=>_bool_out_,`
	`Busy=>_bool_out_,`
	`Error=>_bool_out_,`
	`Status=>_dword_out_,`
	`Err_Loc=>_dword_out_);`
		La instrucción T_CONFIG se puede usar para modificar los parámetros de configuración de IP desde el programa de usuario.
		T_CONFIG funciona de forma asíncrona. La ejecución abarca varias llamadas.

Tabla 11-52 Tipos de datos T_CONFIG para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	Input	Bool	Inicia la instrucción con un flanco ascendente.
INTERFACE	Input	HW_Interface	ID de la interfaz de red
CONF_DATA	Input	Variant	Referencia a la estructura de los datos de configuración; CONF_DATA se define mediante una estructura que contiene un máximo de cuatro tipos de datos de sistema (SDT).
DONE	Output	Bool	• 0: Tarea no iniciada aún o en proceso. • 1: Tarea finalizada sin errores.
BUSY	Output	Bool	• 0: Tarea finalizada. • 1: La tarea no ha finalizado aún. No se puede iniciar una tarea nueva.
ERROR	Output	Bool	Parámetros de estado con los valores siguientes: • 0: No hay error • 1: Ha ocurrido un error durante el procesamiento. STATUS proporciona información detallada sobre el tipo de error.
STATUS	Output	DWord	Información de estado, incluida la información de error. (Consulte los códigos de condición de error y estado en la tabla siguiente.)
ERR_LOC	Output	DWord	Ubicación del fallo (ubicación del campo ID y los subcampos dentro de la estructura CONF_DATA)

La información de la configuración de IP se deposita en el bloque de datos CONF_DATA, junto con un puntero de Variant en el parámetro CONF_DATA, del que se hace referencia más arriba. La ejecución correcta de la instrucción T_CONFIG finaliza con la entrega de los datos de configuración IP a la interfaz de red.

Los avisos de estado y los mensajes de error de la instrucción "T_CONFIG" se visualizan en los parámetros STATUS y ERR_LOC:

- La causa del error se muestra en el parámetro STATUS.
- La ubicación del error ocurrido se muestra en el parámetro ERR_LOC. Aquí están disponibles las siguientes opciones:
  - 16#0000_0000: no hay error o hay error al llamar la instrucción (p. ej. errores cuando se asignan parámetros a la instrucción o en la comunicación con la interfaz PROFINET).
  - 16#0001_0000: error con los datos de configuración en los parámetros del tipo de datos de sistema IF_CONF_HEADER.
  - 16#0001_000x: error con los datos de configuración en los parámetros del tipo de datos de sistema IF_CONF_V4 o IF_CONF_NOS o IF_CONF_NTP (la x indica la posición del sub-bloque defectuoso en la estructura de T_CONFIG. Si, por ejemplo, la estructura de T_CONFIG contiene un sub-bloque para la dirección IP y otro para el nombre de la estación y el error se encuentra en el sub-bloque para el nombre de la estación, ERR_LOC tiene el valor 0001_0002.)

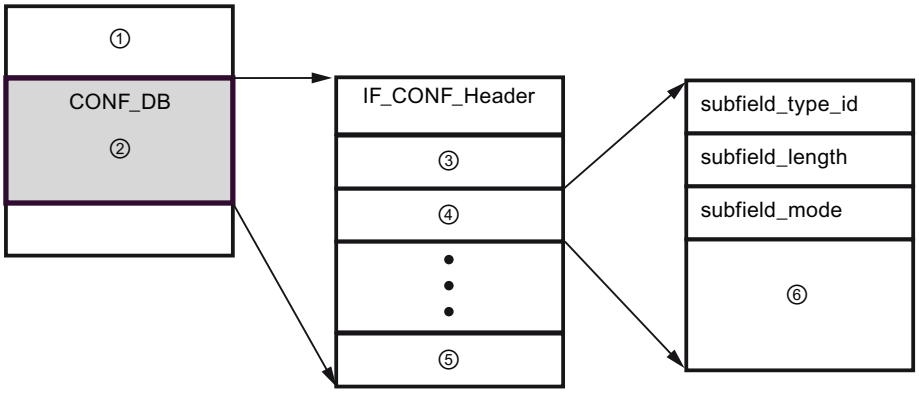
La tabla siguiente muestra los valores posibles de los parámetros STATUS y ERR_LOC:

STATUS*	ERR_LOC*	Explicación
0000_0000	0000_0000	Procesamiento de la orden finalizado sin errores.
0070_0000	0000_0000	No se está procesando ninguna tarea.
0070_0100	0000_0000	Inicio del procesamiento de la orden.
0070_0200	0000_0000	Llamada intermedia (REQ irrelevante).
C08x_yy00	0000_0000	Información general del error.
C080_8000	0000_0000	Error al llamar la instrucción: la ID de hardware en el parámetro Interface no es válida.
C080_8100	0000_0000	Error al llamar la instrucción: la ID de hardware en el parámetro Interface no direcciona una interfaz PROFINET.
C080_8700	0000_0000	Error al llamar la instrucción: longitud incorrecta del bloque de datos en el parámetro CONF_DATA.
C080_8800	0001_0000	Error en el tipo de datos de sistema IF_CONF_HEADER: el parámetro FieldType tiene un valor no válido. Utilice el valor "0" para FieldType.
C080_8900	0001_0000	Error en el tipo de datos de sistema IF_CONF_HEADER: el parámetro FieldId tiene un valor no válido o se ha utilizado más de una vez. Utilice el valor "0" para FieldId.
C080_8A00	0001_0000	Error en el tipo de datos de sistema IF_CONF_HEADER: número incorrecto en el parámetro SubfieldCount. Introduzca el número correcto de los tipos de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP que se utilizan.
C080_8B00	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: el parámetro Id tiene un valor no válido. Utilice "30" para IF_CONF_V4, "40" para IF_CONF_NOS y "17" para IF_CONF_NTP.
C080_8C00	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: se ha utilizado un tipo de datos de sistema incorrecto, su orden es erróneo o se ha utilizado varias veces.
C080_8D00	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: el parámetro Length tiene un valor incorrecto o no válido.

STATUS*	ERR_LOC*	Explicación
C080_8E00	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: el parámetro Mode tiene un valor incorrecto o no válido. • Con IF_CONF_V4 y IF_CONF_NOS solo se permiten los valores "1" (permanente) o "2" (temporal). • Con IF_CONF_NTP solo se permite el valor "2" (temporal).
C080_9000	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: No es posible aplicar los datos de configuración. Causa posible: • Con IF_CONF_V4: en la configuración hardware no se ha seleccionado el ajuste "Ajustar la dirección IP en el dispositivo". • Con IF_CONF_NOS: en la configuración hardware no se ha seleccionado el ajuste "Ajustar el nombre de dispositivo PROFINET en el dispositivo". • Con IF_CONF_NTP: en la configuración hardware no se ha seleccionado el ajuste "Activar sincronización horaria vía servidor NTP" y no se ha ajustado ninguna dirección IP para los servidores NTP.
C080_9400	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: un valor de parámetro no está definido o no es válido.
C080_9500	0001_000x	Error en el tipo de datos de sistema IF_CONF_V4, IF_CONF_NOS o IF_CONF_NTP: los valores de dos parámetros son incoherentes.
C080_C200	0000_0000	Error al llamar la instrucción: no es posible transferir los datos de configuración. Causa posible: la interfaz PROFINET no está accesible.
C080_C300	0000_0000	Error al llamar la instrucción: recursos insuficientes (p. ej. llamada múltiple de "T_CONFIG" con diferentes parámetros).
C080_C400	0000_0000	Error al llamar la instrucción: error de comunicación temporal. Indicación de tiempo para cambiar al horario de verano.
C080_D200	0000_0000	Error al llamar la instrucción: llamada no posible. La interfaz PROFINET seleccionada no soporta esta instrucción.

Bloque de datos CONF_DATA

El diagrama siguiente muestra cómo se guardan en el DB de configuración los datos de configuración que deben transferirse.



① DB de configuración
② Datos de configuración
③ Subcampo 1
④ Subcampo 2
⑤ Subcampo *n*
⑥ Parámetros específicos del subcampo

Los datos de configuración del CONF_DB se componen de un campo que contiene un encabezado (IF_CONF_Header) y varios subcampos. IF_CONF_Header ofrece los siguientes elementos:

- field_type_id (tipo de datos UInt): Cero
- field_id (tipo de datos UInt): Cero
- subfield_cnt (tipo de datos UInt): Número de subcampos

Cada subcampo, a su vez, consta de un encabezado (subfield_type_id, subfield_length, subfield_mode) y de los parámetros específicos del subcampo. Cada subcampo debe estar conformado por un número de bytes par. subfield_mode puede tener un valor 1 o 2. Consulte las tablas de más abajo.

Nota

Actualmente solo se permite un campo (IF_CONF_Header). Sus parámetros field_type_id y field_id deben tener el valor cero. Otros campos con diferentes valores para field_type_id y field_id están sujetos a futuras extensiones.

Tabla 11-53 Subcampos soportados

subfield_type_id	Tipo de datos	Significado
30	IF_CONF_V4	Parámetros de IP: Dirección IP, máscara de subred, dirección del router
40	IF_CONF_NOS	Nombre de dispositivo PROFINET IO (Name of station)
17	IF_CONF_NTP	Network Time Protocol (NTP)

Tabla 11-54 Elementos del tipo de datos IF_CONF_V4

Nombre	Tipo de datos	Valor de arranque	Descripción
Id	UInt	30	subfield_type_id
Length	UInt	18	subfield_length
Mode	UInt	0	subfield_mode (1: permanente o 2: temporal)
InterfaceAddress	IP_V4	-	Dirección de interfaz
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Dirección IP high byte: 200
ADDR[2]	Byte	0	Dirección IP high byte: 12
ADDR[3]	Byte	0	Dirección IP low byte: 1
ADDR[4]	Byte	0	Dirección IP low byte: 144
SubnetMask	IP_V4	-	Máscara de subred
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Máscara de subred high byte: 255
ADDR[2]	Byte	0	Máscara de subred high byte: 255
ADDR[3]	Byte	0	Máscara de subred low byte: 255
ADDR[4]	Byte	0	Máscara de subred low byte: 0
DefaultRouter	IP_V4	-	Router predeterminado
ADDR	Array [1..4] of Byte		
ADDR[1]	Byte	0	Router high byte: 200
ADDR[2]	Byte	0	Router high byte: 12
ADDR[3]	Byte	0	Router low byte: 1
ADDR[4]	Byte	0	Router low byte: 1

Tabla 11-55 Elementos del tipo de datos IF_CONF_NOS

Nombre	Tipo de datos	Valor de arranque	Descripción
Id	UInt	40	subfield_type_id
Length	UInt	246	subfield_length
Mode	UInt	0	subfield_mode (1: permanente o 2: temporal)
NOS (Name of station)	Array[1..240] of Byte	0	Nombre de la estación: Hay que ocupar el ARRAY desde el primer byte. Si el ARRAY es más largo que el nombre que se le va a asignar a la estación, es necesario introducir un byte cero detrás del nombre real de la estación (de conformidad con IEC 61158-6-10). En caso contrario, NOS se rechaza y la instrucción "T_CONFIG (Página 712)" introduce el código de error DW#16#C0809400 en STATUS. Si ocupa el primer byte con cero se borra el nombre de la estación.

El nombre de la estación está sujeto a las siguientes limitaciones:

- Un componente dentro del nombre de la estación, p. ej. una cadena de caracteres entre dos puntos, no debe exceder 63 caracteres.
- No se admiten caracteres especiales, como diéresis, paréntesis, guión bajo, barra, espacio, etc. El único carácter especial permitido es el guión.

- El nombre de la estación no debe comenzar ni terminar con el carácter especial "-".
- El nombre de la estación no debe comenzar con un número.
- Para el nombre de la estación no está permitido el formato n.n.n.n (n = 0, ... 999).
- El nombre de la estación no debe comenzar con la cadena "port-xyz" o "port-xyz-abcde" (a, b, c, d, e, x, y, z = 0, ... 9).

Nota

También es posible crear un ARRAY "NOS" que tenga menos de 240 bytes, pero no menos de 2 bytes. En este caso, deberá ajustarse la variable "Length" (longitud del subcampo) en correspondencia.

Tabla 11-56 Elementos del tipo de datos IF_CONF_NTP

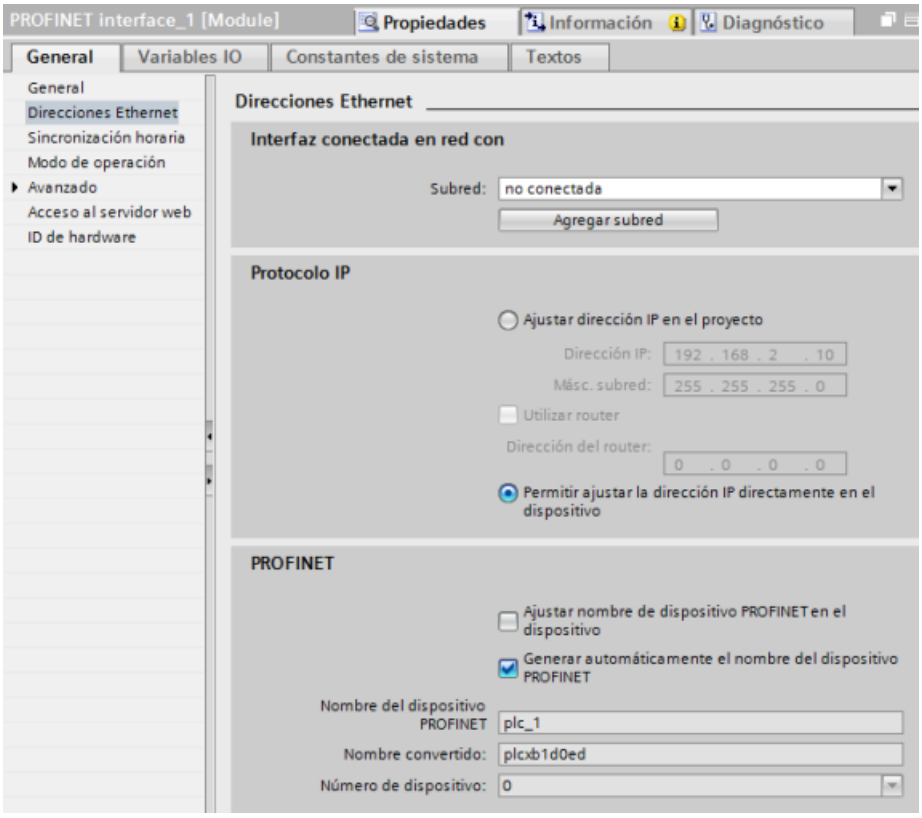
Nombre	Tipo de datos	Valor inicial	Descripción
Id	UInt	17	subfield_type_id
Length	UInt	22	subfield_length
Mode	UInt	0	subfield_mode (2: temporal)
NTP_IP	Array[1...4] of IP_V4	-	Direcciones IP de servidores NTP
NTP_IP[1]	IP_V4		Direcciones IP del servidor NTP 1
ADDR	Array[1...4] of Byte	0	
ADDR[1]	Byte	0	Dirección IP high byte
ADDR[2]	Byte	0	Dirección IP high byte
ADDR[3]	Byte	0	Dirección IP low byte
ADDR[4]	Byte	0	Dirección IP low byte
NTP_IP[2]	IP_V4		Direcciones IP del servidor NTP 2
ADDR	Array[1...4] of Byte	0	
ADDR[1]	Byte	0	Dirección IP high byte
ADDR[2]	Byte	0	Dirección IP high byte
ADDR[3]	Byte	0	Dirección IP low byte
ADDR[4]	Byte	0	Dirección IP low byte
NTP_IP[3]	IP_V4		Direcciones IP del servidor NTP 3
ADDR	Array[1...4] of Byte	0	
ADDR[1]	Byte	0	Dirección IP high byte
ADDR[2]	Byte	0	Dirección IP high byte
ADDR[3]	Byte	0	Dirección IP low byte
ADDR[4]	Byte	0	Dirección IP low byte
NTP_IP[4]	IP_V4		Direcciones IP del servidor NTP 4
ADDR	Array[1...4] of Byte	0	
ADDR[1]	Byte	0	Dirección IP high byte
ADDR[2]	Byte	0	Dirección IP high byte
ADDR[3]	Byte	0	Dirección IP low byte
ADDR[4]	Byte	0	Dirección IP low byte

Ejemplo: utilizar la instrucción T_CONFIG para cambiar parámetros IP

En el ejemplo siguiente, en el subcampo "addr", se han cambiado la "InterfaceAddress" (dirección IP), la "SubnetMask", y el "DefaultRouter" (router IP). En la página "Dirección Ethernet" de las "Propiedades" de la CPU hay que seleccionar el botón de opción "Permitir ajustar la dirección IP directamente en el dispositivo" para cambiar los parámetros IP mediante la instrucción "T_CONFIG" después de descargar el programa.

CONF_DATA_1

	Nombre	Tipo de datos	Valor de arranq...
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

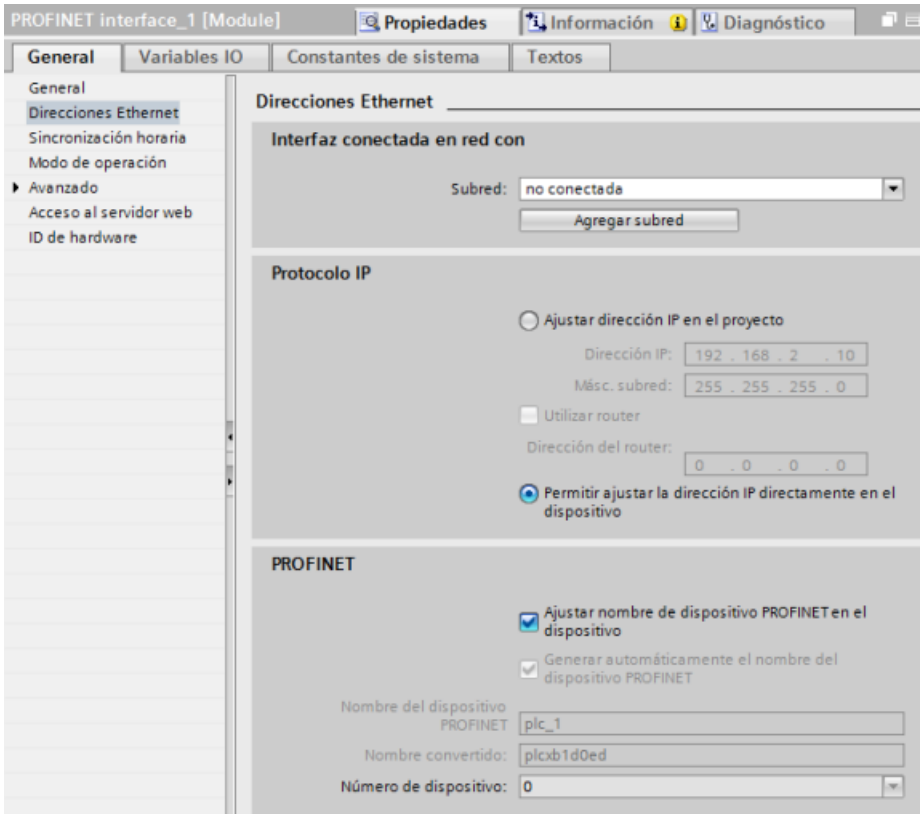

PROFINET interface_1 [Module]
Propiedades
Información
Diagnóstico
General
Variables IO
Constantes de sistema
Textos
General
Direcciones Ethernet
Sincronización horaria
Modo de operación
Avanzado
Acceso al servidor web
ID de hardware
Direcciones Ethernet
Interfaz conectada en red con
Subred:
no conectada
Agregar subred
Protocolo IP
Ajustar dirección IP en el proyecto
Dirección IP:
192 . 168 . 2 . 10
Másc. subred:
255 . 255 . 255 . 0
Utilizar router
Dirección del router:
0 . 0 . 0 . 0
Permitir ajustar la dirección IP directamente en el dispositivo
PROFINET
Ajustar nombre de dispositivo PROFINET en el dispositivo
Generar automáticamente el nombre del dispositivo PROFINET
Nombre del dispositivo PROFINET
plc_1
Nombre convertido:
plcxb1d0ed
Número de dispositivo:
0

Ejemplo: utilizar la instrucción T_CONFIG para cambiar los parámetros IP y los nombres de dispositivo PROFINET IO

En el ejemplo siguiente, se han cambiado los subcampos "addr" y "nos" (Name of station). En la página "Dirección Ethernet" de las "Propiedades" de la CPU hay que seleccionar la casilla de verificación "Permitir ajustar el nombre de dispositivo PROFINET directamente en el dispositivo" para cambiar el nombre de dispositivo PROFINET mediante la instrucción "T_CONFIG" después de descargar el programa.

CONF_DATA_2

	Nombre	Tipo de datos	Valor de arranq...
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	



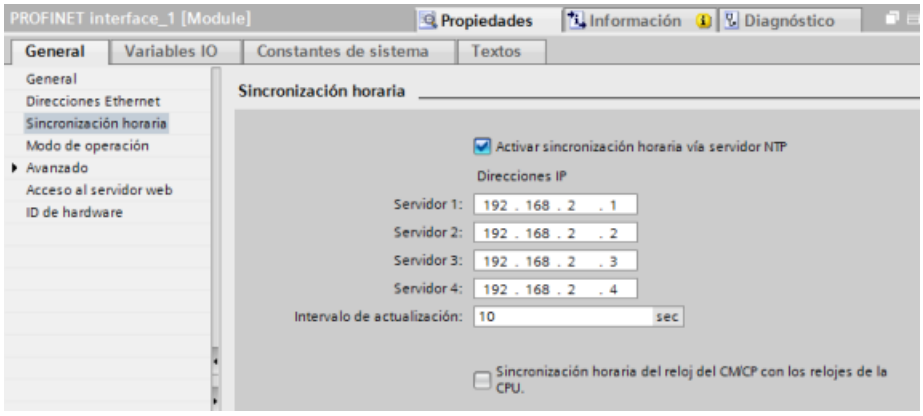
Ejemplo: Uso de la instrucción T_CONFIG para cambiar las direcciones IP en los servidores NTP

En el ejemplo siguiente, la instrucción T_CONFIG cambia las direcciones IP de un máximo de cuatro servidores NTP en el subcampo "ntp" (servidor NTP (Network Time Protocol)).

En la página "Sincronización horaria" de la interfaz PROFINET [X1], en las propiedades de la CPU, la sincronización NTP se configura seleccionando la casilla de verificación "Activar sincronización horaria vía servidor NTP" tal como se muestra en la figura inferior. Las direcciones IP de los servidores NTP se cambian mediante la instrucción "T_CONFIG" después de descargar el programa.

CONF_DATA_3

	Nombre	Tipo de datos	Valor de arranq...
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



11.5.8.20 Parámetros comunes para instrucciones

Parámetro de entrada REQ

Muchas de las instrucciones de la comunicación abierta utilizan la entrada REQ para iniciar la operación en una transición de "low" a "high". El estado lógico de la entrada REQ debe ser "high" (TRUE) durante una ejecución de la instrucción, aunque puede permanecer TRUE durante un tiempo cualquiera. La instrucción no inicia ninguna operación diferente hasta que no sea ejecutada con la entrada REQ puesta a FALSE, de manera que pueda inicializar el histórico de la entrada REQ. Esto es necesario para que la instrucción pueda detectar la transición de "low" a "high" para iniciar la siguiente operación.

Al insertar una de estas instrucciones en su programa, STEP 7 le solicitará identificar el DB de instancia. Utilice un DB unívoco para cada llamada de la instrucción. Esto garantiza que cada instrucción procese correctamente la entradas, p. ej. REQ.

Parámetro de entrada ID

Esta es una referencia de la "ID Local (hex) en la "Vista de red" de "Dispositivos y redes" de STEP 7 y es la ID de la red que usted desea utilizar para este bloque de comunicación. La ID debe ser idéntica al parámetro ID asociado en la descripción de la conexión local.

Parámetros de salida DONE, NDR, ERROR y STATUS

Estas instrucciones ponen a disposición salidas que describen el estado de finalizado:

Tabla 11-57 Parámetros de salida de las instrucciones de la comunicación abierta

Parámetro	Tipo de datos	Valor predeter-minado	Descripción
DONE	Bool	FALSE	Si adopta el estado lógico TRUE durante una ejecución indica que la última petición se ha finalizado sin errores; de lo contrario, FALSE.
NDR	Bool	FALSE	Si adopta el estado lógico TRUE durante una ejecución indica que la acción solicitada se ha finalizado sin errores y que se han recibido datos nuevos; de lo contrario, FALSE.

Parámetro	Tipo de datos	Valor predeterminado	Descripción
BUSY	Bool	FALSE	Si adopta el estado lógico TRUE mientras está activo indica que: • La tarea no ha finalizado aún. • No se puede iniciar una tarea nueva. Adopta el estado lógico FALSE cuando la tarea ha finalizado.
ERROR	Bool	FALSE	Si adopta el estado lógico TRUE durante una ejecución indica que la última petición se ha finalizado con errores. El código de error aplicable aparece en STATUS; de lo contrario, FALSE.
STATUS	Word	0	Resultado del estado: • Si se activa el bit DONE o NDR, STATUS se pone a 0 o a un código de información. • Si se activa el bit ERROR, STATUS se pone a un código de error. • Si no se activa ninguno de estos bits, la instrucción devuelve resultados de estado que describen el estado actual de la función. STATUS conserva su valor durante la ejecución de la función.

Nota

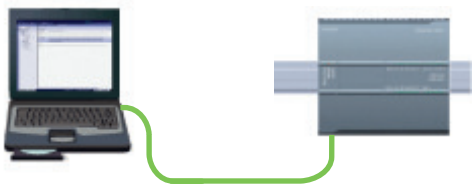
Observe que DONE, NDR y ERROR sólo están activados durante una ejecución.

TSAP y números de puerto restringidos para la comunicación ISO y TCP pasiva

Si la instrucción "TCON" se utiliza para configurar y establecer una conexión pasiva, las siguientes direcciones de puerto están restringidas y no se deben utilizar:

- TSAP ISO (pasivo):
  - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
  - 10.00, 10.01, 11.00, 11.01, ... BF.00, BF.01
- Puerto TCP (pasivo) y puerto UDP (pasivo):
  - 25, 80, 102, 5001, 34962, 34963, 34964

11.5.9 Comunicación con una programadora



Una CPU puede comunicarse con una programadora con STEP 7 en una red.

Al configurar la comunicación entre una CPU y una programadora debe considerarse lo siguiente:

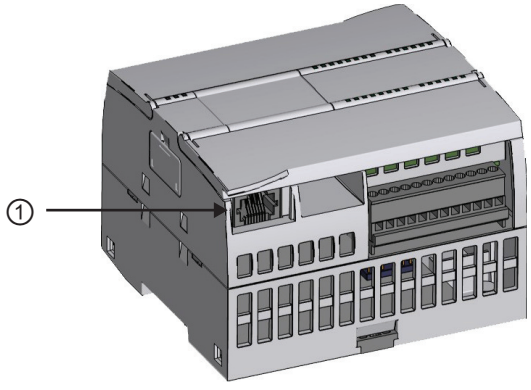
- Configuración/instalación: Es preciso configurar el hardware.
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

11.5.9.1 Establecer la conexión de hardware

Las interfaces PROFINET establecen las conexiones físicas entre una programadora y una CPU. Puesto que la CPU ofrece la función "auto-crossover", es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar una programadora directamente a una CPU no se requiere un switch Ethernet.

Para crear la conexión de hardware entre una programadora y una CPU, proceda del siguiente modo:

1. Monte la CPU (Página 49).
2. Conecte el cable Ethernet al puerto PROFINET que se muestra abajo.
3. Conecte el cable Ethernet a la programadora.



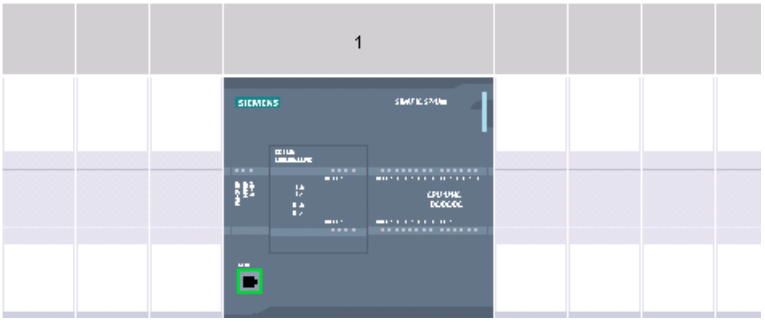
① Puerto PROFINET

Hay una descarga de tracción opcional disponible para reforzar la conexión PROFINET. Encontrará información para realizar el pedido en Repuestos y hardware adicional (Página 1439).

11.5.9.2 Configurar los dispositivos

Si ya se ha creado un proyecto con una CPU, ábralo en STEP 7.

En caso contrario, cree un proyecto e inserte una CPU (Página 138) en el rack. En el proyecto que aparece abajo, una CPU se muestra en la "Vista de dispositivos".



11.5.9.3 Asignar direcciones IP (Internet Protocol)

Asignar direcciones IP

En una red PROFINET todo dispositivo debe tener también una dirección IP (Internet Protocol o Protocolo Internet). Esta dirección permite al dispositivo transferir datos a través de una red enrutada y más compleja:

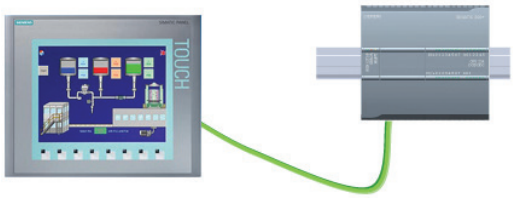
- Si tiene una programadora u otros dispositivos de red que utilicen una tarjeta adaptadora integrada conectada a la LAN corporativa o a una tarjeta adaptadora Ethernet a USB conectada a una red aislada, asígneles direcciones IP. Para más información, consulte el apartado "Asignar direcciones IP a los dispositivos de programación y red" (Página 599).
- También es posible asignar una dirección IP a una CPU o a un dispositivo de red online. Esto es especialmente útil al configurar los dispositivos por primera vez. Para más información, consulte el apartado "Asignar una dirección IP a una CPU online" (Página 599).
- Tras configurar la CPU o el dispositivo de red en su proyecto, usted puede configurar parámetros para la interfaz PROFINET, para incluir la dirección IP correspondiente. Encontrará más información al respecto en "Configurar una dirección IP para una CPU en el proyecto". (Página 602)

11.5.9.4 Comprobar la red PROFINET

Tras finalizar la configuración es necesario cargar el proyecto en la CPU. Todas las direcciones IP se configuran al cargar el proyecto en el dispositivo.

La función de la CPU "Cargar en dispositivo" y su cuadro de diálogo "Carga avanzada en dispositivo" permiten visualizar todos los dispositivos de red accesibles y verificar si se han asignado direcciones IP unívocas a todos ellos. Para más información, consulte "Comprobar la red PROFINET" (Página 607).

11.5.10 Comunicación entre dispositivos HMI y el PLC



La CPU admite conexiones de comunicación PROFINET con dispositivos HMI (Página 32). Los siguientes requisitos deben considerarse al configurar la comunicación entre CPUs y HMIs:

Configuración/instalación:

- El puerto PROFINET de la CPU debe configurarse para poder establecer una conexión con el HMI.
- El HMI se debe instalar y configurar.
- La información de configuración del HMI forma parte del proyecto de la CPU y se puede configurar y cargar desde el proyecto.
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

Nota

El switch Ethernet de 4 puertos CSM1277 montado en un rack puede utilizarse para conectar las CPUs y los dispositivos HMI. El puerto PROFINET de la CPU no contiene un dispositivo de conmutación Ethernet.

Funciones soportadas:

- El HMI puede leer/escribir datos en la CPU.
- Es posible disparar mensajes, según la información consultada de la CPU.
- Diagnóstico del sistema

Tabla 11-58 Pasos necesarios para configurar la comunicación entre un dispositivo HMI y una CPU

Paso	Tarea
1	Establecer la conexión de hardware Una interfaz PROFINET establece la conexión física entre un dispositivo HMI y una CPU. Puesto que la función "auto-crossover" está integrada en la CPU, es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar un HMI a una CPU no se requiere un switch Ethernet. Para más información, consulte "Comunicación con una programadora: Establecer la conexión de hardware". (Página 725)
2	Configurar los dispositivos Para más información, consulte "Comunicación con una programadora: Configurar los dispositivos". (Página 725)
3	Configurar las conexiones de red lógicas entre un dispositivo HMI y una CPU Encontrará más información en "Comunicación entre HMI y PLC: Configurar las conexiones de red lógicas entre dos dispositivos (Página 728)".

Paso	Tarea
4	Configurar una dirección IP en el proyecto Utilice el mismo proceso de configuración. No obstante, es preciso configurar direcciones IP para el HMI y la CPU. Encontrará más información en "Configuración de dispositivos: Configurar una dirección IP para una CPU en el proyecto". (Página 603)
5	Comprobar la red PROFINET La configuración debe cargarse en cada CPU y cada dispositivo HMI. Encontrará más información en "Configuración de dispositivos: Comprobar la red PROFINET". (Página 607)

11.5.10.1 Configurar las conexiones de red lógicas entre dos dispositivos

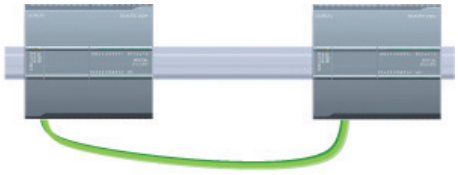
Tras configurar el rack con la CPU podrá configurar las conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. En primer lugar, haga clic en la ficha "Conexiones" y luego seleccione el tipo de conexión en la lista desplegable ubicada justo a la derecha (p. ej. una conexión ISO on TCP).

Para crear una conexión PROFINET, haga clic en el cuadro verde (PROFINET) del primer dispositivo y trace una línea hasta el cuadro PROFINET del segundo dispositivo. Suelte el botón del ratón para crear la conexión PROFINET.

Para más información, consulte el apartado "Configuración de dispositivos: Crear una conexión de red". (Página 596)

11.5.11 Comunicación entre PLCs



Una CPU puede comunicarse con otra CPU utilizando las instrucciones TSEND_C y TRCV_C.

Considere lo siguiente al configurar la comunicación entre dos CPUs:

- Configuración/instalación: Es preciso configurar el hardware.
- Funciones soportadas: Leer/escribir datos en una CPU interlocutora
- Para la comunicación entre dos interlocutores no se requiere un switch Ethernet. Un switch Ethernet se requiere sólo si la red comprende más de dos dispositivos.

Tabla 11-59 Pasos necesarios para configurar la comunicación entre dos CPUs

Paso	Tarea
1	Establecer la conexión de hardware Una interfaz PROFINET establece la conexión física entre dos CPUs. Puesto que la función "auto-crossover" está integrada en la CPU, es posible utilizar un cable Ethernet estándar o cruzado ("crossover") para la interfaz. Para conectar dos CPUs no se requiere un switch Ethernet. Para más información, consulte "Comunicación con una programadora: Establecer la conexión de hardware". (Página 725)
2	Configurar los dispositivos Hay que configurar dos CPUs en el proyecto. Para más información, consulte "Comunicación con una programadora: Configurar los dispositivos". (Página 725)
3	Configurar las conexiones de red lógicas entre dos CPUs Encontrará más información en "Comunicación entre PLC y PLC: Configurar las conexiones de red lógicas entre dos dispositivos (Página 729)".
4	Configurar una dirección IP en el proyecto Utilice el mismo proceso de configuración. No obstante, es preciso configurar direcciones IP para dos CPUs (p. ej. PLC_1 y PLC_2). Encontrará más información en "Configuración de dispositivos: Configurar una dirección IP para una CPU en el proyecto". (Página 603)
5	Configurar los parámetros de transmisión y recepción Las instrucciones TSEND_C y TRCV_C deben configurarse en ambas CPUs para habilitar la comunicación entre ellas. Encontrará más información en el apartado "Configurar la comunicación entre dos CPUs: Configurar los parámetros de transmisión y recepción". (Página 730)
6	Comprobar la red PROFINET La configuración debe cargarse en cada una de las CPUs. Encontrará más información en "Configuración de dispositivos: Comprobar la red PROFINET (Página 607)".

11.5.11.1 Configurar las conexiones de red lógicas entre dos dispositivos

Tras configurar el rack con la CPU podrá configurar las conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. En primer lugar, haga clic en la ficha "Conexiones" y luego seleccione el tipo de conexión en la lista desplegable ubicada justo a la derecha (p. ej. una conexión ISO on TCP).

Para crear una conexión PROFINET, haga clic en el cuadro verde (PROFINET) del primer dispositivo y trace una línea hasta el cuadro PROFINET del segundo dispositivo. Suelte el botón del ratón para crear la conexión PROFINET.

Para más información, consulte el apartado "Configuración de dispositivos: Crear una conexión de red". (Página 596)

11.5.11.2 Configurar la vía de conexión local/interlocutor entre dos dispositivos

Configurar los parámetros generales

Los parámetros de comunicación se especifican en el cuadro de diálogo de configuración "Propiedades" de la instrucción de comunicación. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción.

Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los TSAPs o puertos que deben utilizarse. El TSAP o puerto de una conexión en la CPU se introduce en el campo "TSAP local". El TSAP o puerto asignado a la conexión en la CPU interlocutora se introduce en el campo "TSAP del interlocutor".

11.5.11.3 Configurar los parámetros de transmisión y recepción

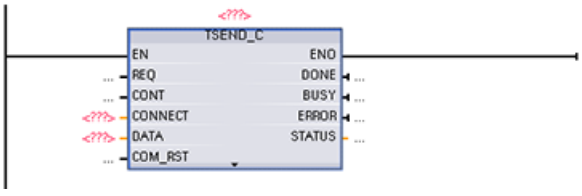
Los bloques de comunicación (por ejemplo TSEND_C y TRCV_C) sirven para establecer conexiones entre dos CPUs. Para que las CPUs puedan intervenir en la comunicación PROFINET es preciso configurar parámetros para transmitir y recibir mensajes. Estos parámetros determinan cómo deben funcionar las comunicaciones al transmitir o recibir mensajes a/de un dispositivo de destino.

Configurar los parámetros de transmisión de la instrucción TSEND_C

Instrucción TSEND_C

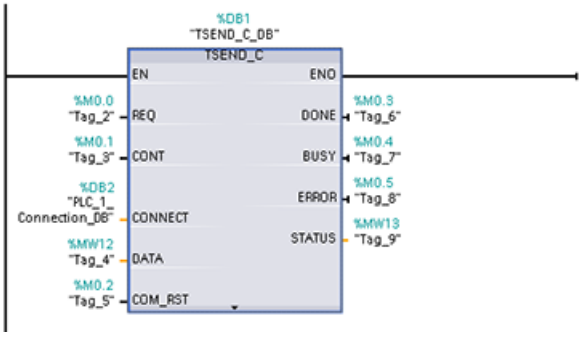
La instrucción TSEND_C (Página 631) crea una conexión con un interlocutor. La conexión se configura, establece y vigila automáticamente hasta que la instrucción ordene que sea desconectada. La instrucción TSEND_C combina las funciones de las instrucciones TCON, TDISCON y TSEND.

En la "Configuración de dispositivos" de STEP 7 es posible configurar cómo la instrucción TSEND_C debe transmitir los datos. Para comenzar, inserte la instrucción en el programa desde la carpeta "Comunicación" de la Task Card "Instrucciones". La instrucción TSEND_C se visualizará junto con el diálogo "Opciones de llamada" en el que se asigna un DB para almacenar los parámetros de la instrucción.





Como muestra la figura siguiente, es posible asignar posiciones de memoria a las entradas y salidas en la memoria de variables:



Configurar los parámetros generales

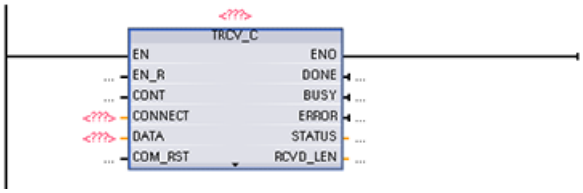
Los parámetros se configuran en el diálogo "Propiedades" de la instrucción TSEND_C. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción TSEND_C.

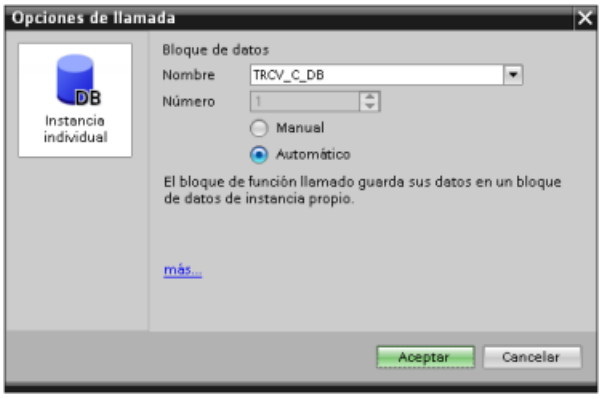
Configurar los parámetros de recepción de la instrucción TRCV_C

Instrucción TRCV_C

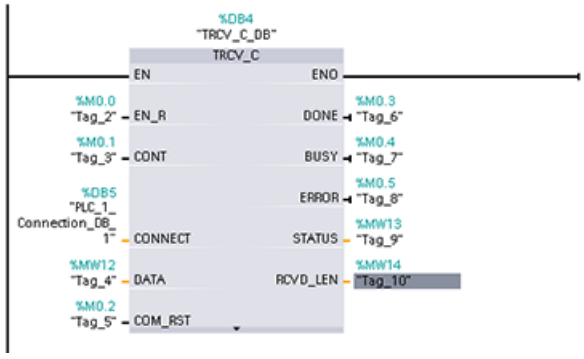
La instrucción TRCV_C (Página 631) crea una conexión con un interlocutor. La conexión se configura, establece y vigila automáticamente hasta que la instrucción ordene que sea desconectada. La instrucción TRCV_C combina las funciones de las instrucciones TCON, TDISCON y TRCV.

Desde la configuración de la CPU en STEP 7 es posible configurar cómo la instrucción TRCV_C debe recibir los datos. Para comenzar, inserte la instrucción en el programa desde la carpeta "Comunicación" de la Task Card "Instrucciones". La instrucción TRCV_C se visualizará junto con el diálogo "Opciones de llamada" en el que se asigna un DB para almacenar los parámetros de la instrucción.





Como muestra la figura siguiente, es posible asignar posiciones de memoria a las entradas y salidas en la memoria de variables:



Configurar los parámetros generales

Los parámetros se configuran en el diálogo "Propiedades" de la instrucción TRCV_C. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción TRCV_C.

11.5.12 Configurar una CPU y un dispositivo PROFINET IO

11.5.12.1 Agregar un dispositivo PROFINET IO

Agregar un dispositivo PROFINET IO

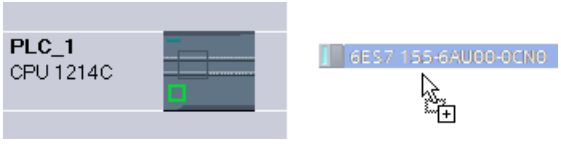
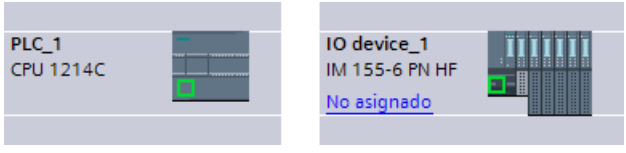
En el portal "Dispositivos y redes", utilice el catálogo de hardware para agregar dispositivos PROFINET IO.

Nota

Para agregar un dispositivo PROFINET IO se puede utilizar STEP 7 Professional o Basic V11 o superior.

Abra por ejemplo las carpetas siguientes en el catálogo de hardware para agregar un dispositivo IO ET 200SP: Periferia descentralizada, ET 200SP, módulos de interfaz y PROFINET. A continuación, seleccione el módulo de interfaz en la lista de dispositivos ET 200SP (ordenados por referencia) y agregue el dispositivo ET 200SP.

Tabla 11-60 Agregar un dispositivo IO ET 200SP a la configuración de dispositivos

Insertar el dispositivo IO	Resultado
	

Ahora, el dispositivo PROFINET IO puede conectarse a la CPU:

1. Haga clic con el botón derecho del ratón en el enlace "No asignado" del dispositivo y seleccione "Asignar a controlador IO nuevo" del menú contextual para visualizar el cuadro de diálogo "Seleccionar controlador IO".
2. Seleccione la CPU S7-1200 (en el ejemplo "PLC_1") en la lista de controladores del proyecto.
3. Haga clic en "Aceptar" para crear la conexión de red.

También es posible ir al portal "Dispositivos y redes" y utilizar la "Vista de redes" para crear las conexiones de red entre los dispositivos del proyecto:

1. Para crear una conexión PROFINET, haga clic en el cuadro verde (PROFINET) del primer dispositivo y trace una línea hasta el cuadro PROFINET del segundo dispositivo.
2. Suelte el botón del ratón para crear la conexión PROFINET.

Encontrará más información en "Configuración de dispositivos: Configurar la CPU para la comunicación" (Página 171).

11.5.12.2 Asignar CPUs y nombres de dispositivo

Asignar CPUs y nombres de dispositivo

Las conexiones de red entre los dispositivos también asignan el dispositivo PROFINET IO a la CPU, algo necesario para que la CPU pueda controlar dicho dispositivo. Para modificar la asignación, haga clic en el nombre del PLC que se muestra en el dispositivo PROFINET IO. Se abre un cuadro de diálogo que permite desconectar el dispositivo PROFINET IO de la CPU actual y reasignarlo o bien dejarlo sin asignación, como se prefiera.

Los dispositivos de la red PROFINET deben tener asignados nombres antes de poder ser conectados a la CPU. Utilice la "Vista de redes" para asignar nombres a los dispositivos PROFINET, en caso de que estos no hayan sido asignados o bien si el nombre del dispositivo debe ser modificado. Para ello, haga clic con el botón derecho del ratón en el dispositivo PROFINET IO y seleccione "Asignar nombre de dispositivo".

A cada dispositivo PROFINET IO se le debe asignar el mismo nombre en el proyecto de STEP 7 y en la red PROFINET. (Para asignar el nombre de dispositivo puede utilizarse la herramienta "Online y diagnóstico" en STEP 7 o la herramienta de puesta en marcha, configuración y diagnóstico PRONETA en la red PROFINET.) Si hace falta un nombre o este no coincide en ninguna ubicación, el modo de intercambio de datos PROFINET IO no se ejecutará. Encontrará más información en "Herramientas online y diagnóstico: Asignar un nombre a un dispositivo PROFINET online (Página 1202)".

11.5.12.3 Asignar direcciones IP (Internet Protocol)

Asignar direcciones IP

En una red PROFINET todo dispositivo debe tener también una dirección IP (Internet Protocol o Protocolo Internet). Esta dirección permite al dispositivo transferir datos a través de una red enrutada y más compleja:

- Si tiene una programadora u otros dispositivos de red que utilicen una tarjeta adaptadora integrada conectada a la LAN corporativa o a una tarjeta adaptadora Ethernet a USB conectada a una red aislada, asígneles direcciones IP. Para más información, consulte el apartado "Asignar direcciones IP a los dispositivos de programación y red" (Página 599).
- También es posible asignar una dirección IP a una CPU o a un dispositivo de red online. Esto es especialmente útil al configurar los dispositivos por primera vez. Para más información, consulte el apartado "Asignar una dirección IP a una CPU online" (Página 602).
- Tras configurar la CPU o el dispositivo de red en su proyecto, usted puede configurar parámetros para la interfaz PROFINET, para incluir la dirección IP correspondiente. Encontrará más información al respecto en "Configurar una dirección IP para una CPU en el proyecto". (Página 603)

11.5.12.4 Configurar el tiempo de ciclo IO

Configurar el tiempo de ciclo IO

Un dispositivo PROFINET IO recibe datos nuevos de la CPU dentro de un intervalo de tiempo "Ciclo IO". El tiempo de actualización puede configurarse por separado para cada dispositivo y determina el intervalo de tiempo en el que los datos se transmiten de la CPU al dispositivo y viceversa.

STEP 7 calcula automáticamente el tiempo de actualización del "Ciclo IO" en la configuración predeterminada de cada dispositivo de la red PROFINET, tomando en consideración el volumen de datos que deben intercambiarse y el número de dispositivos asignados a dicho controlador. Si no desea que el tiempo de actualización se calcule automáticamente, es posible cambiar este ajuste.

Los parámetros del "Ciclo IO" se especifican en el cuadro de diálogo de configuración "Propiedades" del dispositivo PROFINET IO. Este cuadro de diálogo aparece en la parte inferior de la página cuando se ha seleccionado alguna parte de la instrucción.

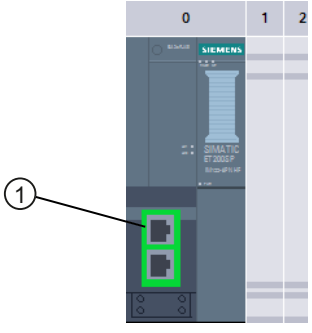
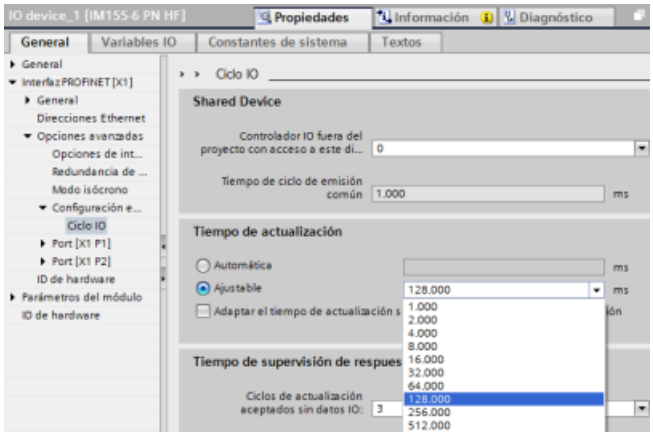
En la "Vista de dispositivos" del dispositivo PROFINET IO, haga clic en el puerto PROFINET. En el cuadro de diálogo "Interfaz PROFINET", acceda a los parámetros del "Ciclo IO" con las siguientes selecciones de menú:

- "Opciones avanzadas"
- "Configuración en tiempo real"
- "Ciclo IO"

Defina el "Tiempo de actualización" del ciclo IO con las selecciones siguientes:

- Para que se calcule automáticamente un tiempo de actualización adecuado, seleccione "Automático".
- Para ajustar uno mismo el tiempo de actualización, seleccione "Ajustable" e introduzca el tiempo de actualización necesario en ms.

Tabla 11-61 Configurar el tiempo de ciclo PROFINET IO en un ET 200SP

Dispositivo PROFINET IO ET 200SP	Cuadro de diálogo del ciclo PROFINET IO en un ET 200SP
	

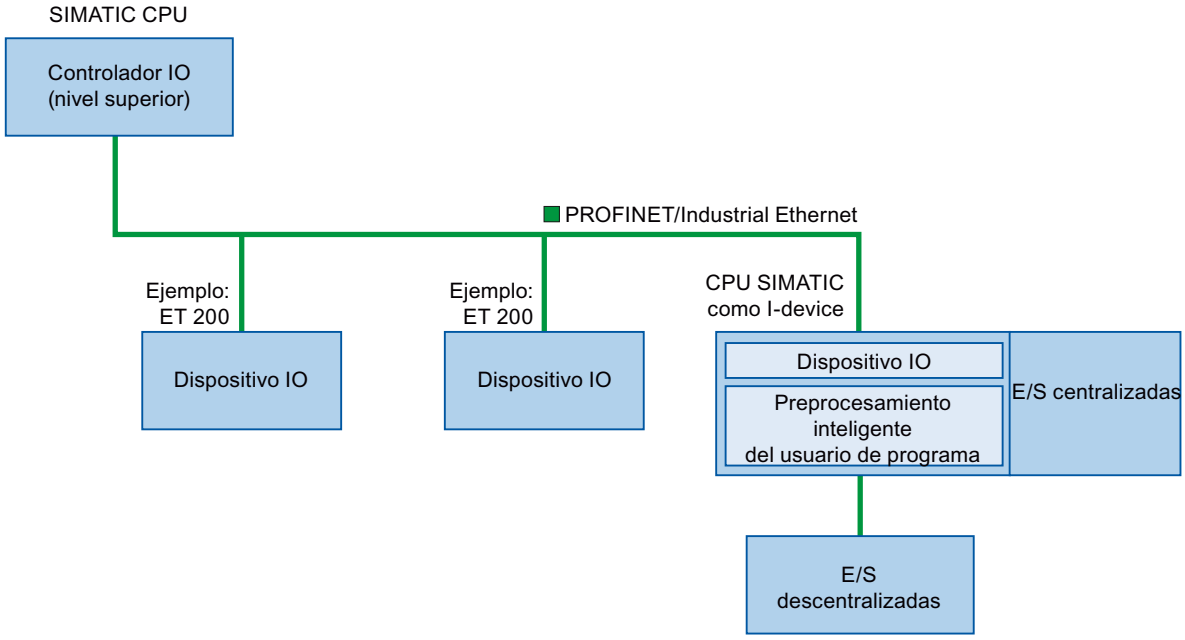
① Puerto PROFINET

11.5.13 Configurar una CPU y un I-device PROFINET

11.5.13.1 Funcionalidad de I-device

La funcionalidad de "I-device" (dispositivo IO inteligente) de una CPU facilita el intercambio de datos con un controlador IO y el funcionamiento de la CPU como una unidad de preprocesamiento inteligente de subprocesos, por ejemplo. El I-device está conectado como un dispositivo IO a un controlador IO de "nivel superior".

El programa de usuario gestiona el preprocesamiento en la CPU. El programa de usuario preprocesa los valores de proceso adquiridos en las E/S centralizadas o descentralizadas (PROFINET IO o PROFIBUS DP) y pasan a estar disponibles mediante la interfaz de PROFINET IO para la CPU de una estación de nivel superior.



Convenciones para nombres de "I-device"

En lo que resta de esta descripción, una CPU o un CP con funcionalidad I-device se denominará "I-device".

11.5.13.2 Propiedades y ventajas del I-device

Campos de aplicación

Campos de aplicación del I-device:

- Procesamiento descentralizado:
  Una tarea de automatización compleja se puede dividir en unidades/subprocesos más pequeños. En consecuencia, se obtienen procesos más manejables que permiten simplificar las subtareas.
- Separación de subprocesos:
  Los procesos complicados, extensos y de amplia distribución se pueden subdividir en varios procesos con interfaces manejables mediante I-devices. Si es necesario, estos subprocesos se pueden almacenar en proyectos STEP 7 individuales, que posteriormente se pueden fusionar para crear un proyecto maestro.
- Protección de know-how:
  Los componentes solo se pueden suministrar con un archivo GSD para la descripción de interfaz de I-device en lugar de con un proyecto STEP 7. El usuario puede proteger su programa puesto que ya no se debe publicar otra vez.

Propiedades

Propiedades de I-device:

- Anulación de enlaces entre proyectos STEP 7:
  Los creadores y usuarios de un I-device pueden tener proyectos de automatización STEP 7 completamente separados. El archivo GSD forma la interfaz entre los proyectos STEP 7. Este permite establecer un enlace con los controladores IO estándar a través de una interfaz estandarizada.
- Comunicaciones en tiempo real:
  el I-device dispone de un sistema PROFINET IO determinístico a través de una interfaz PROFINET IO.

Ventajas

El I-device tiene las siguientes ventajas:

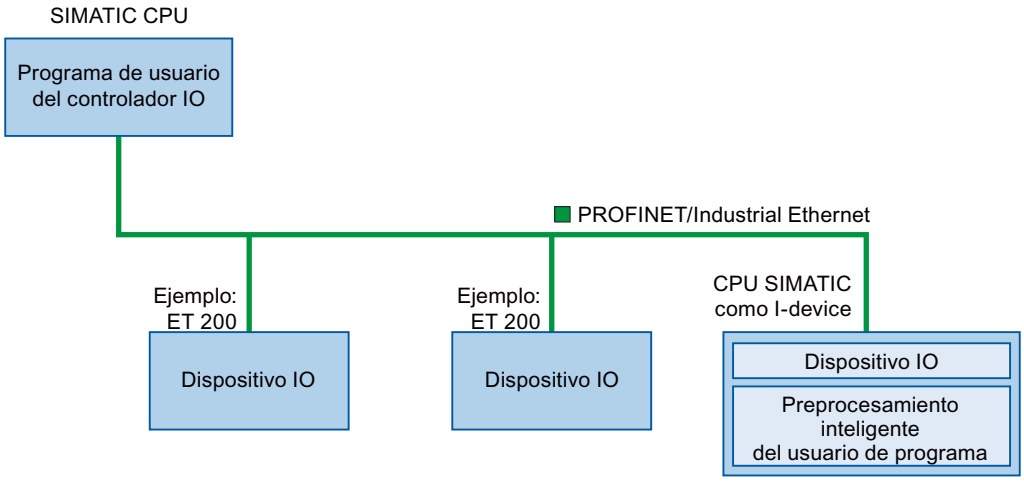
- Enlace sencillo de controladores IO
- Comunicación en tiempo real entre controladores IO
- Descarga de trabajo del controlador IO descentralizando la capacidad informática a I-devices
- Reducción de la carga de comunicación mediante el procesamiento de datos local
- Manejo más fácil gracias al procesamiento de subtareas en proyectos STEP 7 separados

11.5.13.3 Características de un I-device

Un I-device se incluye en un sistema IO como si fuera un dispositivo IO estándar.

I-device sin sistema PROFINET IO subordinado

El I-device no cuenta con su propia periferia descentralizada. La asignación de configuraciones y parámetros de los I-devices asumiendo la función de dispositivo IO es igual que para un sistema de periferia descentralizada (por ejemplo, ET 200).



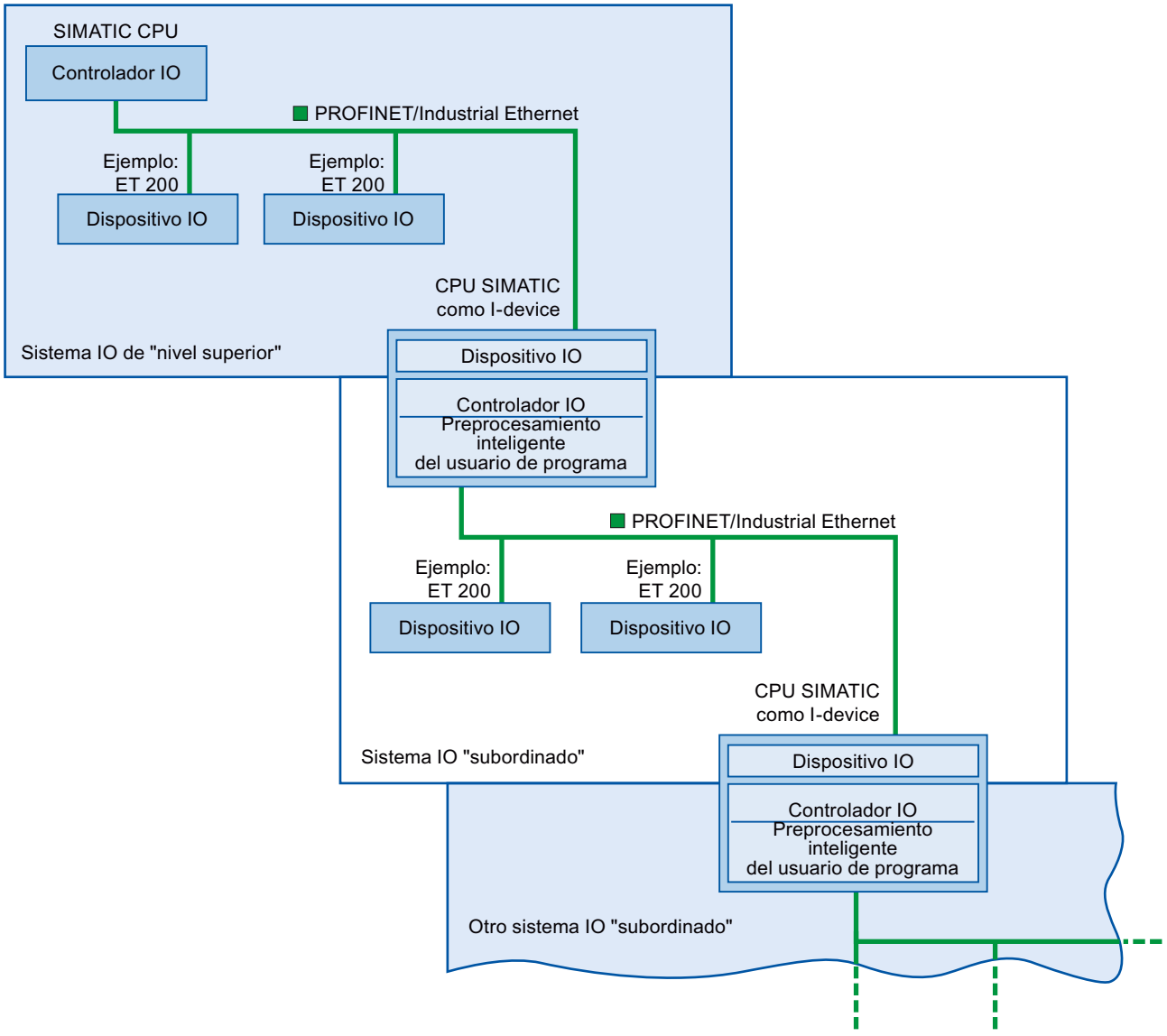
I-device con sistema PROFINET IO subordinado

Dependiendo de la configuración, un I-device también puede ser un controlador IO en una interfaz PROFINET, además de asumir la función de un dispositivo IO.

De este modo, el I-device puede formar parte de un sistema IO de nivel superior a través de su interfaz PROFINET y como controlador IO puede admitir su propio sistema IO subordinado.

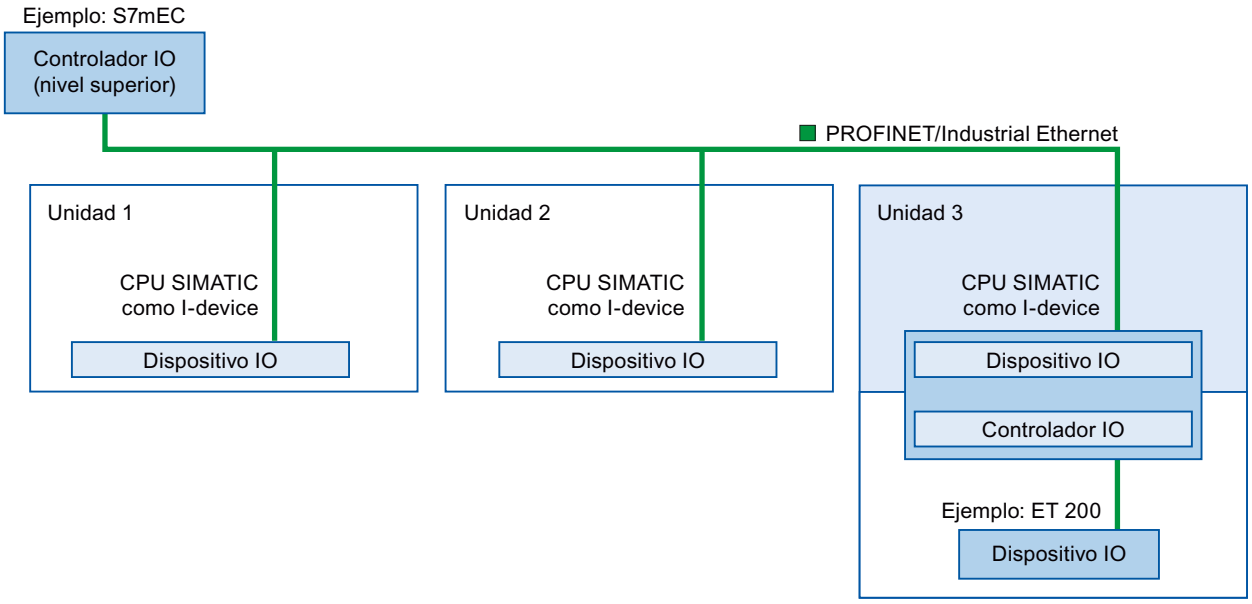
El sistema IO de subordinado, a su vez, puede contener I-devices (véase la figura más abajo), lo cual permite contar con sistemas IO estructurados jerárquicamente.

Además de su función como controlador IO, un I-device también se puede usar a través de una interfaz PROFIBUS como maestro DP para un sistema PROFIBUS subordinado.


SIMATIC CPU
Controlador IO
PROFINET/Industrial Ethernet
Ejemplo:
ET 200
Dispositivo IO
Ejemplo:
ET 200
Dispositivo IO
CPU SIMATIC
como I-device
Sistema IO de "nivel superior"
Dispositivo IO
Controlador IO
Preprocesamiento
inteligente
del usuario de programa
PROFINET/Industrial Ethernet
Ejemplo:
ET 200
Dispositivo IO
Ejemplo:
ET 200
Dispositivo IO
CPU SIMATIC
como I-device
Sistema IO "subordinado"
Dispositivo IO
Controlador IO
Preprocesamiento
inteligente
del usuario de programa
Otro sistema IO "subordinado"

Ejemplo: I-device como dispositivo IO y controlador IO

El I-device como dispositivo IO y controlador IO se explica a partir del ejemplo de un proceso de impresión. El I-device controla una unidad (un subproceso). Una unidad se usa, por ejemplo, para insertar hojas adicionales como octavillas o folletos en un paquete de material impreso.



La unidad 1 y la unidad 2 constan de un I-device con E/S centralizadas. El I-device junto con el sistema de E/S descentralizadas (por ejemplo, ET 200) forma la unidad 3.

El programa de usuario en el I-device se encarga de preprocesar los datos de proceso. Para esta tarea, el programa de usuario del I-device necesita una configuración predeterminada (por ejemplo, datos de control) del controlador IO de nivel superior. El I-device proporciona un controlador IO de nivel superior con los resultados (por ejemplo, el estado de su subtarea).

11.5.13.4 Intercambio de datos entre un sistema IO de nivel superior y otro subordinado

Las áreas de transferencia son una interfaz con el programa del usuario de la CPU del I-device. Las entradas se procesan en el programa de usuario y las salidas se obtienen del procesamiento en el programa del usuario.

Los datos para establecer comunicación entre el controlador IO y el I-device pasan a estar disponibles en las áreas de transferencia. Un área de transferencia contiene una unidad informativa que se intercambia uniformemente entre el controlador IO y el I-device. Encontrará más información sobre la configuración y el uso de áreas de transferencia en "Configuración del I-device" (Página 743).

Las áreas de transferencia de entrada tienen un comportamiento diferente cuando se pierde la conexión de red entre el controlador y el I-device

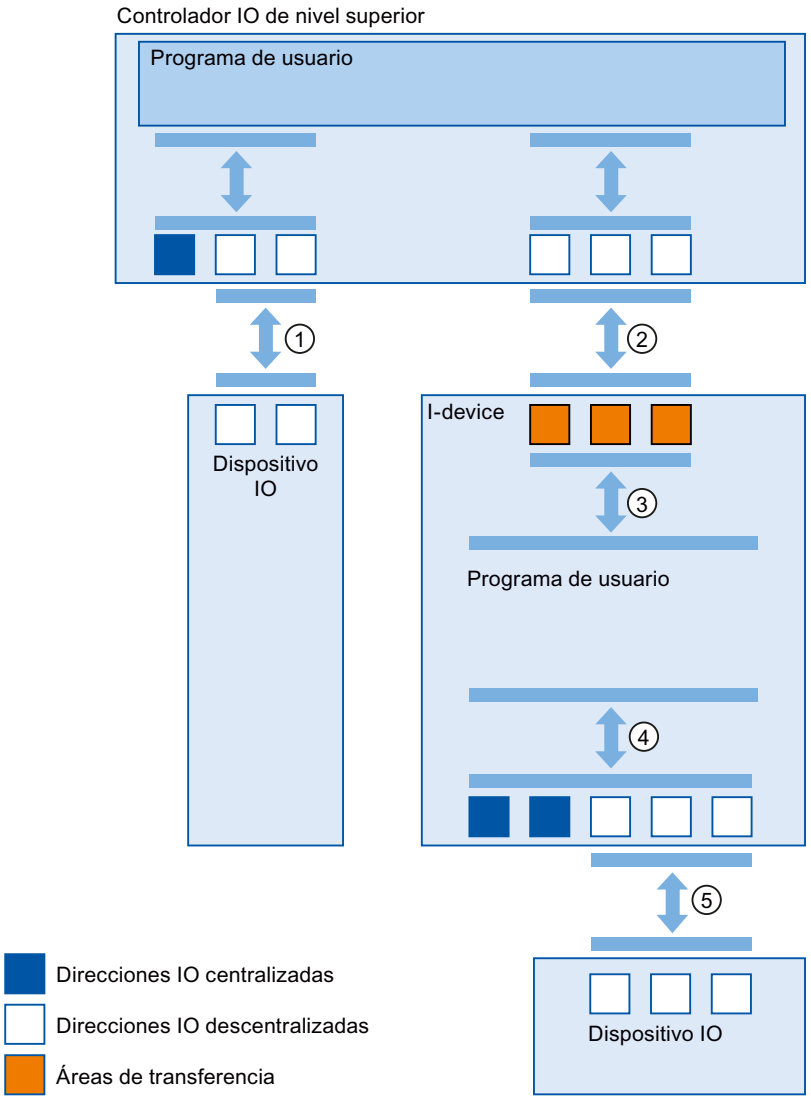
En el controlador, la CPU escribe un cero en las áreas de transferencia de entrada al perderse la conexión de red. En el I-device, las áreas de transferencia de entrada conservan sus últimos valores.

El sistema puede configurarse de forma que se evite esta condición en el caso general de un I-device (I-device no compartido). Para ello, borre las áreas de transferencia de entrada del I-device en un "OB de fallo de rack o estación" para un evento entrante. Proceda del siguiente modo:

1. Agregue un "OB de fallo de rack o estación" al proyecto. (Este OB tiene de forma predeterminada el número de OB 86).
2. Agregue lógica al OB para que los valores de las entradas para el I-device pasen a cero cuando la variable de arranque de LADDR indique el valor del identificador de hardware del I-device y la variable de arranque de Event_Class indique un evento "entrante":
   - Encontrará el identificador de hardware del I-device en la tabla de variables predeterminada de la ficha "Constantes de sistema". El identificador de hardware es un tipo de "HW_Device" y el nombre de la variable indica que es un I-device (p. ej. "Local~PROFINET_interface_1~IODevice").
   - Un valor "16#39" en la Event_Class indica un evento "entrante". Si la variable de entrada "Event_Class" contiene el valor "16#39", esto indica que el "OB de fallo de rack o estación" está activo (en contraposición a cuando se borra).

Flujo de intercambio de datos

En la siguiente figura se muestra el intercambio de datos entre el sistema de nivel superior y el subordinado. Las relaciones de comunicación individuales se explican a continuación en función de los números:



① **Intercambio de datos entre el controlador IO de nivel superior y el dispositivo IO normal**

De este modo, el controlador IO y los dispositivos IO intercambian datos a través dePROFINET.

② **Intercambio de datos entre el controlador IO de nivel superior y el I-device**

De este modo, el controlador IO y el I-device intercambian datos a través dePROFINET.

El intercambio de datos entre un controlador IO de nivel superior y un I-device se basa en la relación convencional entre un controlador IO y un dispositivo IO.

En el caso del controlador IO de nivel superior, las áreas de transferencia de los I-devices representan submódulos de una estación preconfigurada.

Los datos de salida del controlador IO son los datos de entrada del I-device. Por el contrario, los datos de entrada del controlador IO son los datos de salida del I-device.

③ **Relación de transferencia entre el programa de usuario y el área de transferencia**
De este modo, el programa de usuario y el área de transferencia intercambian datos de entrada y de salida.

④ **Intercambio de datos entre el programa del usuario y las E/S del I-device**
De este modo, el programa de usuario y las E/S centralizadas/descentralizadas intercambian datos de entrada y de salida.

⑤ **Intercambio de datos entre el I-device y el dispositivo IO subordinado**
De este modo, el I-device y sus dispositivos IO intercambian datos. La transferencia de datos se realiza a través dePROFINET.

11.5.13.5 Configuración del I-device

Básicamente hay dos posibilidades de configuración:

- Configuración de un I-device en un proyecto
- Configuración de un I-device que se ha usado en otro proyecto o en otro sistema de ingeniería.

STEP 7 permite configurar un I-device para otro proyecto o para otro sistema de ingeniería exportando un I-device configurado a un archivo GSD. El archivo GSD se importa en otros proyectos o sistemas de ingeniería del mismo modo que otros archivos GSD. Las áreas de transferencia para el intercambio de datos, entre otros datos, se almacenan en este archivo GSD.

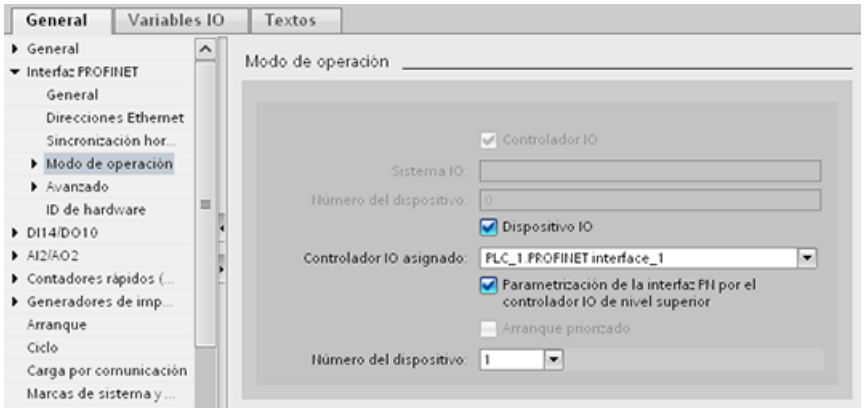
Nota

En caso de utilizar el S7-1200 como shared I-device y como controlador, hay que asegurarse de aumentar los tiempos de actualización de PROFINET IO y del I-device PROFINET con el fin de reducir el impacto en el rendimiento de la comunicación. El sistema es muy estable y funciona bien con un tiempo de actualización de 2 ms para un tiempo de I-device PROFINET individual y de 2 ms para un tiempo de PROFINET IO individual.

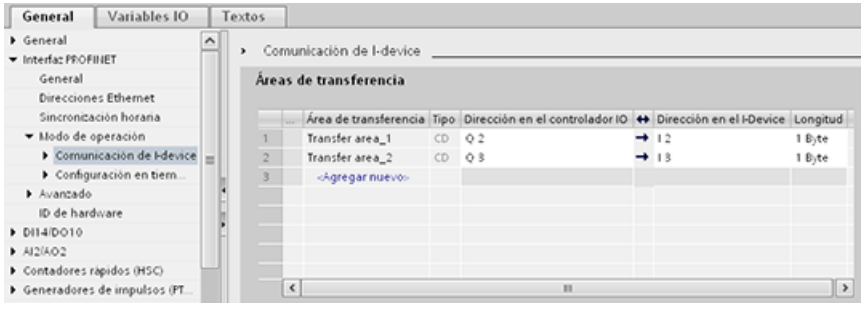
Configuración de un I-device en un proyecto

1. Arrastre y suelte una CPU PROFINET del catálogo de hardware en la vista de red.
2. Arrastre y suelte una CPU PROFINET, que también se puede configurar como un dispositivo IO, del catálogo de hardware en la vista de red. Este dispositivo se ha configurado como I-device (por ejemplo, CPU 1215C).
3. Seleccione la interfaz PROFINET para el I-device.
4. En la ventana Inspector del área de navegación, elija "Estado operativo" y seleccione la casilla de verificación "Dispositivo IO".

5. Ahora tiene la opción de elegir el controlador IO en la lista desplegable "Controlador IO asignado".
   Una vez que haya elegido el controlador IO, el sistema de conexión en red e IO entre los dos dispositivos se mostrará en la vista de red.



6. Mediante la casilla de verificación "Parametrización de la interfaz PN por el controlador IO de nivel superior", se especificará si el propio I-device o bien un controlador IO de nivel superior asignarán los parámetros de interfaz.
   Si utiliza el I-device con un sistema IO subordinado, los parámetros de la interfaz PROFINET de I-device (por ejemplo, el parámetro de puerto) no se podrán asignar con el controlador IO de nivel superior.
7. Configure las áreas de transferencia. Las áreas de transferencia se encuentran en el apartado de navegación del área "Comunicación de I-device":
   - Haga clic en el primer campo de la columna "Área de transferencia". STEP 7 asigna un nombre predeterminado que puede cambiar.
   - Seleccione el tipo de relación de comunicación: actualmente solo puede seleccionar CD o F-CD.
   - Las direcciones se preajustan automáticamente; puede corregir las direcciones, si es necesario, y determinar la longitud del área de transferencia que se transferirá de forma uniforme.



8. En la navegación local se crea una entrada distinta para cada área de transferencia. Si selecciona una de estas entradas, podrá ajustar los detalles del área de transferencia o bien corregirlos y comentarlos.

Nota

Si se configura un S7-1200 como I-device, el tamaño máximo de un área de transferencia es de 1024 bytes de entrada o salida. Hay posibles factores de restricción en función de las E/S locales y de las limitaciones de espacio de direccionamiento en el controlador.

Configuración de un I-device con un archivo GSD

Si utiliza un I-device en otro proyecto o si el I-device se utiliza en otro sistema de ingeniería, configure el controlador IO de nivel superior y el I-device como se ha descrito más arriba.

Sin embargo, haga clic en el botón "Exportar" después de configurar las áreas de transferencia para crear un nuevo archivo GSD con el I-device. Este archivo GSD representa el I-device configurado en otros proyectos.

El botón "Exportar" se encuentra en la sección "Comunicación de I-device" de la ventana de inspección.

La configuración del hardware se compila y el diálogo de exportación se abre.

Asigne un nombre para el proxy de I-device y una descripción en los campos proporcionados. Haga clic en el botón "Exportar" para finalizar el proceso.

Finalmente, importe el archivo GSD, por ejemplo, en otro proyecto.

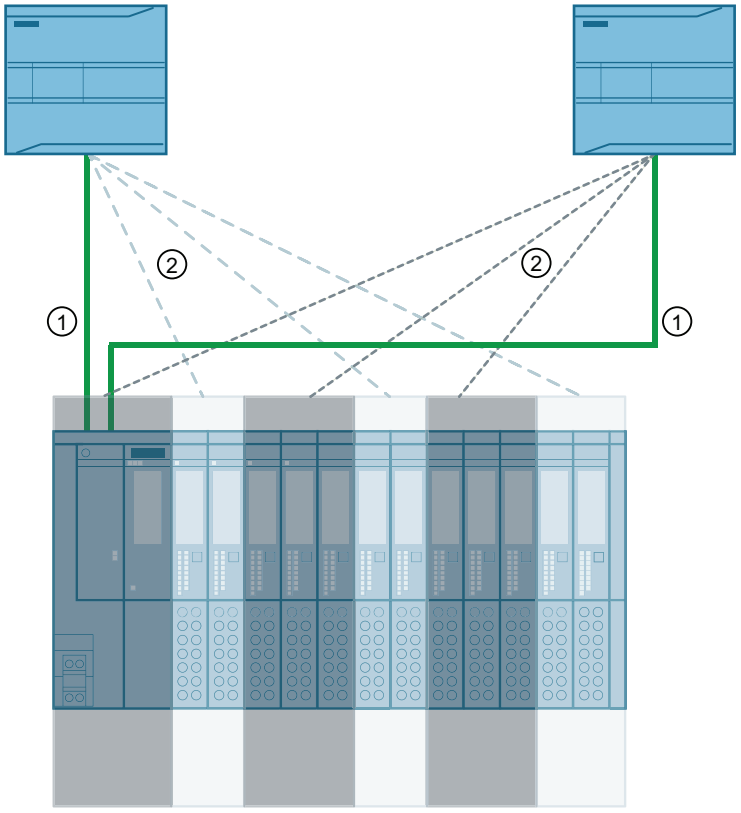
11.5.14 Dispositivos compartidos

11.5.14.1 Funcionalidad de Shared Device

Muchos controladores IO suelen utilizarse en sistemas de gran tamaño o ampliamente descentralizados.

Sin la función "Shared Device", cada módulo E/S de un dispositivo IO se asigna al mismo controlador IO. Si hay sensores que están físicamente cerca unos de otros y deben suministrar datos a diferentes controladores IO, se requieren varios dispositivos IO.

La función "Shared Device" permite dividir los módulos o submódulos de un dispositivo IO entre diferentes controladores IO. De esta forma se consiguen conceptos de automatización flexibles. Así, por ejemplo, existe la posibilidad de combinar módulos E/S que están cerca unos de otros en un solo dispositivo IO.



① PROFINET
② Asignación lógica

Principio

El acceso a los submódulos del Shared Device se divide entre los diferentes controladores IO. Cada submódulo del Shared Device está asignado a un solo controlador IO.

Requisitos (configuración GSD)

- STEP 7 V12 Service Pack 1 o superior
- CPU S7-1200 con firmware V4.1 o superior como controlador IO
- Un dispositivo IO que soporte la funcionalidad de Shared Device, p. ej. un módulo de interfaz IM 155-5 PN ST
- Un archivo GSD instalado para configurar el dispositivo IO
- Una CPU S7-1200 configurada como I-device que soporte la funcionalidad de Shared Device. Hay que exportar el archivo GSD PROFINET para el I-device de STEP 7 (V5.5 o superior) y, seguidamente, importarlo a STEP 7 (TIA Portal).

Configurar el acceso

El dispositivo IO debe estar presente en varios proyectos para que los módulos o submódulos de un dispositivo IO puedan asignarse a diferentes controladores IO. Se necesita un proyecto separado para cada controlador IO.

El parámetro "Shared Device" del módulo de interfaz se utiliza para determinar los módulos o submódulos a los que tiene acceso el controlador IO:

- Si el controlador IO local tiene acceso al módulo configurado, seleccione el nombre del controlador IO en la lista.
- Si el que debe tener acceso al módulo configurado es el controlador IO de otro proyecto y no el controlador IO local, seleccione la entrada "---".

La configuración es coherente en cuanto al acceso si cada módulo o submódulo de un proyecto exactamente está asignado a un controlador IO.

El módulo o submódulo está asignado a otro controlador IO

El párrafo siguiente describe las consecuencias del ajuste "---" en el parámetro "Shared Device" desde el punto de vista del controlador IO local.

En este caso, el controlador IO local no tiene acceso al módulo configurado de este modo. En concreto, esto significa lo siguiente:

- No hay intercambio de datos con el módulo o submódulo
- No se reciben alarmas ni diagnósticos, lo que significa que no se muestra el estado de diagnóstico en la vista online
- No hay asignación de parámetros del módulo o submódulo

Ajuste de las propiedades en tiempo real

STEP 7 calcula la carga de comunicación y, por consiguiente, los tiempos de actualización resultantes. Hay que introducir el número de controladores IO ajenos al proyecto en el proyecto en el que la interfaz PROFINET del Shared Device está asignada al controlador IO, de modo que sea posible realizar un cálculo con las configuraciones del Shared Device.

El número máximo posible de controladores IO para el Shared Device depende del dispositivo. Dicho número está guardado en el archivo GSD del Shared Device.

Es posible ajustar un tiempo de ciclo de emisión muy corto (de 1 ms como mínimo) con una CPU S7-1200 como controlador IO. El tiempo de ciclo de emisión puede ser más corto que el mínimo soportado por el Shared Device. En este caso, el controlador IO hace funcionar el Shared Device con un tiempo de ciclo de emisión que soporte el primero (adaptación del tiempo de ciclo de emisión).

Ejemplo: Una CPU soporta tiempos de ciclo de emisión a partir de 1 ms. Un dispositivo IO configurado soporta tiempos de ciclo de emisión a partir de 1,25 ms; otro dispositivo IO soporta tiempos de ciclo de emisión a partir de 1 ms. En este caso, existe la posibilidad de ajustar el tiempo de ciclo de emisión corto, es decir, de 1 ms, para la CPU. La CPU hace funcionar el dispositivo IO "lento" con el tiempo de ciclo de emisión de 1,25 ms.

Reglas de configuración

- Los controladores IO que utilizan el Shared Device se crean en proyectos distintos. En cada proyecto hay que asegurarse de que el Shared Device se configure por igual en cada estación. Solo un controlador IO puede tener acceso completo a un submódulo a la vez. Las incoherencias en la configuración provocan fallos en el Shared Device.
- Las direcciones E/S de un módulo o submódulo solo pueden editarse si este está asignado al controlador IO en el mismo proyecto.
- El Shared Device debe tener los mismos parámetros IP y el mismo nombre de dispositivo en cada proyecto.
- El tiempo de ciclo de emisión debe ser idéntico para todos los controladores IO que tienen acceso al Shared Device.
- La ID de subred S7 de la subred a la que está conectado el Shared Device debe ser idéntica en todos los proyectos.
- Las funciones siguientes solo están disponibles si la interfaz PROFINET del Shared Device está asignada al controlador IO:
  - Arranque priorizado
  - Asignación de parámetros de las propiedades del puerto

Condiciones límite

Las siguientes condiciones límite surgen porque una configuración de Shared Device está distribuida entre varios proyectos:

- Las direcciones de módulos o submódulos que no están asignados a este controlador IO faltan en la vista general de direcciones de cada controlador IO que tiene acceso a un Shared Device.
- Los módulos o submódulos que no están asignados no se tienen en cuenta a la hora de calcular los límites de la configuración para el Shared Device durante la comprobación de coherencia. Por este motivo hay que comprobar personalmente que no se excede el número máximo de submódulos ni el número máximo de datos E/S cíclicos para el Shared Device. Encontrará información sobre las cantidades máximas en la documentación de los dispositivos utilizados.
- Los errores en la configuración, como la asignación de un módulo o submódulo a varios controladores IO, no se detectan en STEP 7.
- Las CPUs que tienen cargada la configuración de un Shared Device no tienen información acerca de si el dispositivo IO es un Shared Device. Los módulos o submódulos que están asignados a otros controladores IO y, por tanto, a otras CPUs, faltan en la configuración cargada. Por este motivo, dichos módulos o submódulos no se muestran ni en el servidor web de la CPU ni en la pantalla de la CPU.

11.5.14.2 Ejemplo: configurar un Shared Device (configuración GSD)

El presente ejemplo describe cómo configurar un sistema de periferia descentralizada en forma de Shared Device con STEP 7 V13 SP1 o superior.

Es posible realizar una configuración "descentralizada" con diferentes herramientas de ingeniería para distintas familias de controladores IO. El procedimiento descrito a continuación

se basa en STEP 7 a partir de V13 SP1 y se limita a la configuración con dos controladores IO de la serie S7-1200 que comparten un Shared Device.

En el ejemplo se crean dos proyectos con un controlador IO en cada uno:

- Controlador1
- Controlador2

Hay que crear el Shared Device en ambos proyectos, aunque físicamente sea el mismo dispositivo IO.

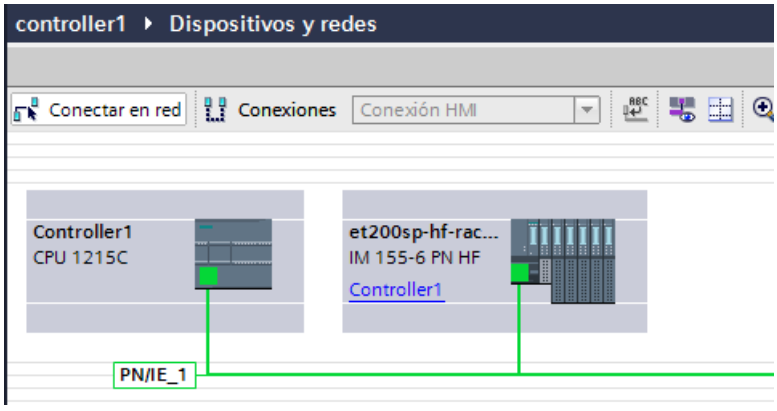
Requisitos

- STEP 7 V13 SP1 o superior
- El dispositivo IO soporta la funcionalidad de Shared Device (por ejemplo ET 200SP IM 155-6 PN HF V3.1).
- Hay instalado un archivo GSD para configurar el dispositivo IO como Shared Device.

Procedimiento: Creación del proyecto 1

Para crear el primer proyecto con un Shared Device, proceda del siguiente modo:

1. Inicie STEP 7.
2. Cree un proyecto nuevo con el nombre "Controlador1".
3. Inserte una CPU 1215C del catálogo de hardware en la vista de redes. Póngale por nombre "Controlador1".
4. Inserte un dispositivo IO con la función "Shared Device" (por ejemplo un ET 200SP) del catálogo de hardware (catálogo de hardware: Otros dispositivos de campo > PROFINET IO > IO).
5. Asigne el controlador IO "Controlador1" al dispositivo IO.

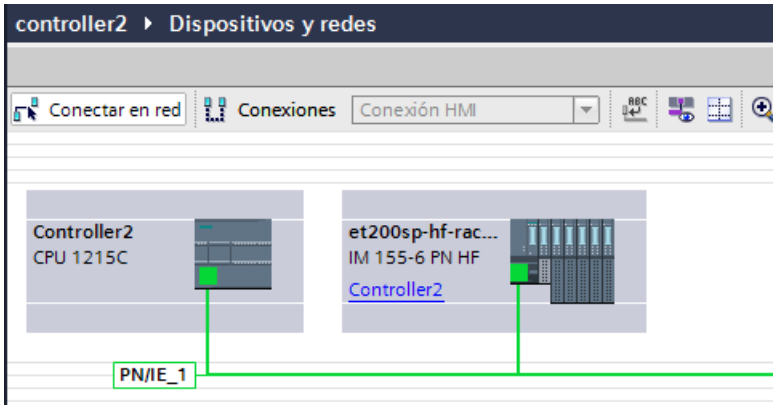


6. Haga doble clic en el dispositivo IO e inserte todos los módulos y submódulos necesarios del catálogo de hardware en la tabla de la vista general de dispositivos.
7. Asigne los parámetros de los módulos.
8. Guarde el proyecto.

Procedimiento: Creación del proyecto 2

Para crear el segundo proyecto con un Shared Device, proceda del siguiente modo:

1. Inicie STEP 7 de nuevo.
   Se abre una instancia nueva de STEP 7.
2. En la instancia nueva, cree un proyecto con el nombre "Controlador2".
3. Inserte una CPU 1215C en la vista de redes. Póngale por nombre "Controlador2".
4. Copie el dispositivo IO del proyecto "Controlador1" e insértelo en la vista de redes del proyecto "Controlador2".
5. Asigne el controlador IO "Controlador2" al dispositivo IO.



6. Guarde el proyecto.

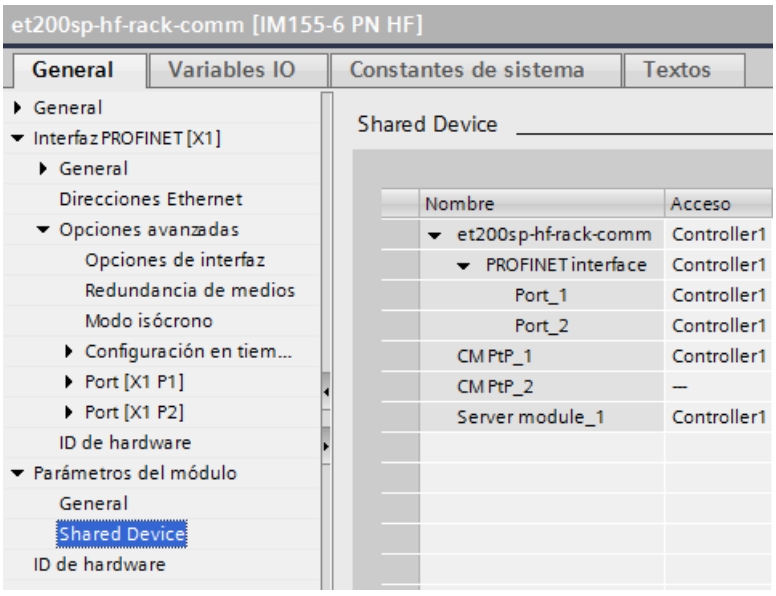
Ahora, ambos proyectos tienen un dispositivo IO con la misma estructura que debe configurarse en el paso siguiente para los diferentes tipos de acceso al controlador IO.

Procedimiento: Configuración del acceso al Shared Device

Los módulos y submódulos que se insertan en el Shared Device se asignan automáticamente a la CPU local. Para cambiar la asignación, proceda del siguiente modo:

1. Seleccione el módulo de interfaz en la vista de redes o dispositivos del proyecto "Controlador1".
2. Seleccione el área "Shared Device" en la ventana de inspección.
   Una tabla muestra qué CPU tiene acceso al respectivo módulo o submódulo para todos los módulos configurados. El ajuste predeterminado es que la CPU local tiene acceso a todos los módulos y submódulos.

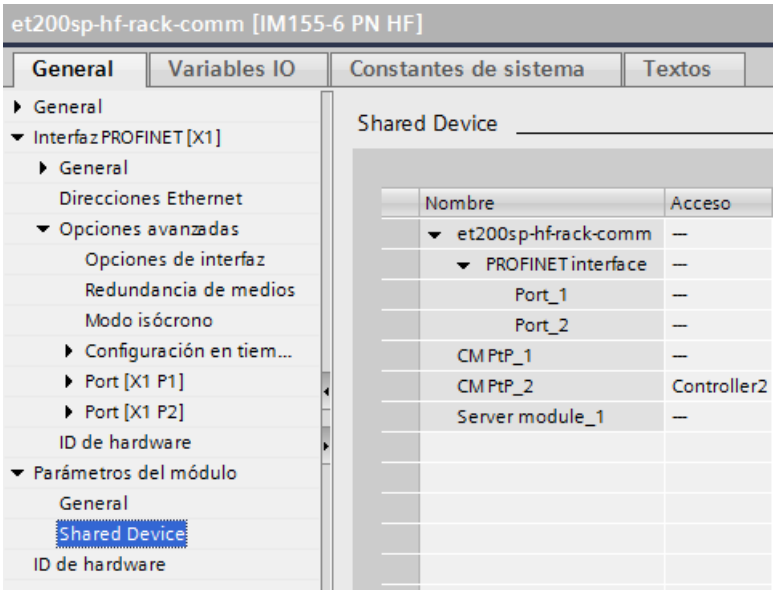
3. Deje el ajuste "Controlador1" en todos los módulos y submódulos que deben permanecer en el rango de direcciones de la CPU local.
   Seleccione el ajuste "---" en todos los módulos y submódulos que deben encontrarse en el rango de direcciones de la CPU del proyecto "Controlador2" (Controlador2). Es decir, un controlador IO que está fuera del proyecto debe tener acceso al módulo o submódulo.



4. Seleccione el módulo de interfaz en la vista de redes o dispositivos del proyecto "Controlador2".

5. Seleccione el área "Shared Device" en la ventana de inspección.
   Una tabla muestra qué CPU tiene acceso al respectivo módulo o submódulo para todos los módulos configurados.

6. Seleccione el ajuste "---" en todos los módulos y submódulos que deben encontrarse en el rango de direcciones de la CPU del proyecto "Controlador1" (Controlador1).



7. Finalmente, compruebe que los ajustes de acceso sean "complementarios" para cada módulo o submódulo de los dos proyectos. Esto significa que si la CPU local tiene acceso a un proyecto, la opción "---" debe estar ajustada en el otro proyecto y viceversa.
   Nota: la opción "---" para la interfaz PROFINET y, en consecuencia, para los puertos, convierte los parámetros asociados en solo lectura y no pueden modificarse. Los parámetros de la interfaz PROFINET y los parámetros de puerto solo pueden editarse en el proyecto en el que la interfaz PROFINET está asignada a la CPU local. Los puertos pueden interconectarse en ambos proyectos con independencia de ello.
8. Compruebe que estén ajustados los mismos parámetros de dirección IP y el mismo nombre de dispositivo para el Shared Device en todos los proyectos.
   Asegúrese de que esté ajustada la misma ID de subred S7 en todos los proyectos para la subred a la que está conectado el Shared Device (propiedades de subred, área "General" en la ventana de inspección).

Nota

Si realiza cambios en el Shared Device: haga los mismos cambios en cada proyecto para el Shared Device. Asegúrese de que solo un controlador IO tiene acceso a un módulo o submódulo.

Procedimiento: Adaptación de los ajustes de tiempo real

Para asegurarse de que todos los controladores IO y Shared Devices funcionan con el tiempo de ciclo de emisión adecuado y que los tiempos de actualización se han calculado correctamente según la carga de comunicación, es necesario adaptar y comprobar los ajustes siguientes:

1. Seleccione el proyecto cuyos controladores IO tengan acceso a la interfaz PROFINET y los puertos del Shared Device.
2. Seleccione el módulo de interfaz del Shared Device en la vista de redes.
3. En la ventana de inspección, navegue hasta el área "Interfaz PROFINET > Opciones avanzadas > Configuración en tiempo real > Ciclo IO".
4. En el área "Shared Device", ajuste el número de controladores IO que estén fuera del proyecto. El número máximo depende del dispositivo IO (especificación en el archivo GSD).
5. Hay que ajustar el mismo tiempo de ciclo de emisión para cada controlador IO que tenga acceso a los módulos y submódulos del Shared Device:

- Si se configura el controlador IO con STEP 7 (TIA Portal):
  - Abra el proyecto correspondiente.
  - Seleccione la interfaz PROFINET del controlador IO.
  - Seleccione el área "Opciones avanzadas > Configuración en tiempo real > Comunicación IO" de la ventana de inspección y ajuste el tiempo de ciclo de emisión compartido.
- Si se configura el controlador IO con otra herramienta de ingeniería:
  - Seleccione la interfaz PROFINET del Shared Device en STEP 7 (TIA Portal) y lea el tiempo de ciclo de emisión en el Shared Device (área "Opciones avanzadas > Configuración en tiempo real").
  - Introduzca el tiempo de ciclo de emisión leído en la herramienta de ingeniería.

Nota

Si se configuran todos los controladores IO que tienen acceso al Shared Device en STEP 7 (TIA Portal), es posible ajustar tiempos de ciclo de emisión más cortos en el controlador IO que los que soporta el Shared Device (adaptación del tiempo de ciclo de emisión).

Compilación y carga

Hay que compilar las configuraciones de los diferentes controladores IO y cargarlas seguidamente en las CPUs una tras otra.

Debido a la configuración descentralizada con proyectos separados, STEP 7 no emite errores de coherencia en caso de que la asignación de parámetros de acceso sea incorrecta. He aquí ejemplos de asignación incorrecta de parámetros de acceso:

- Varios controladores IO tienen acceso al mismo módulo
- Los parámetros de dirección IP y los tiempos de ciclo de emisión no son idénticos

Estos errores no se muestran hasta que el controlador funciona y se muestran como errores de configuración.

11.5.14.3 Ejemplo: Configuración de un I-device como un Shared Device

El presente ejemplo describe cómo configurar un S7-1200 como un I-device con la versión V13 SP1 o superior de STEP 7 y utilizarlo a continuación en dos proyectos como Shared Device.

Es posible realizar una configuración "descentralizada" con diferentes herramientas de ingeniería para distintas familias de controladores IO. El procedimiento descrito a continuación se basa en STEP 7 V13 SP1 y se limita a una configuración con dos controladores IO de la familia S7-1200 que comparten las áreas de transferencia de un I-device como Shared Device. El I-device es una CPU 1215C.

En el ejemplo se crean tres proyectos con un controlador IO en cada uno:

- I-device S7-1200
- Controlador1
- Controlador2

El proyecto I-device S7-1200 se utiliza para configurar el I-device. La variante PROFINET GSD del I-device S7-1200 se utiliza en los proyectos Controlador1 y Controlador2 para asignar las áreas de transferencia en el respectivo controlador IO de nivel superior.

Concepto de Shared I-device

El concepto de Shared I-device requiere como mínimo tres proyectos separados:

- Proyecto I-device: se configura y programa un I-device para llevar a cabo una tarea de automatización concreta. Las áreas de transferencia se definen como interfaz E/S para los controladores de nivel superior y se asignan a diferentes controladores IO. Para la conexión de controladores IO de nivel superior se proporciona un archivo PROFINET GSD y se utilizan las áreas de transferencia para acceder al I-device.
- Controladores que comparten el I-device (dos proyectos): el I-device se utiliza como variante PROFINET GSD durante la configuración del sistema PROFINET IO y, durante este proceso, se especifican las direcciones E/S con las que los controladores IO acceden a las áreas de transferencia.

Nota

Si se configura un S7-1200 como I-device, el tamaño máximo de un área de transferencia es de 1024 bytes de entrada o salida. Hay posibles factores de restricción en función de las E/S locales y de las limitaciones de espacio de direccionamiento en el controlador.

I-device

Los parámetros siguientes se asignan a una CPU S7-1200 como I-device:

- Periferia centralizada y descentralizada
- Áreas de transferencia deseadas
- Número de controladores IO que tienen acceso a este I-device (siempre mayor que 1 para un Shared Device)

Nota

El I-device se configura sin un controlador IO de nivel superior. Como consecuencia, solo es posible utilizar las direcciones E/S locales del área de transferencia (equivale a la "Dirección del I-device") para crear el programa de usuario y editar las direcciones desde el área de transferencia. El I-device se descarga en la CPU S7-1200 completamente configurado, a excepción de la conexión con el controlador IO de nivel superior.

Un archivo PROFINET GSD se exporta desde la configuración del I-device.

Controladores que comparten el I-device

Hay que instalar el archivo PROFINET GSD creado a partir de la configuración del I-device en todos los sistemas de ingeniería que se utilizan para configurar un sistema PROFINET IO con este Shared I-device. Si se configuran todos los usos de este I-device con STEP 7 V13 SP1 bastará con instalar el archivo GSD en STEP 7.

El I-device se configura como variante GSD en el sistema PROFINET IO de los proyectos correspondientes. En STEP 7 V13 SP1 encontrará este I-device en "Otros dispositivos de campo > PROFINET IO > PLCs y CPs" después de la instalación.

En cada uno de los proyectos se asignan áreas de transferencia únicamente a los controladores IO de nivel superior (ajuste predeterminado: todos). Las demás áreas de transferencia se ponen a "---" (no asignadas). De este modo, el controlador IO local no puede acceder a esta área de transferencia, de modo que puede asignarse a otro controlador IO de otro proyecto.

Requisitos

- STEP 7 V13 SP1 o superior
- El dispositivo IO soporta la funcionalidad de Shared Device (por ejemplo ET 200SP IM 155-6 PN HF V3.1).
- Hay instalado un archivo GSD para configurar el dispositivo IO como Shared Device.

Procedimiento: Creación del proyecto I-device S7-1200

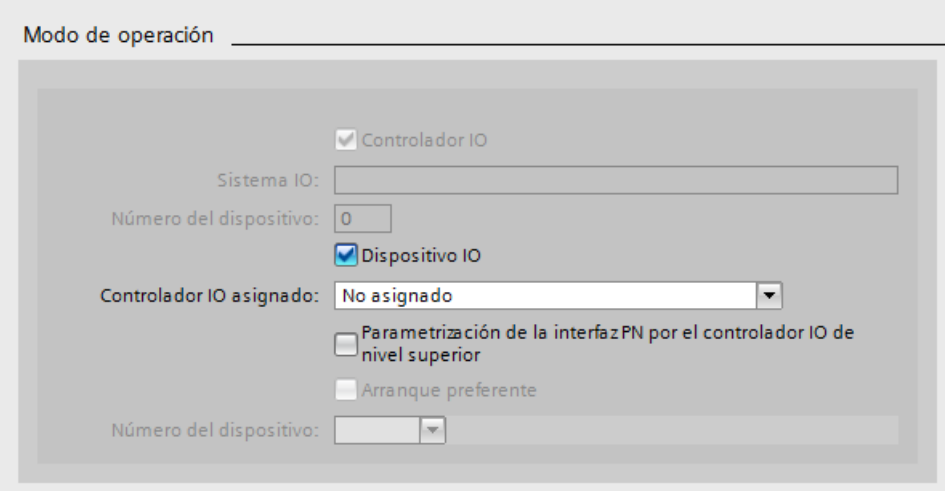
Para crear el proyecto con un Shared I-device, proceda del siguiente modo:

1. Inicie STEP 7.
2. Cree un proyecto nuevo con el nombre "I-device S7-1200".
3. Inserte una CPU 1215C del catálogo de hardware en la vista de redes. Póngale el nombre "I-device S7-1200".



4. Haga doble clic en el dispositivo IO y configure todos los módulos y submódulos necesarios.

5. Asigne los parámetros de los módulos. Concretamente, deben configurarse los ajustes siguientes para la CPU en el área de la interfaz PROFINET [X1]:
   – Active la opción "Dispositivo IO" en el área "Modo de operación".

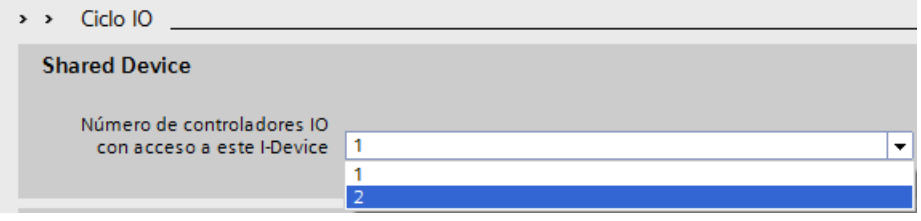


   – Configure las áreas de transferencia en el área "Modo de operación" > "Configuración del I-device". La columna "Dirección en el controlador IO" queda vacía porque no hay ningún controlador IO asignado.

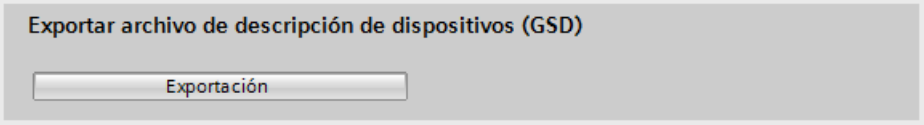


   Nota: Para cambiar un área de entrada a un área de salida y viceversa, hay que navegar hasta el área de la correspondiente área de transferencia.

   – Seleccione el número de controladores IO (como mínimo dos) que accederán al Shared I-device durante el funcionamiento ("Modo de operación" > área "Configuración en tiempo real" > área "Shared Device").



6. Guarde el proyecto.
7. Haga clic en el botón "Exportar" ("Modo" > área "Configuración del I-device", sección "Exportar archivo de descripción de estación general (GSD)"). Si no se cambia el nombre en el cuadro de diálogo Exportar, el archivo GSD utiliza un nombre de formato asignado (por ejemplo "GSDML-V2.31-#Siemens-PreConf_S7-1200-I-Device-20130925-123456").



Procedimiento: Creación del proyecto Controlador1

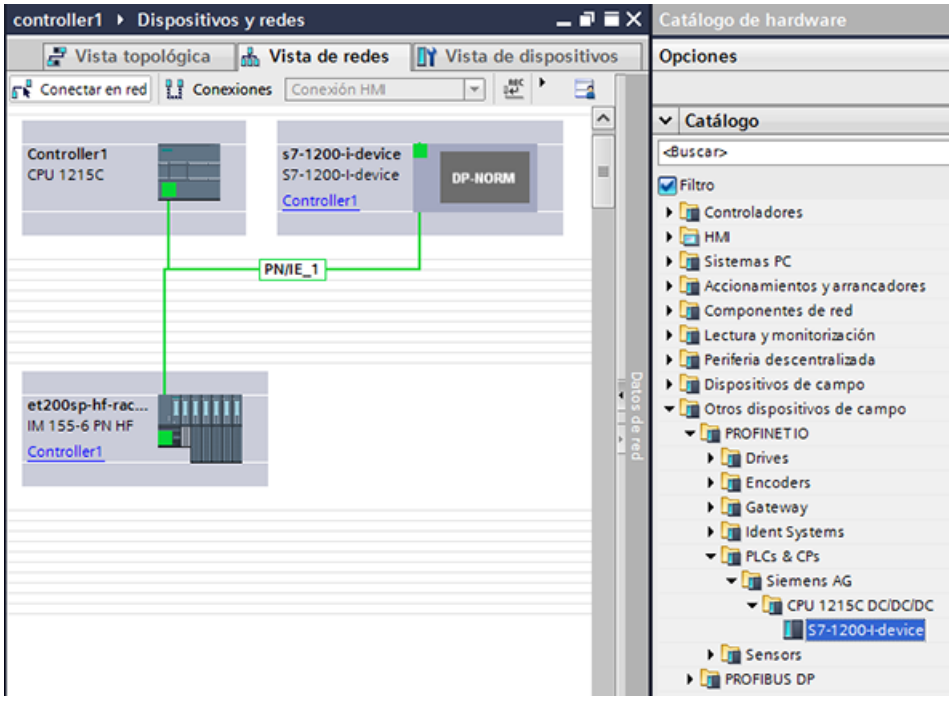
Para crear el primer proyecto con un Shared I-device, proceda del siguiente modo:

1. Inicie STEP 7.
2. Instale el archivo PROFINET GSD procedente de la exportación de la CPU I-device (I-device S7-1200).

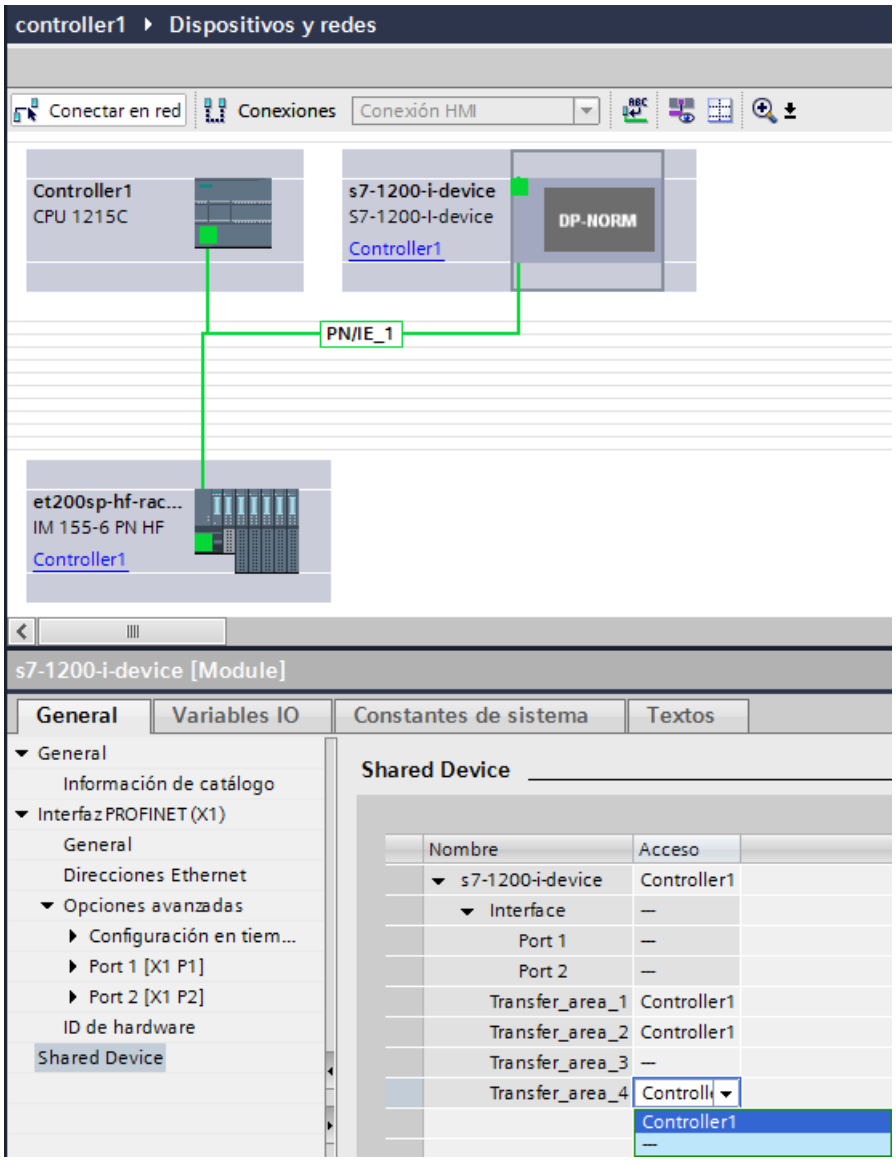


3. Cree un proyecto nuevo con el nombre "Controlador1".
4. Inserte la CPU 1215C en la vista de redes. El nombre de la CPU debería ser "Controlador1".
5. Inserte el I-device desde el catálogo de hardware (catálogo de hardware: Otros dispositivos de campo > PROFINET IO > PLCs y CPs).

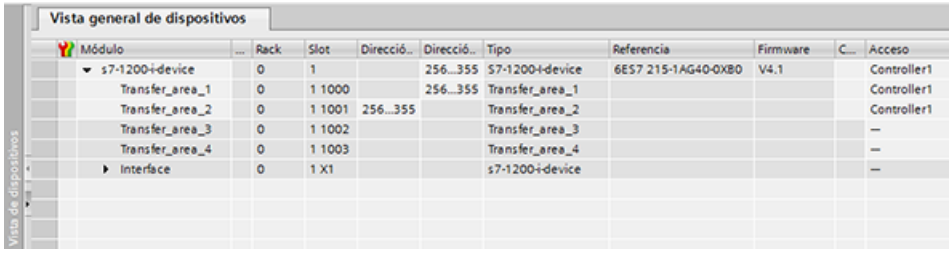
6. Asigne el controlador IO "Controlador1" al I-device.



7. Seleccione el área "Shared Device" en las propiedades del I-device:
   - En la tabla, todas las áreas de transferencia y la interfaz PROFINET se asignan al controlador IO local (Controlador1).
   - Defina las áreas de transferencia a las que la CPU Controlador1 **no** debe tener acceso. Seleccione la entrada "---" para dichas áreas. Estas áreas de transferencia están previstas para el Controlador2.



8. Las direcciones pueden adaptarse desde la vista de dispositivos del controlador IO en la vista general de dispositivos. Para abrir la vista general de dispositivos haga doble clic en el I-device.



9. Guarde el proyecto.

Procedimiento: Creación del proyecto Controlador2

Para crear el segundo proyecto con un Shared Device, proceda del siguiente modo:

1. Inicie STEP 7 de nuevo.
   Se abre una instancia nueva de STEP 7.
2. En la instancia nueva, cree un proyecto con el nombre "Controlador2".
3. Inserte la CPU 1215C en la vista de redes. Póngale el nombre "Controlador2".
4. Inserte el I-device desde el catálogo de hardware (catálogo de hardware: Otros dispositivos de campo > PROFINET IO > PLCs y CPs).
5. Asigne el controlador IO "Controlador2" al I-device.
6. Adapte el acceso a las áreas de transferencia igual que en el proyecto Controlador 1. Asegúrese de que no haya asignaciones duplicadas.
7. Adapte los parámetros de la subred y la interfaz PROFINET. Puesto que el Shared I-device es el mismo dispositivo en diferentes proyectos, estos datos deben coincidir.
8. Guarde el proyecto.

Ahora, ambos proyectos tienen un Shared I-device con la misma configuración. El acceso del controlador IO y los parámetros de la interfaz PROFINET deben comprobarse aún en los distintos proyectos durante el paso siguiente.

Resumen: asignar parámetros para acceder al Shared Device

Las áreas de transferencia se asignan automáticamente al controlador IO local. Para cambiar la asignación, proceda del siguiente modo:

1. Haga clic en el dispositivo "I-device S7-1200" en la vista de redes del proyecto "Controlador1" y seleccione el área "Shared Device".
2. Una tabla muestra qué CPU tiene acceso a cada una de las áreas de transferencia configuradas. El ajuste predeterminado es que la CPU local tiene acceso a todos los módulos y submódulos.

3. Deje el ajuste "Controlador1" en todas las áreas de transferencia que deben permanecer en el rango de direcciones de la CPU local.
   Seleccione el ajuste "---" en todas las áreas de transferencia que deben encontrarse en el rango de direcciones de la CPU "Controlador2" del proyecto "Controlador2". Es decir, un controlador IO que está fuera del proyecto debe tener acceso al área de transferencia.
4. Repita el mismo procedimiento para los proyectos restantes.
5. Finalmente, compruebe que los ajustes de acceso sean "complementarios" para cada módulo o submódulo de los dos proyectos. Esto significa que si la CPU local tiene acceso a un proyecto, la opción "---" debe estar ajustada en el otro proyecto y viceversa.
   Nota: la opción "---" para la interfaz PROFINET y, en consecuencia, para los puertos, convierte los parámetros asociados en solo lectura y no pueden modificarse. Los parámetros de la interfaz PROFINET y los parámetros de puerto solo pueden editarse en el proyecto en el que la interfaz PROFINET está asignada a la CPU local. Los puertos pueden interconectarse en ambos proyectos con independencia de ello.
6. Compruebe que estén ajustados los mismos parámetros de dirección IP y el mismo nombre de dispositivo para el Shared Device en todos los proyectos.
   Asegúrese de que esté ajustada la misma ID de subred S7 en todos los proyectos para la subred a la que está conectado el Shared Device (propiedades de subred, área "General" en la ventana de inspección).

Nota

Si realiza cambios en el I-device (por ejemplo en el número o la longitud de las áreas de transferencia), exporte el I-device de nuevo como archivo GSD. Vuelva instalar el archivo GSD en cada proyecto que utilice el I-device como Shared Device. Asegúrese de que solo un controlador IO tiene acceso a un área de transferencia.

Nota

En caso de utilizar el S7-1200 como shared I-device y como controlador, hay que asegurarse de aumentar los tiempos de actualización de PROFINET IO y del I-device PROFINET con el fin de reducir el impacto en el rendimiento de la comunicación. El sistema es muy estable y funciona bien con un tiempo de actualización de 2 ms para un tiempo de I-device PROFINET individual y de 2 ms para un tiempo de PROFINET IO individual.

Los parámetros del "Ciclo IO" se especifican en el cuadro de diálogo de configuración "Propiedades" del I-device PROFINET o del dispositivo PROFINET IO. Encontrará más información en "Configurar el tiempo de ciclo IO" (Página 735).

Procedimiento: Adaptación de los ajustes de tiempo real

Para asegurarse de que todos los controladores IO y Shared Devices funcionan con el tiempo de ciclo de emisión adecuado y que los tiempos de actualización se han calculado correctamente según la carga de comunicación, es necesario adaptar y comprobar los ajustes siguientes:

1. Hay que ajustar el mismo tiempo de ciclo de emisión para cada controlador IO que tenga acceso a los módulos y submódulos del Shared Device:

- Si se configura el controlador IO con STEP 7 (TIA Portal), proceda del siguiente modo:
  - Abra el proyecto correspondiente.
  - Seleccione la interfaz PROFINET del controlador IO.
  - Seleccione el área "Opciones avanzadas > Configuración en tiempo real > Comunicación IO" de la ventana de inspección y ajuste el tiempo de ciclo de emisión compartido.
- Si se configura el controlador IO con otra herramienta de ingeniería, proceda del siguiente modo:
  - Seleccione la interfaz PROFINET del Shared Device en STEP 7 (TIA Portal) y lea el tiempo de ciclo de emisión en el Shared Device (área "Opciones avanzadas > Configuración en tiempo real").
  - Introduzca el tiempo de ciclo de emisión leído en la herramienta de ingeniería.

Nota

Si se configuran **todos** los controladores IO que tienen acceso al Shared I-device en STEP 7 (TIA Portal), es posible ajustar tiempos de ciclo de emisión más cortos en el controlador IO que los que soporta el Shared Device (adaptación del tiempo de ciclo de emisión).

Compilación y descarga

Hay que compilar las configuraciones de los diferentes controladores IO y descargarlas seguidamente en las CPUs una tras otra.

Debido a la configuración descentralizada con proyectos separados, STEP 7 no emite errores de coherencia en caso de que la asignación de parámetros de acceso sea incorrecta. He aquí ejemplos de asignación incorrecta de parámetros de acceso:

- Varios controladores IO tienen acceso al mismo módulo.
- Los parámetros de dirección IP y los tiempos de ciclo de emisión no son idénticos.

Estos errores no se muestran hasta que el controlador funciona y se muestran como errores de configuración.

11.5.15 Media Redundancy Protocol (MRP)

Con S7-1200 V4.5 y TIA Portal V17, las siguientes CPU soportan la funcionalidad MRP como administrador y como cliente MRP.

- CPU 1215C
- CPU 1217C
- CPU 1215FC

En calidad de "Cliente", la 1215/1217 debe funcionar en un anillo MRP enviando paquetes MRP a través de su interfaz y notificando pérdidas de conexión a un administrador de la red.

En calidad de "Administrador", la 1215/1217 debe enviar paquetes MRP por la red, detectar puertos abiertos en el anillo, administrar puertos bloqueados y negociar estados de administrador con otros posibles administradores.

La familia de CPU S7-1200 tiene tres modelos (consulte más arriba) que soportan el protocolo MRP y los parámetros de configuración que se utilizan para inicializar la operación de cliente y administrador MRP. Estas CPU tienen dos puertos PN.

Con estas tres CPU es posible configurar un anillo MRP con S7-1200 como cliente o como administrador. En calidad de administrador, la CPU utiliza frames de prueba para comprobar y asegurase de que no hay interrupciones en el anillo. En calidad de cliente, envía estos frames de prueba en lugar de realizar la comprobación ella misma. Esto permite utilizar S7-1200 con ambas funciones si fuera necesario.

Nota

S7-1200 V4.4 y TIA Portal V16 y anteriores

Con versiones V4.4 y anteriores, S7-1200 solo puede configurarse como cliente en una configuración de anillo MRP.

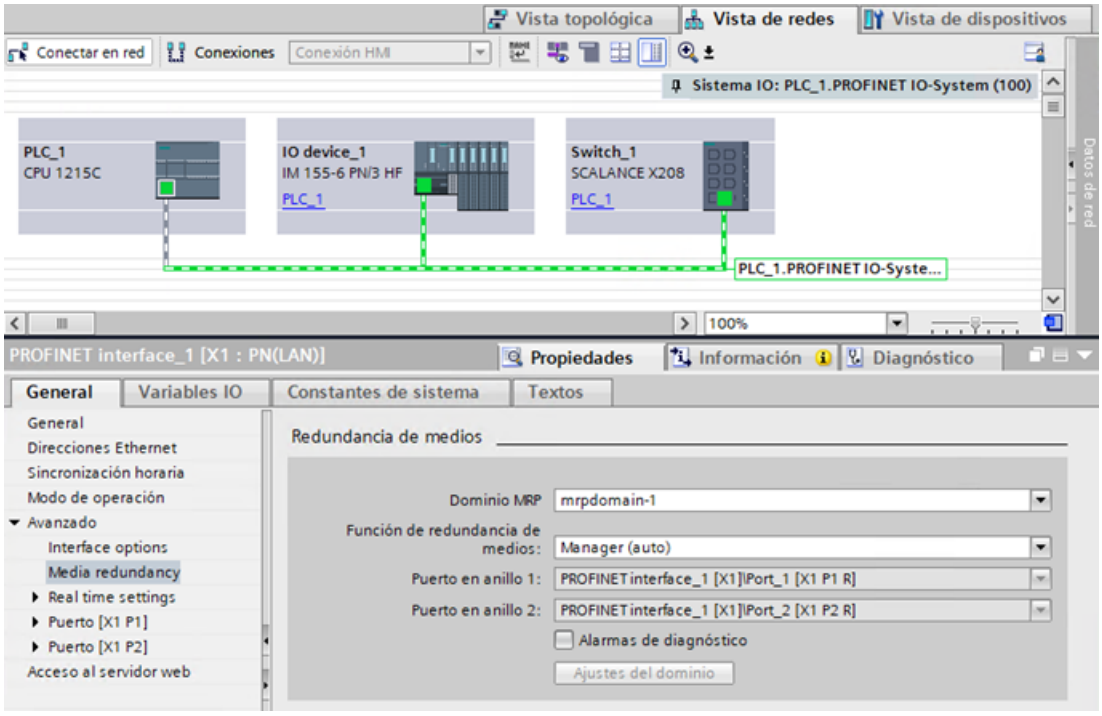


Figura 11-1 Configuración de redundancia de medios en TIA Portal

Función MRP Manager (auto) y Manager

S7-1200 (1215C/1217C) puede usarse con las funciones Manager y Manager (auto) en un anillo con redundancia de medios (MRP). MRP permite conectar los dispositivos en una configuración en anillo. El administrador MRP generalmente fuerza el flujo de datos en un sentido lineal bloqueando un puerto del anillo. Cuando esta configuración en anillo sufre una interrupción, el administrador la detecta y desbloquea el puerto, permitiendo que los datos fluyan en el otro sentido. MRP hace que la red permanezca operativa en caso de rotura de hilo o de fallo de un

dispositivo. La especificación MRP permite un máximo de 50 dispositivos en una configuración en anillo.

MRP Manager
El administrador MRP permite al PLC S7-1200 actuar como administrador de redundancia. Si se ha configurado para esta función, S7-1200 utiliza frames de prueba para comunicarse con los PLC cliente y garantizar que la conexión no se interrumpe dentro del anillo. Si se detecta una interrupción de la comunicación con un cliente, S7-1200, en calidad de administrador MRP, informa inmediatamente a los dispositivos cliente del anillo sobre el cambio a través de los puertos del anillo. TIA Portal solo permite definir un dispositivo como administrador MRP dentro de un lazo MRP.

Función MRP Manager (auto)
Cuando se asigna la función MRP Manager (auto) a varios dispositivos, estos negocian el estado de administrador entre ellos. Si el administrador MRP negociado se desconecta de la configuración, los dispositivos restantes definidos como Manager (auto) renegocian la función de administrador entre ellos hasta que se restaura la configuración original. Si el administrador original se recupera correctamente, volverán a negociar el estado de administrador y retoman la configuración original. Las configuraciones de administrador MRP de S7-1200 (1215C/1217C) solo pueden llevarse a cabo en versiones V4.5 o superiores.

Nota

Reconfiguración del anillo

La reconfiguración del anillo puede tardar hasta 200 ms. Por eso, para cada dispositivo debe definirse un tiempo de supervisión de respuesta de PROFINET superior a 200 ms.

Nota

El modo MRP Manager (auto) es el ajuste predeterminado

Si no hay un proyecto, el dispositivo V4.5 estará por defecto en modo MRP Manager (auto). Es importante conocer este hecho si se enchufa un dispositivo listo para usar en una topología que no sea en anillo y detecta frames de prueba en su red.

El S7-1200 (CPU 1214C, 1212C y 1211C) no puede habilitar las funciones MRP Manager y Manager (auto). Las opciones de menú en TIA Portal no están disponibles para estas CPU.

S7-1200 no soporta MRPD porque no tiene capacidad IRT.

Las alarmas de diagnóstico pueden activarse y desactivarse en la ventana de configuración de la redundancia de medios de TIA portal para permitir al administrador y a los clientes facilitar notificaciones de cambios relevantes en MRP, tales como:

- Un dispositivo vecino del puerto en anillo no soporta MRP.
- Un puerto del anillo está conectado a un vecino con un puerto que no es del anillo.
- Un puerto del anillo está conectado a otro dominio MRP.
- (Solo administrador) Alarmas de diagnóstico de interrupción/retorno cuando el anillo MRP sufre una interrupción y cuando se recupera la configuración original.

Consulte también

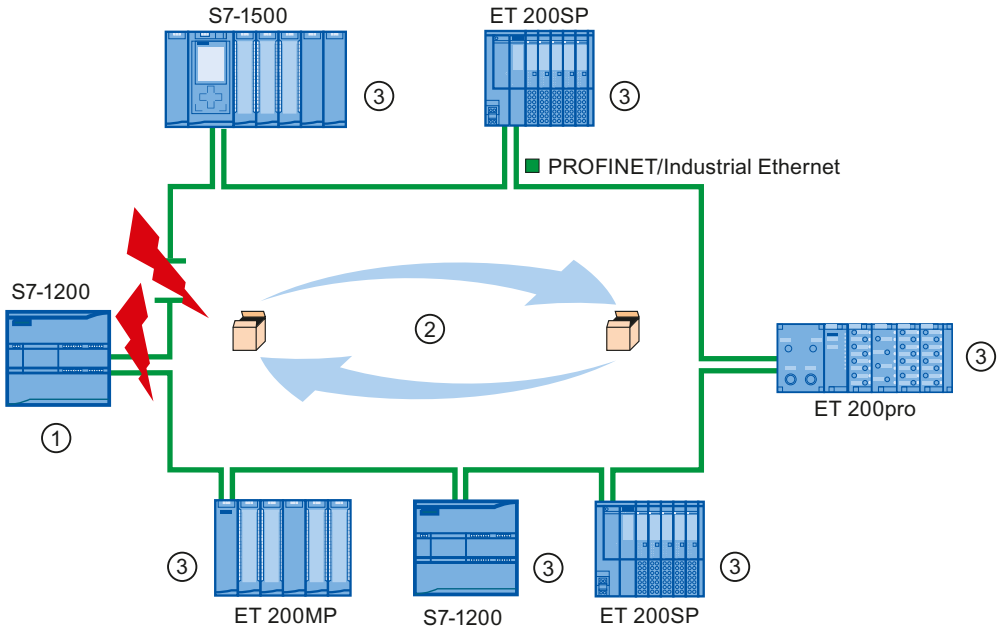
Nuevas funciones (Página 35)

11.5.15.1 Redundancia de medios con topologías en anillo

Con el fin de aumentar la disponibilidad de una red Industrial Ethernet con topologías de bus lineal ópticas o eléctricas, existe la posibilidad de convertir una topología de bus lineal en una topología en anillo uniendo los extremos.

Los dispositivos de una topología en anillo pueden ser dispositivos IO, controladores IO, switches externos y/o los switches integrados de los módulos de comunicación.

Para configurar una topología en anillo con redundancia de medios hay que unir los dos extremos libres de una topología de bus linean en un dispositivo. Para cerrar la topología de bus lineal y formar un anillo se necesitan dos puertos (puertos en anillo) de un dispositivo del anillo. Seguidamente, un dispositivo del anillo creado asume la función de administrador MRP. Todos los demás dispositivos del anillo son clientes MRP.



① Administrador MRP (puede ser una CPU S7-1500, una CPU S7-1200 1215C, una CPU S7-1200 1215FC o una CPU S7-1200 1217C)

② Frames de prueba

③ Clientes MRP

Los puertos en anillo de un dispositivo son los que establecen la conexión con los dos dispositivos vecinos en la topología en anillo. Los puertos en anillo se seleccionan y ajustan en la configuración del dispositivo en cuestión (dado el caso está predefinido).

Funcionamiento de la redundancia de medios en una topología en anillo

El recorrido de los datos entre los diferentes dispositivos se reconfigura automáticamente si el anillo queda interrumpido en algún punto. Los dispositivos vuelven a estar disponibles tras la reconfiguración.

En el administrador MRP, uno de los dos puertos en anillo se bloquea en el funcionamiento de red continuo para la comunicación normal, de modo que los frames de datos no circulan. El administrador MRP vigila el anillo en busca de interrupciones. Lo hace enviando frames de

prueba desde el puerto en anillo 1 y el puerto en anillo 2. Los frames de prueba circulan por el anillo en ambos sentidos hasta que llegan al otro puerto en anillo del administrador MRP.

El anillo puede interrumpirse por la pérdida de la conexión entre dos dispositivos o por el fallo de un dispositivo del anillo.

Si los frames de prueba del administrador MRP dejan de llegar al otro puerto en anillo durante una interrupción del anillo, el administrador MRP conecta sus dos puertos en anillo. Esta ruta alternativa vuelve a establecer una conexión operativa entre todos los dispositivos restantes en forma de topología de bus lineal.

El tiempo que transcurre entre la interrupción del anillo y el restablecimiento de una topología lineal operativa se conoce como tiempo de reconfiguración.

En cuanto se elimina la interrupción, se recupera la ruta de transmisión original, los dos puertos en anillo del administrador MRP se desconectan y los clientes MRP son informados del cambio. Seguidamente, los clientes MRP vuelven a utilizar el recorrido original hacia los otros dispositivos.

Método de redundancia de medios

El método estándar de redundancia de medios en SIMATIC es el Media Redundancy Protocol (MRP) con un tiempo de reconfiguración típico de 200 ms. En un anillo pueden participar hasta 50 dispositivos.

Nota

Reconfiguración del anillo

La reconfiguración del anillo puede tardar hasta 200 ms. Para cada dispositivo debe ajustarse un tiempo de supervisión de respuesta de PROFINET superior a 200 ms.

11.5.15.2 Uso de Media Redundancy Protocol (MRP)

El proceso "MRP" funciona de conformidad con el Media Redundancy Protocol (MRP), que está especificado en IEC 61158 tipo 10 "PROFINET".

Requisitos

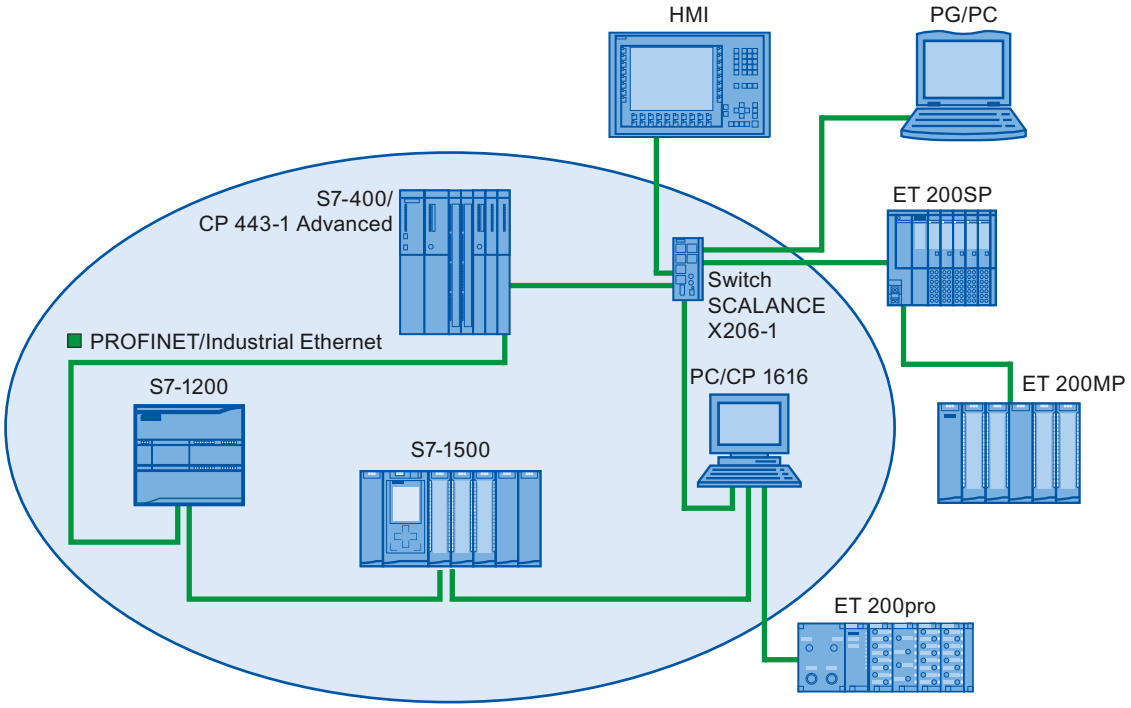
Para que MRP funcione correctamente deben cumplirse los requisitos siguientes:

- El anillo en el que debe usarse MRP debe estar formado únicamente por dispositivos que soporten esta función.
- "MRP" debe estar activado para todos los dispositivos del anillo.
- Todos los dispositivos deben estar interconectados con sus puertos en anillo.
- Debe haber disponible como mínimo un administrador MRP o una función "Manager".
- El anillo no debe contener más de 50 dispositivos. En caso contrario pueden darse tiempos de reconfiguración de 200 ms o más.
- Todos los puertos interlocutores dentro de los anillos deben tener ajustes idénticos.

Topología

El esquema siguiente muestra una posible topología para dispositivos de un anillo con MRP. Los dispositivos que están dentro del óvalo ensombrecido están en el dominio de redundancia.

Este es un ejemplo de una topología en anillo con MRP:



Las reglas siguientes rigen para una topología en anillo con redundancia de medios que utiliza MRP:

- Todos los dispositivos del anillo forman parte del mismo dominio de redundancia.
- Un dispositivo del anillo tiene la función de administrador MRP.
- Todos los demás dispositivos del anillo son clientes MRP.

Es posible conectar a la red dispositivos no aptos para MRP a través de puertos que no estén configurados como puertos en anillo. Solo puede hacerse con dispositivos que tienen más de dos puertos (p. ej. un switch SCALANCE X o un PC con un CP1616).

Condiciones límite

Existen las siguientes posibilidades de comunicación:

- MRP y RT: RT puede utilizarse junto con MRP.

  Nota

  La comunicación RT se interrumpe (fallo de estación) si el tiempo de reconfiguración del anillo es mayor que el tiempo de supervisión de respuesta seleccionado para el dispositivo IO. Hay que seleccionar un tiempo de supervisión de respuesta mayor que 200 ms para los dispositivos IO. Encontrará más información en el apartado inferior "Tiempo de supervisión de respuesta".

- MRP y TCP/IP (TSEND, HTTP, ...): la comunicación TCP/IP con MRP es posible porque los paquetes de datos perdidos se reenvían, dado el caso.
- MRP y arranque priorizado:
  - si se configura MRP en un anillo no es posible utilizar la función "arranque priorizado" en aplicaciones PROFINET para los dispositivos implicados.
  - Si desea utilizar la función "arranque priorizado" hay que desactivar MRP en la configuración (el dispositivo no puede formar parte del anillo).
- MRP en dispositivos PROFINET con más de dos puertos: si se utiliza un dispositivo PROFINET con más de dos puertos en un anillo, debe establecerse un límite de sincronización en los puertos que no están en el anillo. Cuando se ajusta el límite de sincronización se define un límite para un dominio Sync. No es posible enviar frames de sincronización transmitidos para sincronizar dispositivos dentro de un dominio Sync.

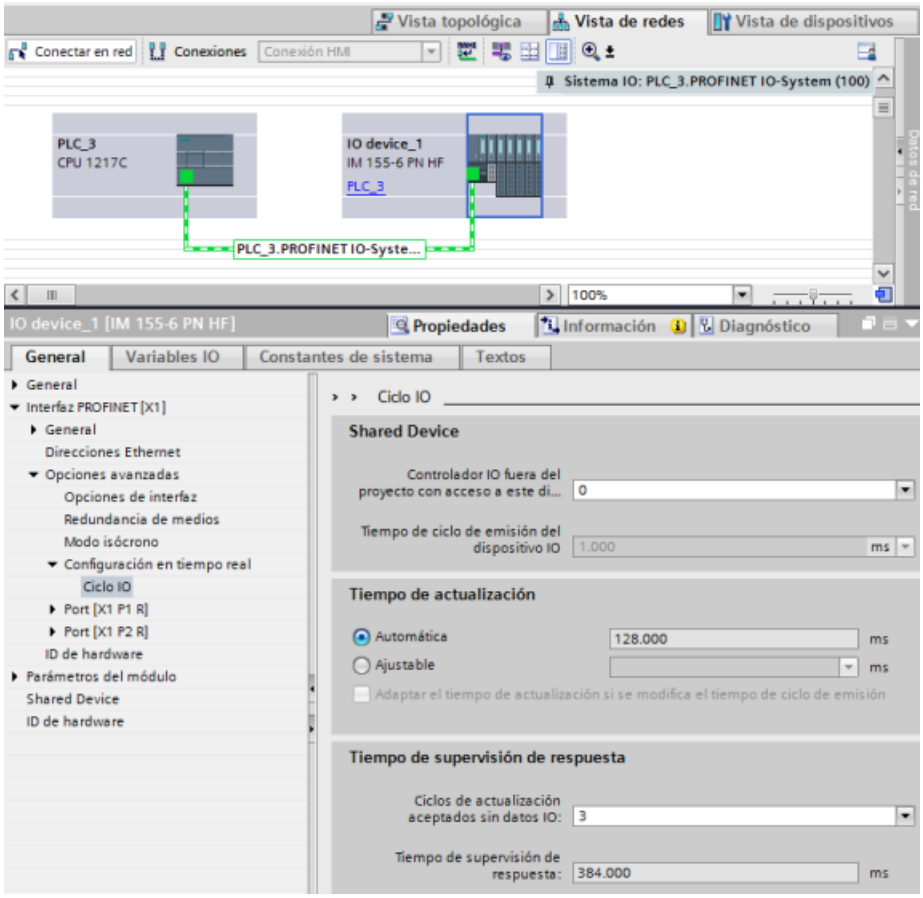
Tiempo de supervisión de respuesta

El tiempo de supervisión de respuesta de PROFINET es el intervalo de tiempo que un controlador IO o dispositivo IO deja pasar sin recibir datos IO. Si el controlador IO no suministra datos al dispositivo IO dentro del tiempo de supervisión de respuesta, el dispositivo detecta los frames que faltan y emite valores sustitutivos. Esto se notifica al controlador IO como fallo de estación.

El tiempo de supervisión de respuesta para dispositivos PROFINET IO puede configurarse. El tiempo de supervisión de respuesta no debe introducirse directamente, sino como "Número aceptado de ciclos de actualización cuando faltan datos IO". El tiempo de supervisión de respuesta resultante se calcula automáticamente a partir del número de ciclos de actualización.

Para asignar el tiempo de supervisión de respuesta, proceda del siguiente modo:

1. Seleccione la interfaz PROFINET del dispositivo IO en la vista de redes o de dispositivos.
2. En las propiedades de la interfaz navegue hasta: Opciones avanzadas > Configuración en tiempo real > Ciclo IO
3. Seleccione el número necesario de ciclos en la lista desplegable.



11.5.15.3 Configurar la redundancia de medios

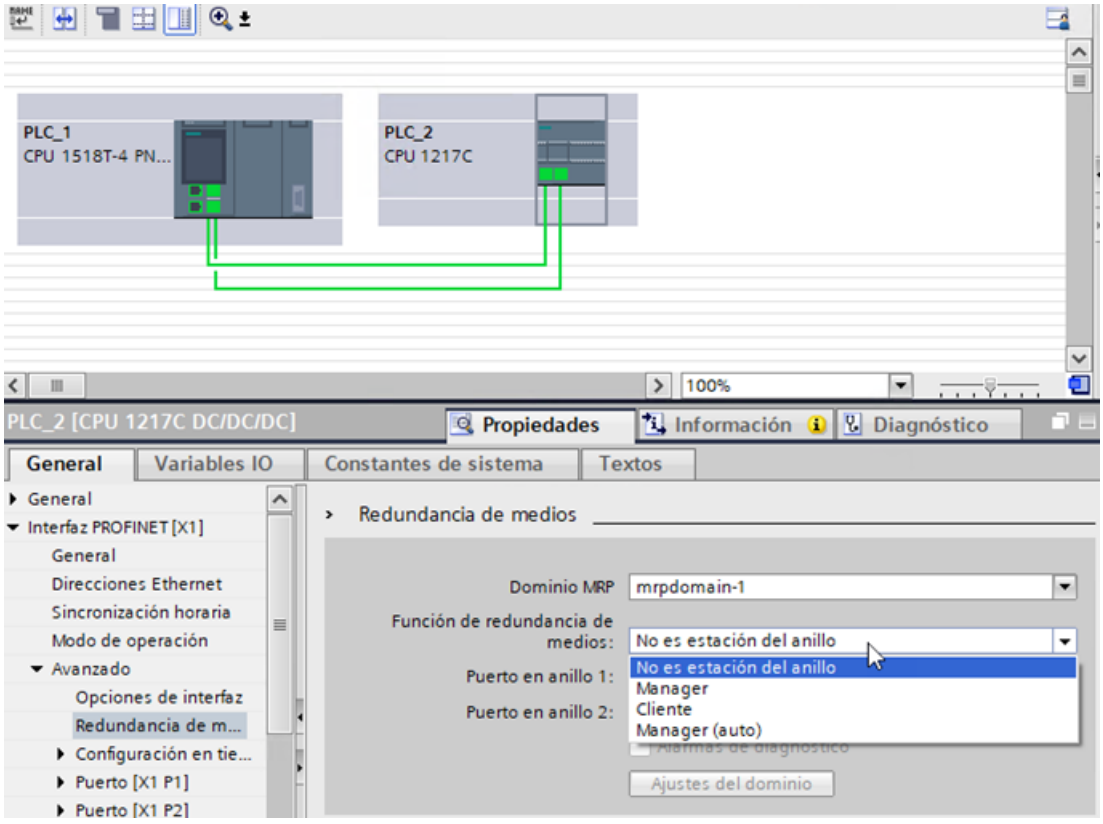
Todos los componentes de la aplicación deben ser compatibles con Media Redundancy Protocol (MRP).

Procedimiento

Para configurar la redundancia de medios, proceda del siguiente modo:

1. Cree un anillo interconectando los puertos de la forma conveniente (p. ej. en la vista topológica).
2. Seleccione un dispositivo PROFINET para el que desee configurar la redundancia de medios.

3. En la ventana de inspección, navegue hasta "Interfaz PROFINET [X1]" > "Opciones avanzadas" > "Redundancia de medios".



4. En "Función de redundancia de medios", asigne la función "Manager (Auto)", "Cliente", o "No es estación del anillo" al dispositivo.
   Cuando se configura un anillo en la vista topológica de TIA Portal, TIA Portal se encarga automáticamente de asignar la función de redundancia de medios. Si un dispositivo puede ser administrador, el TIA Portal ajusta la función de redundancia de medios en "Manager (Auto)". Para S7-1200, la función de redundancia de medios ahora se puede ajustar como "Cliente" o como "Manager".

   Nota

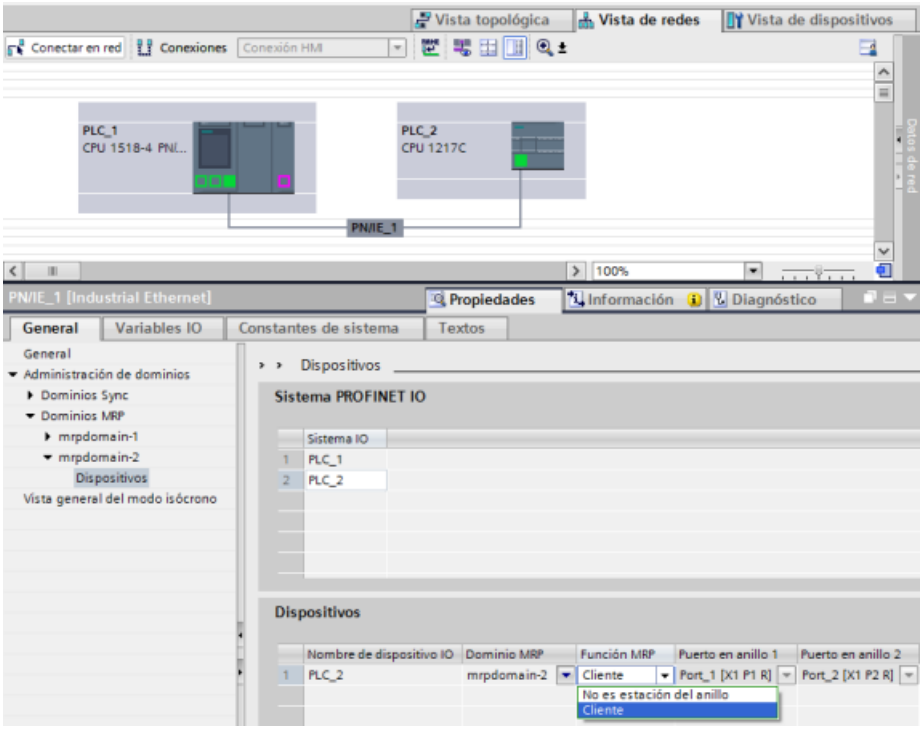
   Con V4.5 y TIA Portal V17 ahora es posible asignar las funciones MRP "Manager" o "Manager (Auto)" a la CPU S7-1200 1215C/1215FC/1217C.

5. Repita los pasos 2 a 4 para todos los dispositivos PROFINET del anillo.

O bien:

1. Seleccione el sistema PROFINET IO en la vista de redes.
2. Haga clic en el sistema PROFINET IO.

3. Navegue hasta el dispositivo del dominio MRP requerido en la ventana de inspección.



4. Asigne a los dispositivos PROFINET la función "Manager (Auto)", "Cliente" o "No es estación del anillo".

Opción de ajuste "Redundancia de medios": función MRP

Según sea el dispositivo utilizado están disponibles las funciones "Manager", "Manager (Auto)", "Cliente" y "No es estación del anillo".

Reglas:

- En un anillo solo puede haber un dispositivo que tenga la función "Manager". No puede haber más dispositivos con la función "Manager" o "Manager (Auto)". Todos los demás dispositivos del anillo solo pueden tener la función "Cliente". Los dispositivos que no estén en el anillo pueden tener la función "No es estación del anillo".
- Si en el anillo no hay ningún dispositivo con la función "Manager", debe haber al menos uno que tenga la función "Manager (Auto)". En un anillo puede haber un número cualquiera de dispositivos con las funciones "Cliente" y "Manager (Auto)".

Opción de ajuste "Redundancia de medios": puerto en anillo 1 y puerto en anillo 2

Seleccione uno a uno los puertos que desee configurar como puerto en anillo 1 o puerto en anillo 2. El cuadro de la lista desplegable muestra la selección de puertos posibles para cada tipo de dispositivo. Si los puertos están ajustados de fábrica, los campos no están disponibles.

Nota

En el S7-1200 no es necesario configurar los puertos en anillo, porque la CPU S7-1200 solo tiene dos puertos.

Alarmas de diagnóstico

Si en la CPU local deben visualizarse alarmas de diagnóstico relacionadas con el estado de MRP, seleccione la casilla de verificación "Alarmas de diagnóstico". Es posible configurar las siguientes alarmas de diagnóstico:

- Error de cableado o de puerto:
  La CPU genera alarmas de diagnóstico para los errores siguientes en los puertos en anillo:
  - Un vecino del puerto en anillo no soporta MRP.
  - Un puerto en anillo está conectado a un puerto que no es del anillo.
  - Un puerto en anillo está conectado al puerto en anillo de otro dominio MRP.
- Interrupción / retorno (solo administrador MRP):
  Si el anillo está interrumpido y se recupera la configuración original, la CPU genera alarmas de diagnóstico. Si estas dos alarmas se producen en un intervalo de 0,2 segundos, significa que el anillo está interrumpido.

Es posible responder a estos eventos en el programa de usuario programando la respuesta adecuada en el OB de alarma de diagnóstico (OB 82).

Nota

Dispositivos de terceros como administrador MRP

Para garantizar un funcionamiento correcto cuando se utiliza un dispositivo de terceros como administrador MRP, hay que asignar la función fija "Cliente" a todos los demás dispositivos del anillo antes de cerrarlo. En caso contrario, podrían darse frames de datos circulantes y fallos de red.

11.5.16 Routing S7

Desde la vista de redes de STEP 7 existe la posibilidad de crear una topología de comunicación compleja conectando dispositivos en diferentes subredes S7. Es posible conectar CPU y CP S7-300/S7-400 clásicos, así como las últimas versiones de CPU y CP S7, además de incluir estaciones HMI y PC como un servidor OPC.

Una vez haya decidido los dispositivos que deben comunicarse y haya realizado las conexiones necesarias en STEP 7, el sistema de ingeniería (ES) puede descargar las tablas de routing correspondientes en los diferentes routers S7 como parte de la configuración hardware. Tras

descargar las tablas de routing en los diferentes routers S7, el ES u otros interlocutores de la comunicación pueden comunicarse con cada uno de los dispositivos, aunque estos estén en subredes S7 distintas. Esto es posible porque las CPU y/o los CP actúan como routers S7. Las CPU y/o los CP reenvían peticiones de conexión entrantes al próximo router S7 hasta que la petición de conexión llega al dispositivo de destino y este establece la conexión S7.

La CPU utiliza el mecanismo de escribir registro para transferir las tablas de routing que necesitan los dispositivos CP en la base local. Las tablas de routing establecen la ruta de un dispositivo a otro en el momento de producirse una petición de conexión que incluye una ID de subred S7 remota. El dispositivo que recibe la petición de conexión consulta su tabla de routing, busca la estación más cercana en el recorrido hasta la subred S7 de destino y reenvía la petición de conexión. Finalmente, la petición de conexión llega al destino previsto y la respuesta recorre la ruta en sentido inverso.

Las CPU S7-1200 tienen una sola interfaz PN y un máximo de tres dispositivos CP conectados al bus de comunicación local. Por tanto, hay dos opciones de routing dentro de la estación S7-1200:

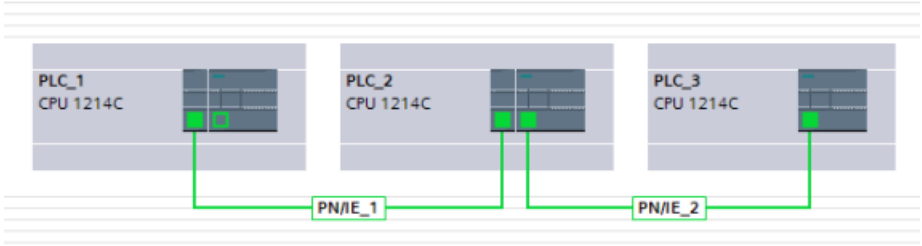
- Routing entre la CPU y un CP
- Routing de un CP a otro CP

Consulte los CP S7-1200 en el soporte de producto de Siemens Industry Online Support si desea más información sobre todos los CP S7-1200 que soportan la función de routing S7. El CP 1243-1 (https://support.industry.siemens.com/cs/es/es/view/584459) se utiliza como ejemplo de búsqueda de módulos CP con capacidades de routing S7.

11.5.16.1 Routing S7 entre interfaces de CPU y CP

Puesto que las CPU S7-1200 están limitadas a una sola interfaz PN, una CPU autónoma no puede adoptar la función de router. No es posible conectar una CPU autónoma a más de una subred S7 a la vez. Cuando se instalan módulos CP en la base local de la CPU, se obtiene la posibilidad de conectarla a varias subredes S7 y utilizar el routing.

En el sistema de ejemplo que se muestra más abajo, el sistema de ingeniería (ES) debe enrutar mensajes a través del PLC_2 para que el PLC_1 pueda comunicarse con el PLC_3. El ES debe descargar la tabla de routing del PLC_2 y este debe suministrar la tabla de routing del módulo CP en su base local. Cuando estas tablas de routing están en su sitio, el PLC_1 y el PLC_3 pueden comunicarse entre sí aunque no estén conectados directamente.

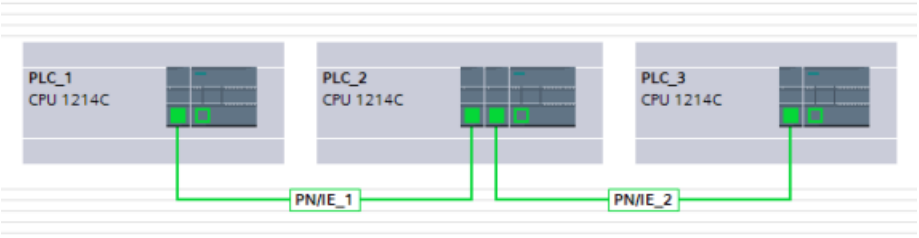


Con el fin de comprobar el routing de una subred S7 a otra, el PLC_1 debe establecer una conexión de transporte con el PLC_3 y este debe establecer una conexión con el PLC_1. De este modo se garantiza que son posibles tanto el routing de la interfaz PN/IE del PLC a un módulo CP como de un módulo CP a la interfaz PN/IE del PLC.

11.5.16.2 Routing S7 entre dos interfaces de CP

Puesto que las CPU S7-1200 soportan un máximo de tres módulos CP, estos tres módulos pueden conectarse a diferentes subredes S7. Cuando se instalan al menos dos módulos CP en la base local de la CPU y se conectan a diferentes subredes S7, es posible utilizar el routing.

En el sistema de ejemplo que se muestra más abajo, el sistema de ingeniería (ES) debe enrutar mensajes en el PLC_2, desde el módulo CP hasta el módulo CP en su base local, para que el PLC_1 pueda comunicarse con el PLC_3. El ES debe descargar la tabla de routing del PLC_2 y este debe suministrar la tabla de routing de los dos módulos CP. Cuando estas tablas de routing están en su sitio, el PLC_1 y el PLC_3 pueden comunicarse entre sí aunque no estén conectados directamente. Además, el routing tiene lugar desde el módulo CP hasta el módulo CP sin que se envíen mensajes a través de la interfaz PN/IE del PLC_2.



11.5.17 Desactivar SNMP

Simple Network Management Protocol (SNMP) es un protocolo estándar de Internet para recopilar y organizar información relacionada con dispositivos administrados en redes IP y para modificar dicha información con el fin de cambiar el comportamiento del dispositivo. Los dispositivos que soportan típicamente SNMP son, entre otros, routers, switches, servidores, estaciones de trabajo, impresoras y racks de módems.

SNMP se utiliza ampliamente en sistemas de gestión de redes para supervisar las condiciones de los dispositivos conectados en red que merecen atención administrativa. SNMP utiliza diferentes servicios y herramientas para detectar y diagnosticar la topología de red. La información sobre las propiedades de los dispositivos aptos para SNMP está incluida en los archivos MIB (Management Information Base), para los que el usuario necesita los derechos adecuados. SNMP presenta datos administrativos en forma de variables sobre los sistemas administrados que describen la configuración del sistema. Estas variables pueden ser consultadas (y a veces ajustadas) por aplicaciones administrativas.

SNMP utiliza el protocolo de transporte UDP y tiene dos componentes de red:

- Administrador SNMP: supervisa los nodos de la red
- Cliente SNMP: recopila la diferente información específica de la red en los nodos de red y la guarda de forma estructurada en la MIB (Management Information Base). Estos datos permiten realizar diagnósticos detallados de la red.

En determinadas condiciones, la aplicación puede solicitar que se desactive SNMP. Algunos ejemplos de estas condiciones son los siguientes:

- Los ajustes de seguridad de la red no permiten el uso de SNMP.
- Se utiliza una solución SNMP propia (p. ej. con instrucciones de comunicación propias).

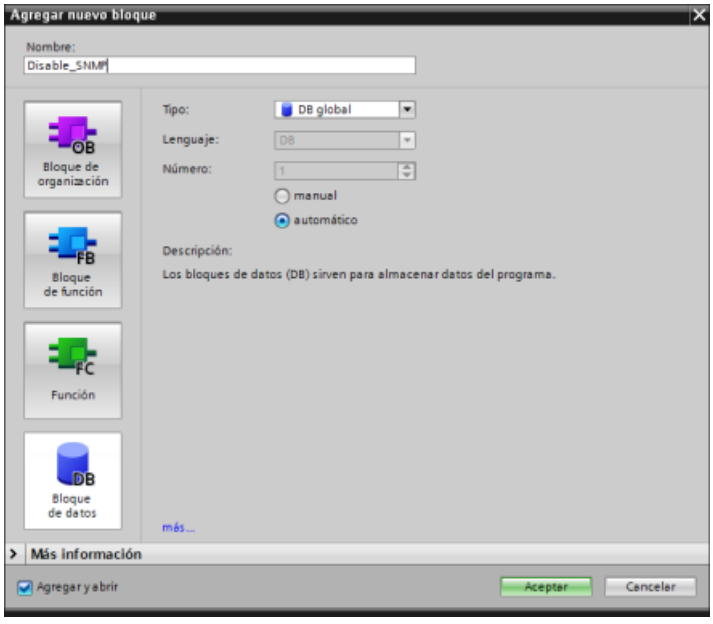
Si se desactiva SNMP para un dispositivo, algunas opciones de diagnóstico de la topología de red dejarán de estar disponibles (p. ej. el uso de la herramienta PRONETA o el servidor web de la CPU).

11.5.17.1 Desactivar SNMP

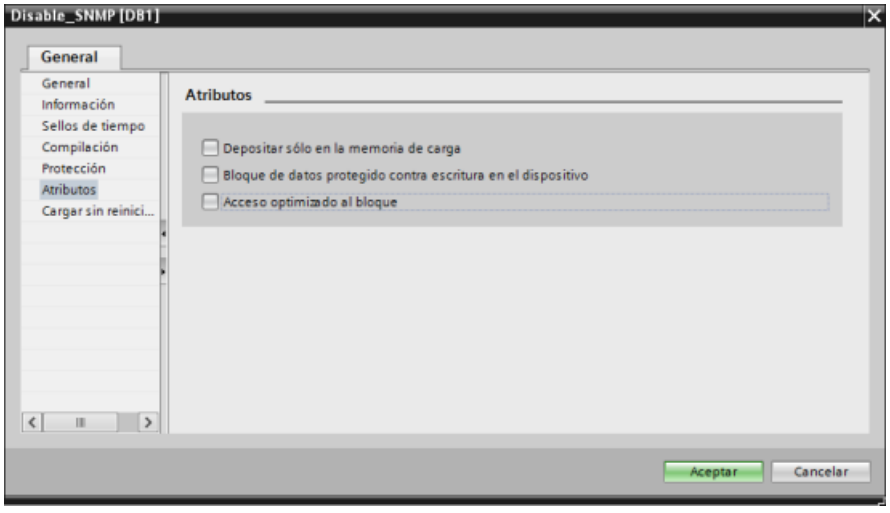
Desactivar SNMP

Siga estos pasos para desactivar SNMP en la CPU S7-1200:

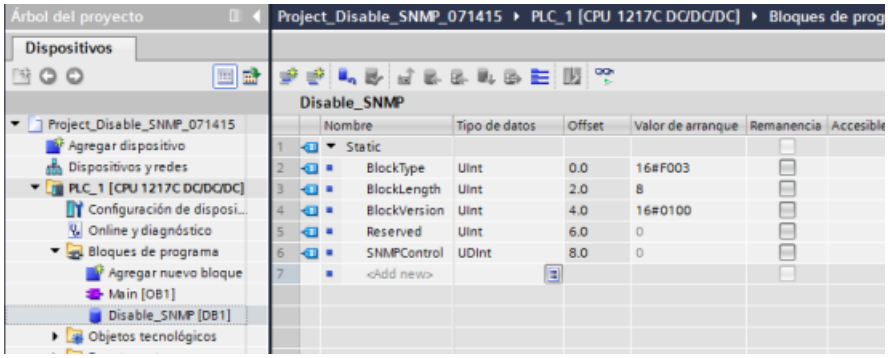
1. Cree un bloque de datos clásico (DB):



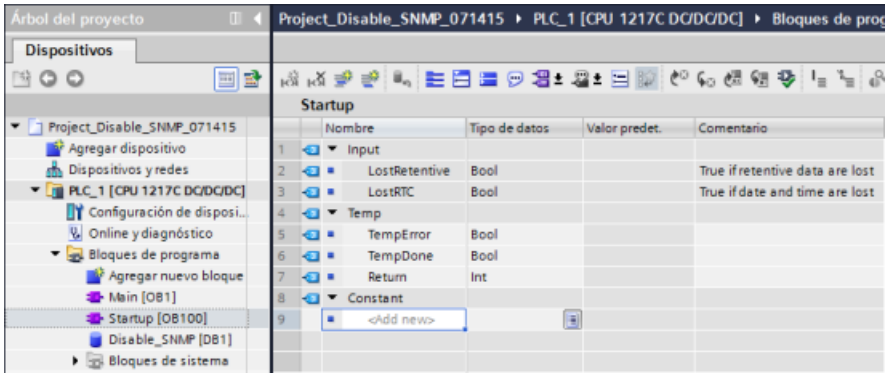
2. Seleccione las propiedades del DB que acaba de crear.
3. Seleccione la ficha "Atributos". Deseleccione la casilla de verificación "Acceso optimizado al bloque":



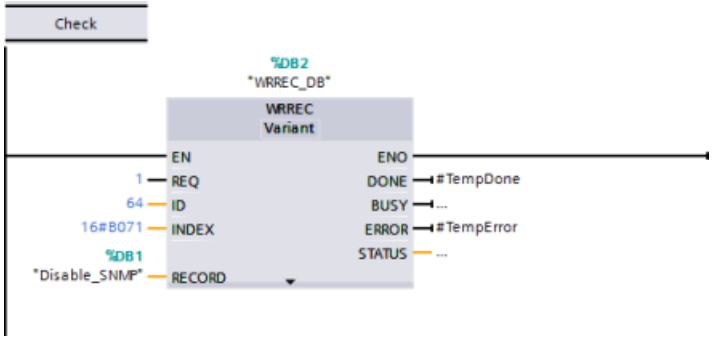
4. Haga clic en el botón "Aceptar".
   Aparece un aviso indicando que debe volver a compilarse el programa. Compile el programa ahora.
5. En la interfaz de bloque del DB clásico, cree las siguientes variables estáticas con los valores mostrados. Dichas variables se utilizarán en el programa para desactivar la implementación interna de SNMP:



6. En el OB de arranque (OB100), agregue las variables temporales tal como se muestra:



7. Desde el editor KOP, inserte una etiqueta (etiqueta de salto) (en el ejemplo que se muestra la etiqueta se denomina "Check") y una instrucción WRREC (Escribir registro) con las entradas y salidas indicadas en el segmento 1 del OB de arranque (OB100):



8. Inserte el siguiente código de bucle y verificación con la salida Etiqueta de salto (JMP). El código hace que finalice la llamada y se desactive SNMP antes de salir del OB de arranque:



11.5.18 Diagnóstico

Consulte "Bloques de organización (OB)" (Página 75) para obtener información sobre cómo usar bloques de organización (OB) para el diagnóstico con estas redes de comunicaciones.

11.5.19 Instrucciones E/S descentralizadas

Consulte "E/S descentralizadas (PROFINET, PROFIBUS o AS-i)" (Página 369) para obtener información sobre cómo utilizar las instrucciones de E/S descentralizadas con estas redes de comunicación.

11.5.20 Instrucciones de diagnóstico

Consulte "Diagnóstico (PROFINET o PROFIBUS)": "Instrucciones de diagnóstico" (Página 416) para obtener información sobre cómo utilizar estas instrucciones con estas redes de comunicación.

11.5.21 Eventos de diagnóstico de E/S descentralizadas

Consulte "Diagnóstico (PROFINET o PROFIBUS)": "Eventos de diagnóstico de E/S descentralizadas" (Página 465) para obtener información sobre cómo utilizar esta información de diagnóstico con estas redes de comunicación.

11.6 PROFIBUS

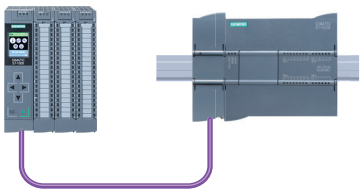
Un sistema PROFIBUS utiliza un maestro de bus para sondear dispositivos esclavos distribuidos de forma multipunto en un bus serie RS485. Un esclavo PROFIBUS es cualquier dispositivo de la periferia (convertidor E/S, válvula, motor o instrumento de medición) que procesa información y envía su salida al maestro. El esclavo conforma una estación pasiva en la red debido a que no tiene derechos de acceso al bus y solo puede acusar mensajes recibidos o bien enviar mensajes de respuesta al maestro sobre petición. Todos los esclavos PROFIBUS tienen la misma prioridad y toda la comunicación de red se inicia desde el maestro.

Un maestro PROFIBUS conforma una "estación activa" en la red. PROFIBUS DP define dos clases de maestro. Un maestro clase 1 (por lo general un controlador central programable (PLC) o un equipo dotado de un software especial) procesa la comunicación normal o intercambia datos con los esclavos que tiene asignados. Un maestro clase 2 (por lo general un dispositivo de configuración, p. ej. un portátil o una consola de programación utilizada para la puesta en marcha, mantenimiento o con fines de diagnóstico) es un dispositivo especial utilizado principalmente para poner en marcha esclavos y para fines de diagnóstico.

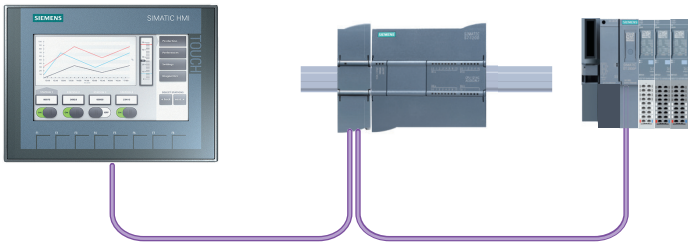
El S7-1200 se conecta a una red PROFIBUS como esclavo DP con el módulo de comunicación CM 1242-5. El módulo CM 1242-5 (esclavo DP) puede ser el interlocutor de maestros DP V0/V1. Si desea configurar el módulo en un sistema de otro fabricante, hay un archivo GSD disponible para el CM 1242-5 (esclavo DP) en el CD suministrado con el módulo y en las páginas del Siemens

Automation Customer Support (https://support.industry.siemens.com/cs/ww/es/ps/6GK7242-5DX30-0XE0) de Internet.

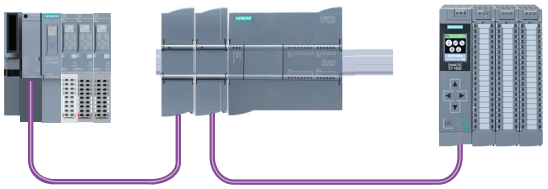
En la figura inferior el S7-1200 es un esclavo DP de un controlador S7-300:



El S7-1200 se conecta a una red PROFIBUS como maestro DP con el módulo de comunicación CM 1243-5. El módulo CM 1243-5 (maestro DP) puede ser el interlocutor de esclavos DP V0/V1. En la figura inferior, el S7-1200 es un maestro que controla un esclavo DP ET 200SP:



Si un CM 1242-5 y un CM 1243-5 están instalados conjuntamente, un S7-1200 puede actuar simultáneamente tanto de un esclavo de un sistema maestro DP de nivel superior como de maestro de un sistema esclavo DP subordinado:



A partir de V3.0 se pueden configurar tres CM PROFIBUS como máximo por estación, pudiendo haber cualquier combinación de CM maestros DP o esclavos DP. Los maestros DP en una implementación de firmware de CPU V3.0 o superior pueden controlar un máximo de 32 esclavos.

Los datos de configuración de los CMs PROFIBUS se guardan en la CPU local. Ello permite una sustitución rápida de estos módulos de comunicación en caso de necesidad.

Para utilizar PROFIBUS con S7-1200 V4.0 o CPU posteriores hay que actualizar el firmware del CM maestro PROFIBUS a V1.3 como mínimo.

Nota

Siempre debe actualizarse el firmware del CM PROFIBUS a la última versión disponible (http://support.automation.siemens.com/WW/view/es/42131407). Es posible actualizar el firmware utilizando cualquiera de los métodos siguientes:

- Las herramientas online y de diagnóstico de STEP 7 (Página 1204)
- Una SIMATIC Memory Card (Página 132)
- La página web estándar "Información del módulo" del servidor web (Página 867)
- La SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/es/view/98161300)

11.6.1 Servicios de comunicaciones de CM PROFIBUS

Los CM PROFIBUS utilizan el protocolo PROFIBUS DP-V1.

Tipos de comunicación con DP-V1

Los siguientes tipos de comunicación están disponibles con DP-V1:

- Comunicación cíclica (CM 1242-5 y CM 1243-5)
  Los dos módulos PROFIBUS admiten la comunicación cíclica para la transferencia de datos de proceso entre el esclavo DP y el maestro DP.
  El sistema operativo de la CPU procesa la comunicación cíclica. Para ello, no se necesitan bloques de software. Los datos de E/S se leen o escriben directamente de/en la imagen de proceso de la CPU.
- Comunicación acíclica (solo en CM 1243-5)
  El módulo maestro DP también admite la comunicación acíclica mediante los bloques de software:
  - La instrucción "RALRM" está disponible para el procesamiento de alarmas.
  - Las instrucciones "RDREC" y "WRREC" están disponibles para transferir datos de configuración y diagnóstico.

Funciones que no admite el CM 1243-5: SYNC/FREEZE y Get_Master_Diag

Otros servicios de comunicaciones del CM 1243-5

El módulo maestro del CM 1243-5 DP admite los siguientes servicios de comunicaciones adicionales:

- Comunicación S7
  - Servicios PUT/GET
    El maestro DP funciona como un cliente y servidor para consultas de otros PC o controladores S7 mediante PROFIBUS.
  - Comunicación PG/OP
    Las funciones de PG permiten cargar los datos de configuración y los programas de usuario de un PG y transferir los datos de diagnóstico a un PG.
    Entre los posibles interlocutores para la comunicación OP se encuentran los paneles HMI, los PC de panel SIMATIC con WinCC flexible o los sistemas SCADA que admiten la comunicación S7.

11.6.2 Referencia a los manuales de usuario de CM PROFIBUS

Más información

Puede encontrar información detallada sobre los CM PROFIBUS en los manuales de los dispositivos. Puede encontrarlos en Internet en las páginas de Customer Support de Siemens Industrial Automation con las ID de entrada siguientes:

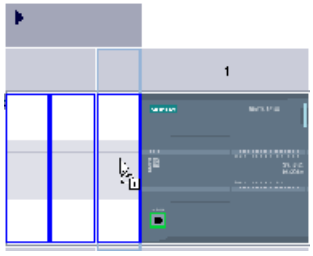
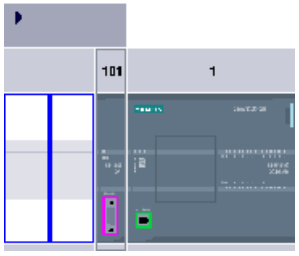
- CM 1242-5 (https://support.industry.siemens.com/cs/ww/es/ps/15667)
- CM 1243-5 (https://support.industry.siemens.com/cs/ww/es/ps/15669)

11.6.3 Configurar un maestro DP y un dispositivo esclavo

11.6.3.1 Agregar el módulo CM 1243-5 (maestro DP) y un esclavo DP

En el portal "Dispositivos y redes", utilice el catálogo de hardware para agregar módulos PROFIBUS a la CPU. Estos módulos se conectan a la izquierda de la CPU. Para insertar un módulo en la configuración de hardware, selecciónelo en el catálogo de hardware y haga doble clic en él, o bien arrástrelo hasta el slot resaltado.

Tabla 11-62 Agregar un módulo PROFIBUS CM 1243-5 (maestro DP) a la configuración de dispositivos

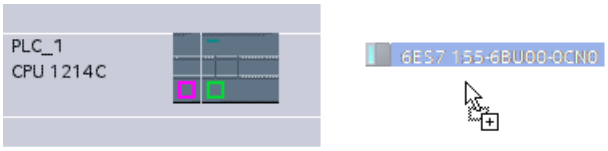
Módulo	Seleccionar el módulo	Insertar el módulo	Resultado
CM 1243-5 (maestro DP)			

Asimismo, utilice el catálogo de hardware para agregar esclavos DP. Abra por ejemplo las carpetas siguientes en el catálogo de hardware para agregar un esclavo DP ET 200SP:

- E/S descentralizadas
- ET 200SP
- Módulos de interfaz
- PROFIBUS

A continuación, seleccione "6ES7 155-6BU00-0CN0" (IM155-6 DP HF) en la lista de referencias y agregue el esclavo DP ET 200SP tal como se muestra en la figura inferior.

Tabla 11-63 Agregar un esclavo DP ET 200SP a la configuración de dispositivos

Inserte el esclavo DP	Resultado
	

11.6.3.2 Configurar las conexiones de red lógicas entre dos dispositivos PROFIBUS

Tras configurar el módulo CM 1243-5 (maestro DP) podrá proceder a configurar sus conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. Para crear la conexión PROFIBUS, seleccione la casilla (PROFIBUS) lila en el primer dispositivo. Arrastre una línea hasta la casilla PROFIBUS del segundo dispositivo. Suelte el botón del ratón para crear la conexión PROFIBUS.

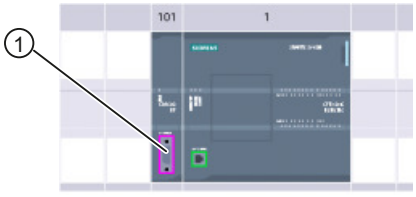
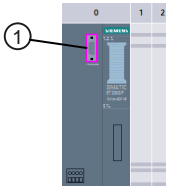
Para más información, consulte el apartado "Configuración de dispositivos: Crear una conexión de red". (Página 596)

11.6.3.3 Asignar direcciones PROFIBUS al módulo CM 1243-5 y al esclavo DP

Configurar la interfaz PROFIBUS

Tras configurar las conexiones de red lógicas entre dos dispositivos PROFIBUS, puede proceder a configurar los parámetros de las interfaces PROFIBUS. Para tal fin, haga clic en la casilla PROFIBUS lila en el módulo CM 1243-5; seguidamente, la ficha "Propiedades" de la ventana de inspección mostrará la interfaz PROFIBUS. La interfaz PROFIBUS del esclavo DP se configura del mismo modo.

Tabla 11-64 Configurar las interfaces PROFIBUS del módulo CM 1243-5 (maestro DP) y del esclavo DP ET 200SP

Módulo CM 1243-5 (maestro DP)	Esclavo DP ET 200SP
	

① Puerto PROFIBUS

Asignar la dirección PROFIBUS

En una red PROFIBUS a cada dispositivo se le asigna una dirección PROFIBUS. Esta dirección tiene un rango de 0 a 127, con las excepciones siguientes:

- Dirección 0: Reservada para la configuración de red y/o herramientas de programación asignadas al bus
- Dirección 1: Reservada por Siemens para el primer maestro
- Dirección 126: Reservada para dispositivos de fábrica que no disponen de un ajuste por interruptor y deben ser predireccionados a través de la red
- Dirección 127: Reservada para transmitir mensajes a todos los dispositivos de la red y no puede ser asignada a dispositivos operativos.

Por lo tanto, las direcciones que se pueden utilizar para dispositivos operativos PROFIBUS están comprendidas entre 2 y 125.

En la ventana de propiedades, seleccione la entrada de configuración "Dirección PROFIBUS". STEP 7 muestra el cuadro de diálogo de configuración de la dirección PROFIBUS, mediante el cual se asigna la dirección PROFIBUS del dispositivo.

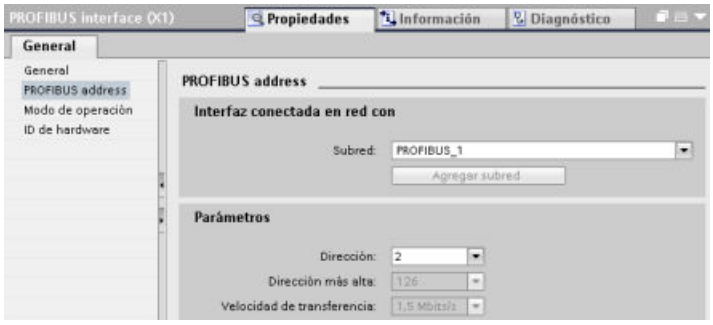


Tabla 11-65 Parámetros de la dirección PROFIBUS

Parámetro		Descripción
Subred	Nombre de la subred a la que está conectada el dispositivo. Haga clic en el botón "Agregar nueva subred" para crear una subred nueva. La opción predeterminada es "No conectado". Son posibles dos tipos de conexión: • El ajuste predeterminado "no conectado" ofrece una conexión local. • Una subred se requiere cuando la red comprende dos o más dispositivos.	
Parámetros	Dirección	Dirección PROFIBUS asignada al dispositivo
	Dirección más alta	La dirección PROFIBUS más alta está basada en las estaciones activas en PROFIBUS (por ejemplo, maestro DP). Los esclavos DP pasivos tienen independientemente direcciones PROFIBUS entre 1 y 125 incluso si la dirección PROFIBUS más alta está ajustada p. ej. en 15. La dirección PROFIBUS más alta es relevante para el envío del token (envío de los derechos de transmisión). El token solo se envía a estaciones activas. Al especificar la dirección PROFIBUS más alta se optimiza el bus.
	Velocidad de transferencia	Velocidad de transferencia de la red PROFIBUS configurada: Las velocidades de transferencia de PROFIBUS abarcan un rango de 9,6 Kbits/s a 12 Mbits/s. El ajuste de la velocidad de transferencia depende de las propiedades de los nodos PROFIBUS utilizados. La velocidad de transferencia no debe exceder la velocidad soportada por el nodo más lento. La velocidad de transferencia se ajusta normalmente para el maestro en la red PROFIBUS. Todos los esclavos DP utilizan automáticamente la misma velocidad de transferencia (auto-baud).

11.6.4 Instrucciones E/S descentralizadas

Consulte "E/S descentralizadas (PROFINET, PROFIBUS o AS-i)" (Página 369) para obtener información sobre cómo utilizar las instrucciones de E/S descentralizadas con estas redes de comunicación.

11.6.5 Instrucciones de diagnóstico

Consulte "Diagnóstico (PROFINET o PROFIBUS)": "Instrucciones de diagnóstico" (Página 416) para obtener información sobre cómo utilizar estas instrucciones con estas redes de comunicación.

11.6.6 Eventos de diagnóstico de E/S descentralizadas

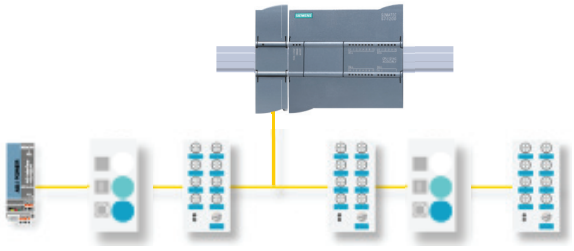
Consulte "Diagnóstico (PROFINET o PROFIBUS)": "Eventos de diagnóstico de E/S descentralizadas" (Página 465) para obtener información sobre cómo utilizar esta información de diagnóstico con estas redes de comunicación.

11.7 AS-i

El maestro AS-i CM 1243-2 del S7-1200 permite conectar una red AS-i a una CPU S7-1200.

La interfaz actuador/sensor (o bien AS-i) es un sistema de conexión de red con maestro individual que se utiliza en el nivel más bajo de sistemas de automatización. El CM 1243-2 asume la función de maestro AS-i en la red. Mediante un único cable AS-i, los sensores y actuadores (dispositivos esclavos AS-i) se pueden conectar a la CPU a través del CM 1243-2. El CM 1243-2 gestiona toda la coordinación de la red AS-i y transmite datos e información de estado de los actuadores y sensores a la CPU a través de las direcciones de E/S asignadas al CM 1243-2. Se puede acceder a valores binarios o analógicos dependiendo del tipo de esclavo. Los esclavos AS-i son los canales de entrada y salida del sistema AS-i y solo están activos cuando el CM 1243-2 los llama.

En la siguiente figura, el S7-1200 es un maestro AS-i que controla los dispositivos esclavos digitales/analógicos del módulo E/S AS-i.



Para utilizar AS-i con CPUs S7-1200 V4.0 hay que actualizar el firmware del CM del maestro AS-i a V1.1.

Esta actualización se realiza empleando el servidor web o una SIMATIC Memory Card.

Nota

Con las CPUs S7-1200 V4.0, si se utiliza el servidor web o una SIMATIC Memory Card para actualizar el firmware del AS-i de V1.0 a V1.1, es preciso actualizar el firmware del AS-i en el maestro AS-i CM 1243-2 conforme al siguiente procedimiento:

1. Descargue la actualización de firmware al maestro AS-i CM 1243-2.
2. Cuando la descarga se haya completado, desconecte y vuelva a conectar la CPU S7-1200 para completar el proceso de actualización de firmware en el maestro AS-i CM 1243-2.
3. Repita los pasos 1 y 2 para cada maestro AS-i CM 1243-2. El PLC S7-1200 permite un máximo de tres maestros AS-i CM 1243-2.

Nota

Se recomienda siempre actualizar el firmware del CM AS-i a la última versión disponible (http://support.automation.siemens.com/WW/view/es/43416171) en la página de atención al cliente de Siemens.

11.7.1 Configuración de un maestro AS-i y un dispositivo esclavo

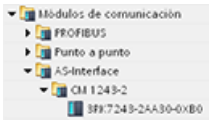
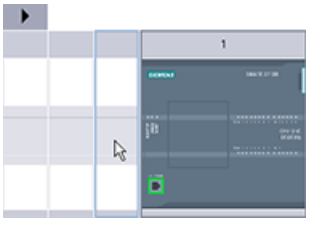
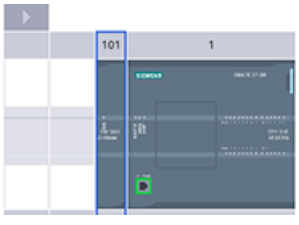
El maestro AS-i CM 1243-2 está integrado en el sistema de automatización S7-1200 como módulo de comunicación.

Encontrará información detallada sobre el maestro AS-i CM 1243-2 en el manual "Maestro AS-i CM 1243-2 y módulo de desacoplamiento de datos AS-i DCM 1271 para SIMATIC S7-1200" (https://support.industry.siemens.com/cs/ww/es/ps/15750/man).

11.7.1.1 Agregar el módulo maestro AS-i CM 1243-2 y esclavo AS-i

Utilice el catálogo de hardware para agregar módulos maestros AS-i CM1243-2 a la CPU. Estos módulos se conectan por el lado izquierdo de la CPU, siendo posible utilizar un máximo de tres módulos maestros AS-i CM1243-2. Para insertar un módulo en la configuración de hardware, selecciónelo en el catálogo de hardware y haga doble clic en él, o bien arrástrelo hasta el slot resaltado.

Tabla 11-66 Agregar un módulo maestro AS-i CM1243-2 a la configuración de dispositivos

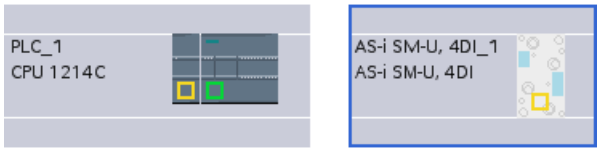
Módulo	Seleccionar el módulo	Insertar el módulo	Resultado
CM 1243-2 maestro AS-i			

Asimismo, utilice el catálogo de hardware para agregar esclavos AS-i. Por ejemplo, para agregar un esclavo "módulo E/S, compacto, digital, de entrada" en el catálogo de hardware, expanda las siguientes carpetas:

- Dispositivos de campo
- Esclavos ASInterface

A continuación, seleccione "3RG9 001-0AA00" (AS-i SM-U, 4DI) en la lista de referencias y agregue el esclavo "módulo E/S, compacto, digital, de entrada" como se muestra en la figura de abajo.

Tabla 11-67 Agregar un esclavo AS-i a la configuración de dispositivos

Insertar el esclavo AS-i	Resultado
	

11.7.1.2 Configuración de las conexiones de red lógicas entre dos dispositivos AS-i

Después de configurar el maestro AS-i CM1243-2, ya puede configurar las conexiones de red.

En el portal "Dispositivos y redes", utilice la "Vista de red" para crear las conexiones de red entre los dispositivos del proyecto. Para crear la conexión AS-i, seleccione la casilla (AS-i) amarilla en el primer dispositivo. Arrastre una línea hasta la casilla AS-i del segundo dispositivo. Suelte el botón del ratón para crear la conexión AS-i.

Para más información, consulte el apartado "Configuración de dispositivos: Crear una conexión de red". (Página 596)

11.7.1.3 Configurar las propiedades del maestro AS-i CM1243-2

Para configurar los parámetros para la interfaz AS-i, haga clic en la casilla AS-i amarilla del módulo maestro AS-i CM1243-2; en la ficha "Propiedades" de la ventana de inspección se mostrará la interfaz AS-i.

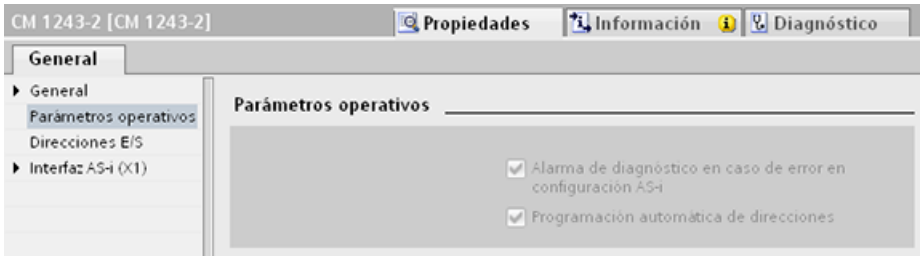
En la ventana de inspección de STEP 7 puede ver, configurar y cambiar la información general, las direcciones y los parámetros operativos:

Tabla 11-68 Propiedades del módulo maestro AS-i CM1243-2

Propiedad	Descripción
General	Nombre del maestro AS-i CM1243-2
Parámetros operativos	Parámetros para la respuesta del maestro AS-i
Direcciones de E/S	Área para las direcciones de E/S de esclavos
Interfaz AS-i (X1)	Red AS-i asignada

Nota

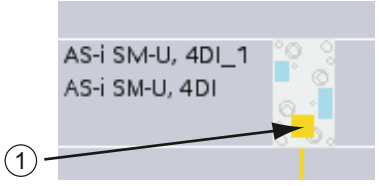
Las opciones "Alarma de diagnóstico en caso de error en la configuración AS-i" y "Programación automática de direcciones" siempre están activas y, por consiguiente, aparecen en gris.



11.7.1.4 Asignar una dirección AS-i a un esclavo AS-i

Configurar la interfaz AS-i del esclavo

Para configurar los parámetros para la interfaz AS-i, haga clic en la casilla amarilla AS-i del esclavo AS-i; en la ficha "Propiedades" de la ventana de inspección se mostrará la interfaz AS-i.



① Puerto AS-i

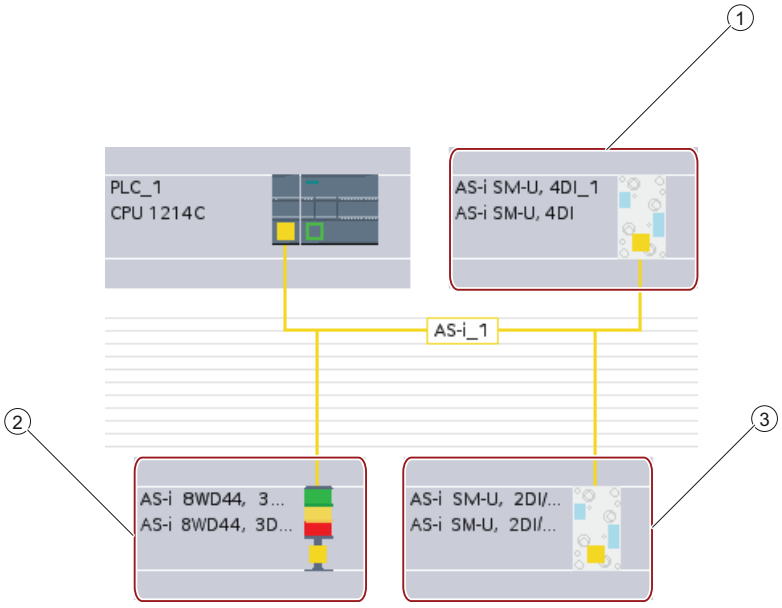
Asignar la dirección de esclavo AS-i

En una red AS-i, a cada dispositivo se le asigna una dirección de esclavo AS-i. Dicha dirección puede estar comprendida entre 0 y 31; no obstante, el 0 está reservado exclusivamente para nuevos dispositivos esclavos. Las direcciones de esclavo van de 1(A o B) a 31(A o B) para un total de 62 dispositivos esclavos.

Los dispositivos AS-i "estándar" utilizan la dirección completa, que es numérica y no tiene la designación A o B. Los dispositivos AS-i con "nodo A/B" utilizan la parte A o B de cada dirección, lo que permite utilizar dos veces cada una de las 31 direcciones. El rango de direcciones va de 1A a 31A y de 1B a 31B.

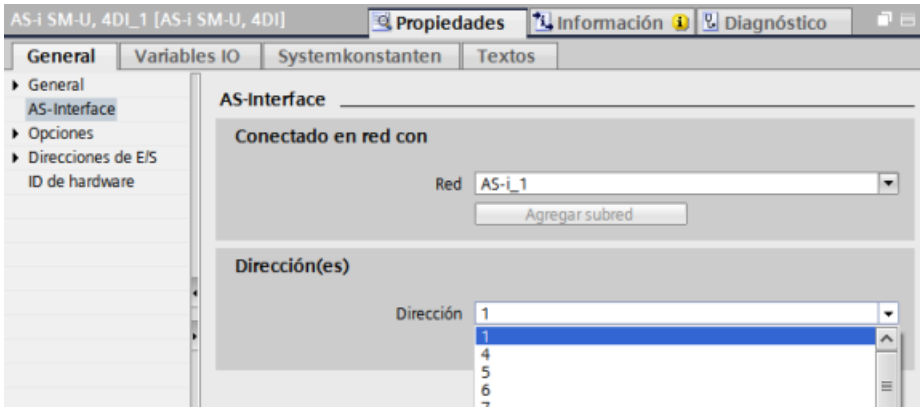
Cualquier dirección que se encuentre en el rango de 1 - 31 puede ser asignada a un dispositivo esclavo, es decir, no es relevante si los esclavos comienzan por la dirección 21 o si el primer esclavo tiene asignada realmente la dirección 1.

En el ejemplo inferior, tres dispositivos AS-i se han direccionado como "1" (un dispositivo estándar), "2A" (un dispositivo con nodo A/B) y "3" (un dispositivo estándar):



① Dirección de esclavo AS-i 1; dispositivo: AS-i SM-U, 4DI; referencia: 3RG9 001-0AA00
② Dirección de esclavo AS-i 2A; dispositivo: AS-i 8WD44, 3DO, A/B; referencia: 8WD4 428-0BD
③ Dirección de esclavo AS-i 3; dispositivo: AS-i SM-U, 2DI/2DO; referencia: 3RG9 001-0AC00

Asigne aquí la dirección de esclavo AS-i:



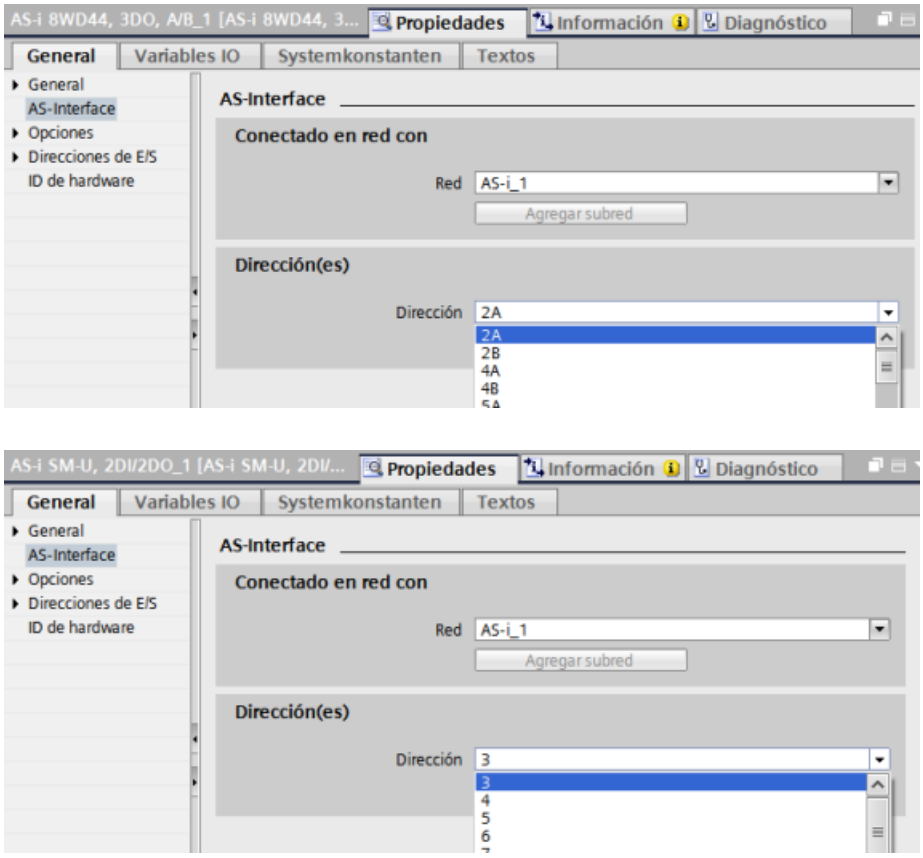


Tabla 11-69 Parámetros de la interfaz AS-i

Parámetro	Descripción
Red	Nombre de la red a la que está conectado el dispositivo
Direcciones	Dirección AS-i asignada para el dispositivo esclavo dentro del rango de 1(A o B) a 31(A o B) para un total de 62 dispositivos esclavos

11.7.2 Intercambio de datos entre el programa de usuario y los esclavos AS-i

11.7.2.1 Configuración básica de STEP 7

El maestro AS-i reserva un área de datos de 62 bytes en el área E/S de la CPU. Se accede a los datos digitales en bytes; para cada esclavo, hay un byte de datos de entrada y un byte de datos de salida.

La asignación de las conexiones AS-i de los esclavos digitales AS-i a los bits de datos del byte asignado se indica en la ventana de inspección del maestro AS-i CM 1243-2.



Se accede a los datos de los esclavos AS-i en el programa de usuario utilizando las direcciones E/S visualizadas con las operaciones de lógica binaria adecuadas (p. ej. "Y") o bien asignaciones de bits.

Nota

El parámetro "Asignación de sistema" se activa automáticamente si no se configuran los esclavos AS-i con STEP 7.

Si no se configuran esclavos, hay que informar al maestro AS-i CM 1243-2 acerca de la configuración de bus real utilizando la función online "ES > DEBE".

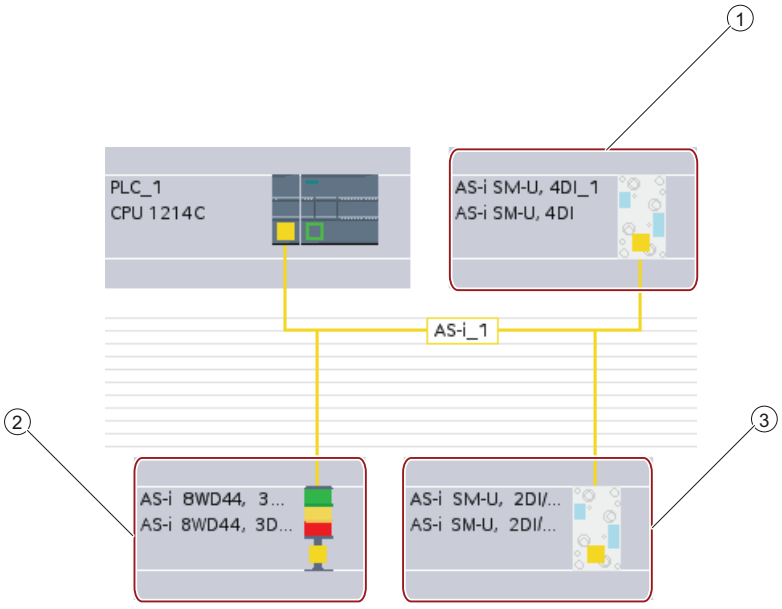
Más información

Encontrará información detallada sobre el maestro AS-i CM 1243-2 en el manual "Maestro AS-i CM 1243-2 y módulo de desacoplamiento de datos AS-i DCM 1271 para SIMATIC S7-1200" (https://support.industry.siemens.com/cs/ww/es/ps/15750/man).

11.7.2.2 Configurar esclavos con STEP 7

Transferir valores digitales AS-i

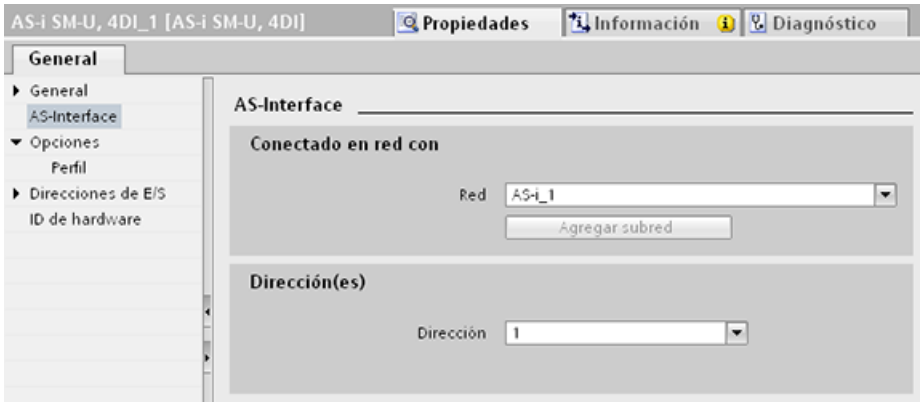
La CPU accede a las entradas y salidas digitales de los esclavos AS-i a través del maestro AS-i CM 1243-2 durante el funcionamiento cíclico. Accede a los datos utilizando las direcciones E/S o bien mediante una transferencia de registros.



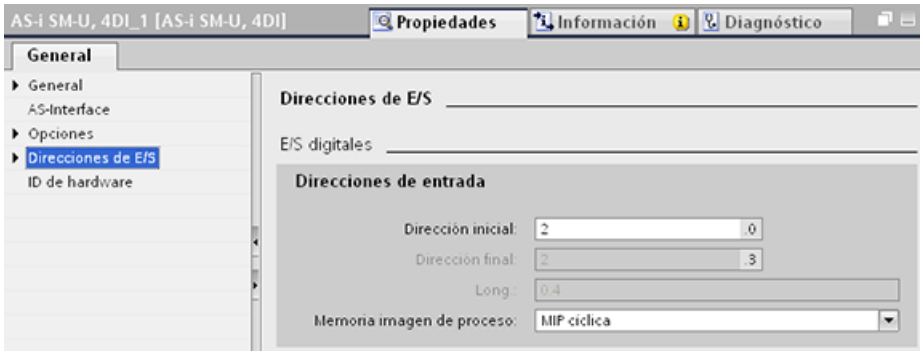
① Dirección esclavo AS-i 1
② Dirección esclavo AS-i 2A
③ Dirección esclavo AS-i 3

En este caso, se accede a los datos digitales en bytes (es decir, cada esclavo digital AS-i tiene asignado un byte). Al configurar los esclavos AS-i en STEP 7, la dirección E/S para acceder a los datos desde el programa de usuario se muestra en la ventana de inspección del esclavo AS-i correspondiente.

El módulo de entradas digitales (AS-i SM-U, 4DI) en la red AS-i superior ha recibido la dirección de esclavo 1. Al hacer clic en el módulo de entradas digitales, la ficha "AS-Interface" de "Propiedades" muestra la dirección del esclavo, como se muestra a continuación:

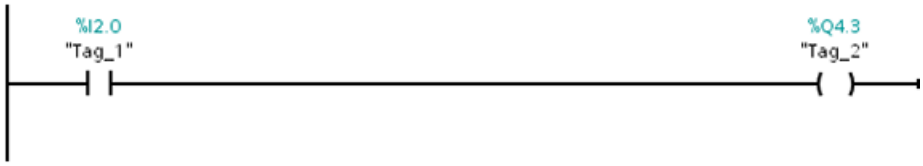


El módulo de entradas digitales (AS-i SM-U, 4DI) en la red AS-i anterior ha recibido la dirección de E/S 2. Al hacer clic en el módulo de entradas digitales, la ficha "Direcciones de E/S" de "Propiedades" muestra la dirección de E/S, como se muestra a continuación:



Para acceder a los datos de los esclavos AS-i en el programa de usuario, se utilizan las direcciones E/S con las operaciones de lógica binaria adecuadas (p. ej. "Y") o bien asignaciones de bits. El siguiente programa sencillo muestra cómo funciona la asignación:

La entrada 2.0 se interroga en este programa. En el sistema AS-i, dicha entrada pertenece al esclavo1 (byte de entrada 2, bit 0). La salida 4.3, que se activa entonces, corresponde al esclavo AS-i 3 (byte de salida 4, bit 3)



Transferir valores analógicos AS-i

Se puede acceder a los datos analógicos de un esclavo AS-i desde la memoria imagen de proceso de la CPU siempre que se haya configurado dicho esclavo AS-i en STEP 7 como esclavo analógico.

Si no se ha configurado el esclavo analógico en STEP 7, solo será posible acceder a los datos del esclavo AS-i utilizando funciones acíclicas (interfaz de registro). En el programa de usuario de la

CPU, las llamadas AS-i se leen y escriben utilizando las instrucciones E/S descentralizadas RDREC (leer registro) y WRREC (escribir registro).

Nota

Si se ha especificado una configuración de los esclavos AS-i con STEP 7 y se ha descargado en la estación S7, esta es transferida por la CPU al maestro AS-i CM 1243-2 durante el arranque de la estación S7. Cualquier configuración existente que se haya determinado mediante la función online "Asignación de sistema" (Página 789) ("ES -> DEBE") se sobrescribirá.

Más información

Encontrará información detallada sobre el maestro AS-i CM 1243-2 en el manual "Maestro AS-i CM 1243-2 y módulo de desacoplamiento de datos AS-i DCM 1271 para SIMATIC S7-1200" (http://support.automation.siemens.com/WW/view/en/50414115/133300).

11.7.3 Instrucciones E/S descentralizadas

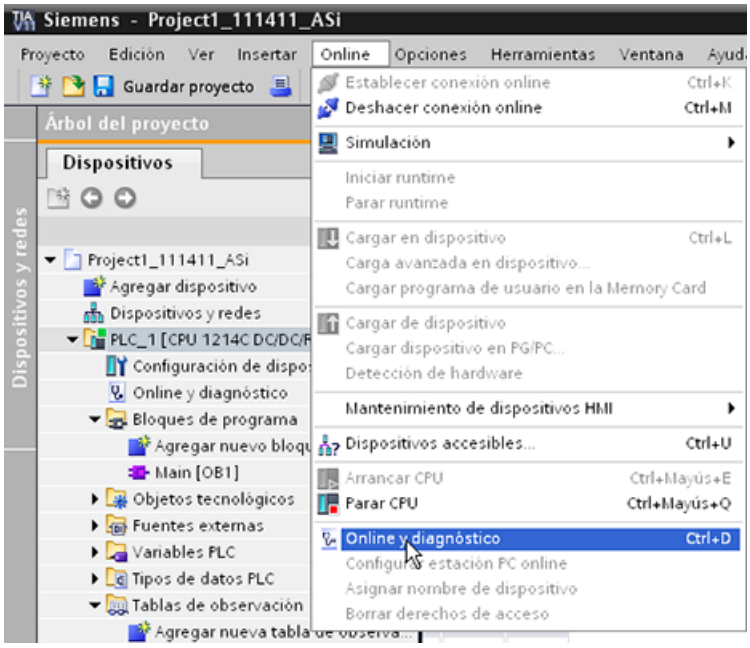
Consulte "E/S descentralizadas (PROFINET, PROFIBUS o AS-i)" (Página 369) para obtener información sobre cómo utilizar las instrucciones de E/S descentralizadas con estas redes de comunicación.

11.7.4 Trabajar con herramientas online AS-i

Cambiar los modos de operación AS-i online

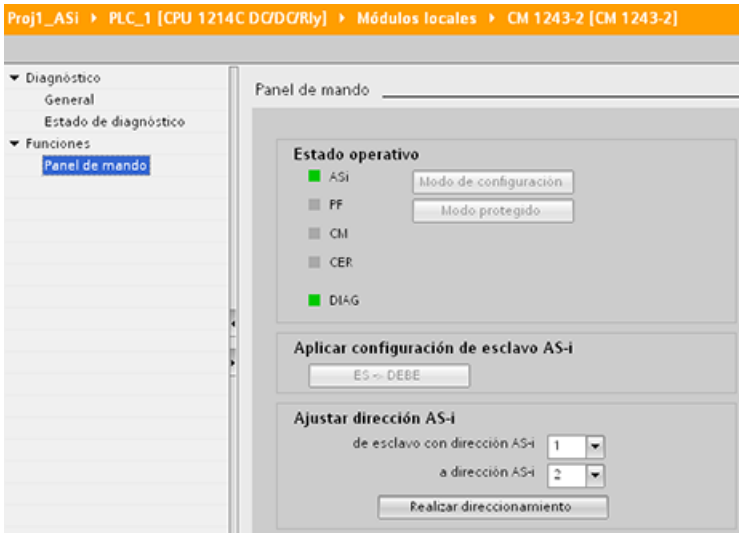
Hay que pasar a online para ver y modificar los modos de operación AS-i.

Para cambiar al modo online, vaya a "Configuración de dispositivos" con el módulo maestro AS-i CM 1243-2 seleccionado y, seguidamente, haga clic en el botón "Establecer conexión online" de la barra de herramientas. A continuación, seleccione el comando "Online y diagnóstico" del menú "Online".



Existen dos modos de operación AS-i:

- Modo de protección:
  - No es posible modificar el dispositivo esclavo AS-i ni las direcciones E/S de la CPU.
  - El LED verde "CM" está apagado.
- Modo de configuración:
  - Es posible realizar los cambios necesarios en el dispositivo esclavo AS-i y en las direcciones E/S de la CPU.
  - El LED verde "CM" está encendido.



En el campo "Ajustar dirección AS-i", puede cambiar la dirección del esclavo AS-i. A un esclavo nuevo que aún no tenga dirección se le asigna siempre la dirección 0. El maestro lo reconoce

como esclavo nuevo aunque no tenga asignada una dirección y no lo incluye en la comunicación normal hasta que no le sea asignada una dirección.

Error de configuración

Si el LED amarillo "CER" está encendido, existe un error en la configuración del dispositivo esclavo AS-i. Seleccione el botón "ES > DEBE" para sobrescribir la configuración del dispositivo esclavo del módulo maestro AS-i CM 1243-2 con la configuración del dispositivo esclavo de la red de campo AS-i.

11.8 Comunicación S7

11.8.1 GET y PUT (leer y escribir datos de una CPU remota)

Las instrucciones GET y PUT se pueden utilizar para comunicarse con CPU S7 a través de conexiones PROFINET y PROFIBUS. Esto solo es posible si la función "Permitir acceso vía comunicación PUT/GET" está activada para la CPU del interlocutor en la propiedad "Protección" de las propiedades de la CPU local:

- Acceso a datos en una CPU remota: una CPU S7-1200 solo puede utilizar direcciones absolutas en el campo de entrada ADDR_x para direccionar variables de CPU remotas (S7-200/300/400/1200).
- Acceder a los datos en un DB estándar: una CPU S7-1200 solo puede utilizar direccionar absolutas en el campo de entrada ADDR_x para direccionar variables de DB en un DB estándar de una CPU S7 remota.
- Acceder a los datos en un DB optimizado: una CPU S7-1200 no puede acceder a variables de DB en un DB optimizado de una CPU S7-1200 remota.
- Acceso a datos en una CPU local: una CPU S7-1200 puede usar direcciones tanto absolutas como simbólicas como entradas de los campos de entrada RD_x o SD_x de las instrucciones GET o PUT.

Nota

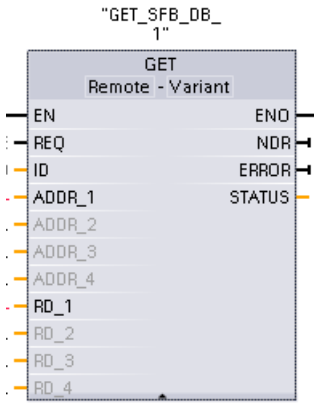
La operación GET/PUT del programa de la CPU V4.0 no está activada automáticamente

Una operación GET/PUT del programa de una CPU V3.0 está activada automáticamente en una CPU V4.0.

Sin embargo, una operación GET/PUT del programa de una CPU V4.0 no está activada automáticamente en una CPU V4.0. Hay que ir a la "Configuración de dispositivos" de la CPU, ficha "Propiedades" de la ventana de inspección, propiedad "Protección" para activar Acceso GET/PUT (Página 163).

Tabla 11-70 Instrucciones GET y PUT

KOP / FUP	SCL	Descripción

	`"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_]);`
		Utilice la instrucción GET para leer datos desde una CPU S7 remota. La CPU remota puede estar tanto en modo RUN como STOP.
		STEP 7 crea automáticamente el DB al introducir la instrucción.

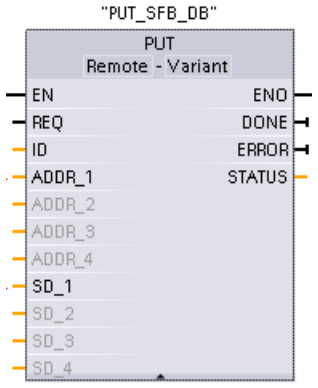
	`"PUT_DB"(`
	`req:=_bool_in_,`
	`ID:=_word_in_,`
	`done=>_bool_out_,`
	`error=>_bool_out_,`
	`status=>_word_out_,`
	`addr_1:=_remote_inout_,`
	`[...addr_4:=_remote_inout_,]`
	`sd_1:=_variant_inout_,`
	`[....sd_4:=_variant_inout_]);`
		Utilice la instrucción PUT para escribir datos en una CPU S7 remota. La CPU remota puede estar tanto en modo RUN como STOP.
		STEP 7 crea automáticamente el DB al introducir la instrucción.

Tabla 11-71 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	Input	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación.
ID	Input	CONN_PRG (Word)	ID de conexión S7 (Hex)
NDR (GET)	Output	Bool	Nuevos datos listos: • 0: petición no iniciada aún o en proceso • 1: tarea ejecutada correctamente
DONE (PUT)	Output	Bool	DONE: • 0: petición no iniciada aún o en proceso • 1: tarea ejecutada correctamente

Parámetro y tipo		Tipo de datos	Descripción
ERROR STATUS	Output Output	Bool Word	• ERROR=0 Valor de STATUS: – 0000H: ni advertencia ni error – <> 0000H: Advertencia, STATUS suministra información detallada • ERROR=1 Existe un error. STATUS suministra información detallada sobre la naturaleza del error.
ADDR_1	InOut	Remoto	Puntero a las áreas de memoria de la CPU remota que almacena los datos que deben leerse (GET) o que se envían (PUT).
ADDR_2	InOut	Remoto	
ADDR_3	InOut	Remoto	
ADDR_4	InOut	Remoto	
RD_1 (GET) SD_1 (PUT)	InOut	Variante	Puntero a las áreas de memoria de la CPU local que almacena los datos que deben leerse (GET) o enviarse (PUT). Tipos de datos permitidos: Bool (solo se permite un único bit), Byte, Char, Word, Int, DWord, DInt o Real. Nota: Si el puntero accede a un DB, se debe especificar la dirección absoluta, p. ej.: P# DB10.DBX5.0 Byte 10 En este caso, 10 representa el número de bytes que deben leerse (GET) o transmitirse (PUT).
RD_2 (GET) SD_2 (PUT)	InOut	Variante	
RD_3 (GET) SD_3 (PUT)	InOut	Variante	
RD_4 (GET) SD_4 (PUT)	InOut	Variante	

Asegúrese de que la longitud (número de bytes) y los tipos de datos para los parámetros ADDR_x (CPU remota) y RD_x o SD_x (CPU local) concuerden. El número tras el identificador "Byte" es el número de bytes referenciado por el parámetro ADDR_x, RD_x o SD_x .

Nota

El número total de bytes recibidos en una instrucción GET o el número total de bytes enviados en una instrucción PUT es limitado. Las limitaciones se basan en el número de las cuatro posibles direcciones y áreas de memoria que se utilizan:

- Si solo se utilizan ADDR_1 y RD_1/SD_1, una instrucción GET puede recibir 222 bytes y una instrucción PUT puede enviar 212 bytes.
- Si se utilizan ADDR_1, RD_1/SD_1, ADDR_2 y RD_2/SD_2, una instrucción GET puede recibir un total de 218 bytes y una instrucción PUT puede enviar un total de 196 bytes.
- Si se utilizan ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3 y RD_3/SD_3 , una instrucción GET puede recibir un total de 214 bytes y una instrucción PUT puede enviar un total de 180 bytes.
- Si se utilizan ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3, RD_3/SD_3, ADDR_4 y RD_4/SD_4, una instrucción GET puede recibir un total de 210 bytes y una instrucción PUT puede enviar un total de 164 bytes.

La suma del número de bytes de cada uno de los parámetros de dirección y área de memoria debe ser menor o igual que los límites definidos. Si se superan esos límites, las instrucciones GET o PUT devuelven un error.

Con un flanco ascendente del parámetro REQ, la operación de lectura (GET) o de escritura (PUT) carga los parámetros ID, ADDR_1, y RD_1 (GET) o SD_1 (PUT).

- Para GET: La CPU remota devuelve los datos solicitados a las áreas de recepción (RD_x), comenzando por el siguiente ciclo. Una vez la operación de lectura ha finalizado sin errores, el parámetro NDR se pone a 1. Solo es posible iniciar una operación nueva tras finalizar la operación anterior.
- Para PUT: La CPU local empieza a enviar los datos (SD_x) a la posición de memoria (ADDR_x) en la CPU remota. Una vez la operación de escritura ha finalizado sin errores, la CPU remota devuelve un acuse de ejecución. El parámetro DONE de la instrucción PUT se pone a 1. Solo es posible iniciar una operación de escritura nueva tras finalizar la operación anterior.

Nota

Para garantizar la coherencia de los datos, antes de acceder a los datos o inicializar otra operación de lectura o escritura, compruebe siempre que la operación haya finalizado (NDR = 1 para GET o DONE = 1 para PUT).

Los parámetros ERROR y STATUS suministran información sobre el estado de la operación de lectura (GET) o escritura (PUT).

Tabla 11-72 Información de error

ERROR	STATUS (decimal)	Descripción
0	11	• La nueva tarea no se puede ejecutar debido a que aún no ha finalizado una tarea anterior. • La tarea se está procesando ahora en una clase de prioridad más baja.
0	25	Se ha iniciado la comunicación. La tarea se está procesando.
1	1	Problemas de comunicación, p. ej.: • No se ha cargado la descripción de la conexión (local o remota) • Conexión interrumpida (p. ej.: cable, CPU desactivada o CM/CB/CP en modo STOP) • Conexión con el interlocutor aún no establecida
1	2	Acuse negativo por parte del dispositivo interlocutor. No se puede ejecutar la tarea.
1	4	Errores en los punteros del área de emisión (RD_x para GET o SD_x para PUT) relacionados con la longitud de los datos o el tipo de datos.
1	8	Error de acceso a la CPU interlocutora
1	10	No es posible acceder a la memoria de usuario local (p. ej. intentar acceder a un DB borrado)
1	12	Al llamar el SFB: • Se especificó un DB de instancia que no pertenece a GET o PUT • No se especificó un DB de instancia sino un DB compartido • No se ha encontrado ningún DB de instancia (al cargar un nuevo DB de instancia)
1	20	• Se ha excedido el número máximo de tareas/instancias simultáneas • Las instancias se sobrecargaron al arrancar la CPU Este estado puede presentarse al ejecutar por primera vez las instrucciones GET o PUT
1	27	No hay una instrucción GET o PUT que concuerde con la CPU.

11.8.2 Crear una conexión S7

Mecanismos de conexión

Para acceder a interlocutores remotos con las instrucciones PUT/GET, el usuario también debe tener permiso.

De forma predeterminada, la opción "Permitir acceso vía comunicación PUT/GET" está desactivada. En ese caso, el acceso de lectura y escritura a los datos de la CPU solo es posible para conexiones que necesiten configuración o programación tanto para la CPU local como para el interlocutor. El acceso mediante instrucciones BSEND/BRCV es posible, por ejemplo.

En consecuencia, no se pueden realizar conexiones para las cuales la CPU local es solo un servidor cuando se usa la CPU, lo que significa que no hay configuración/programación de la comunicación con el interlocutor en la CPU local, por ejemplo:

- Acceso PUT/GET, FETCH/WRITE o FTP mediante módulos de comunicaciones
- Acceso con PUT/GET desde otras CPU S7
- Acceso a HMI mediante la comunicación PUT/GET

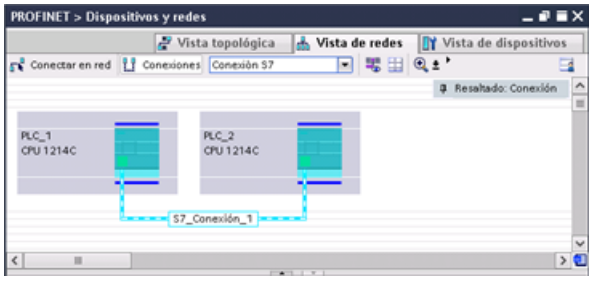
Si desea permitir el acceso a los datos de la CPU desde el lado del cliente, es decir, si no desea limitar los servicios de comunicación de la CPU, puede configurar la protección de acceso para la CPU S7-1200 (Página 163) para ese nivel de seguridad.

Tipos de conexión

El tipo de conexión seleccionado crea una conexión con un interlocutor. La conexión se configura, establece y vigila automáticamente.

En el portal "Dispositivos y redes", utilice la "Vista de redes" para crear las conexiones de red entre los dispositivos del proyecto. En primer lugar, haga clic en la ficha "Conexiones" y luego seleccione el tipo de conexión en la lista desplegable ubicada justo a la derecha (p. ej. una conexión S7). Haga clic en el cuadro verde (PROFINET) del primer dispositivo y trace una línea hasta el cuadro PROFINET del segundo dispositivo. Suelte el botón del ratón para crear la conexión PROFINET.

Encontrará más información en "Crear una conexión de red" (Página 596).



Haga clic en el botón "Resaltado: conexión" para acceder al cuadro de diálogo de configuración "Propiedades" de la instrucción de comunicación.

11.8.3 Configurar la vía de conexión local/interlocutor entre dos dispositivos

Configurar los parámetros generales

Los parámetros de comunicación se especifican en el cuadro de diálogo de configuración "Propiedades" de la instrucción de comunicación. Este diálogo aparece en el lado inferior de la página cuando se ha seleccionado alguna parte de la instrucción.

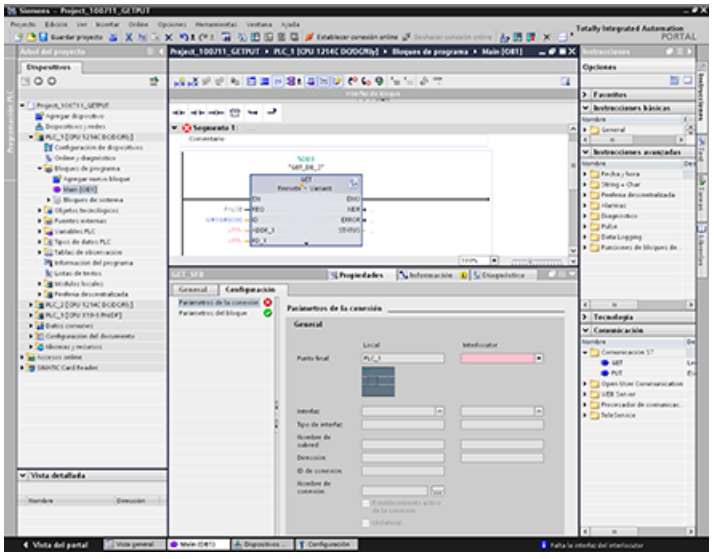
Encontrará más información en "Configuración de dispositivos: Configurar la vía de conexión local/interlocutor (Página 596)"

En el área "Detalles de dirección" del diálogo "Parámetros de la conexión" se definen los TSAPs o puertos que deben utilizarse. El TSAP o puerto de una conexión en la CPU se introduce en el campo "TSAP local". El TSAP o puerto asignado a la conexión en la CPU interlocutora se introduce en el campo "TSAP del interlocutor".

11.8.4 Asignación de parámetros de conexión GET/PUT

La asignación de parámetros de conexión mediante las instrucciones GET/PUT es una ayuda al usuario para configurar conexiones de comunicación S7 de CPU a CPU.

Tras insertar un bloque GET o PUT, STEP 7 muestra el cuadro de diálogo para la asignación de parámetros de conexión de las instrucciones GET/PUT:



La ventana de inspección muestra las propiedades de la conexión cuando se selecciona cualquier parte de la instrucción. Los parámetros de comunicación se configuran en la ficha "Configuración" de "Propiedades" de la instrucción de comunicación.

Nota

La operación GET/PUT del programa de la CPU V4.1 y superior no está activada automáticamente

Una operación GET/PUT del programa de una CPU V3.0 está activada automáticamente en una CPU V4.1 y superior.

Sin embargo, una operación GET/PUT del programa de una CPU V4.1 y superior no está activada automáticamente en una CPU V4.1 y superior. Hay que ir a la "Configuración de dispositivos" de la CPU, ficha "Propiedades" de la ventana de inspección, propiedad "Protección" para activar Acceso GET/PUT (Página 163).

11.8.4.1 Parámetros de la conexión

La página "Parámetros de conexión" le permite configurar la conexión S7 necesaria y configurar el parámetro "ID de conexión" referenciado por la ID del parámetro de bloque GET/PUT. El contenido de la página ofrece información sobre el punto final local y le permite que defina la interfaz local. También puede definir el punto final del interlocutor.

La página "Parámetros del bloque" le permite configurar los parámetros de bloque adicionales.

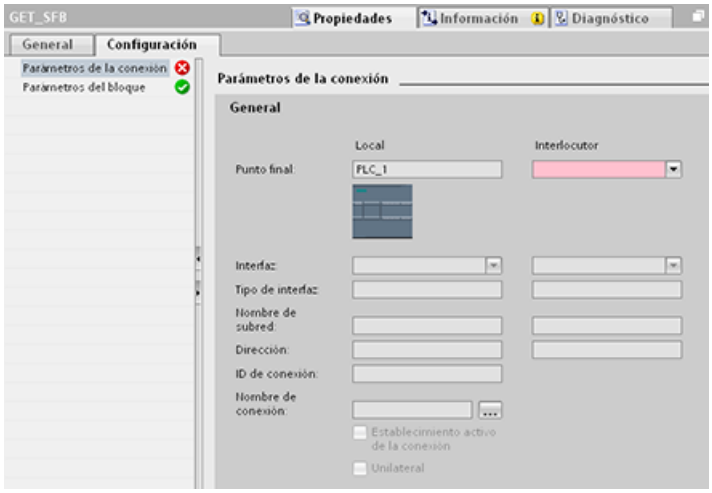


Tabla 11-73 Parámetro de conexión: definiciones generales

Parámetro		Definición
Parámetro de conexión: General	Punto final	"Punto final local": nombre asignado a la CPU local "Punto final del interlocutor": nombre asignado a la CPU interlocutora (remota) Nota: en la lista desplegable "Punto final del interlocutor", el sistema ofrece todos los interlocutores S7 potenciales del proyecto actual además de la opción "no especificado". Un interlocutor no especificado representa a un interlocutor que no está actualmente en el proyecto STEP 7 (por ejemplo, un dispositivo interlocutor de terceros).
	Interfaz	Nombre asignado a las interfaces Nota: Puede modificar la conexión cambiando las interfaces local y del interlocutor.
	Tipo de interfaz	Tipo de interfaz
	Nombre de subred	Nombre asignado a las subredes
	Dirección	Direcciones IP asignadas Nota: Puede especificar la dirección remota de un dispositivo de terceros para un interlocutor "no especificado".
	ID de conexión	Número de ID: es generado automáticamente por la asignación de parámetros de conexión GET/PUT
	Nombre de conexión	Ubicación de los datos de las CPU local e interlocutora: es generado automáticamente por la asignación de parámetros de conexión GET/PUT
	Iniciativa local	Casilla de verificación para seleccionar la CPU local como la conexión activa
	Unilateral	Casilla de verificación para especificar una conexión de uno o dos sentidos; solo lectura Nota: en una conexión GET/PUT PROFINET, tanto el dispositivo local como el interlocutor pueden actuar de servidor o cliente. Esto hace posible una conexión de dos sentidos y la casilla de verificación "Unilateral" está deseleccionada. En una conexión GET/PUT PROFIBUS, en algunos casos el dispositivo interlocutor solo puede actuar de servidor (por ejemplo, un S7-300) y la casilla de verificación "Unilateral" está marcada.

Parámetro de ID de conexión

Hay tres formas de cambiar las ID de conexión definidas por el sistema:

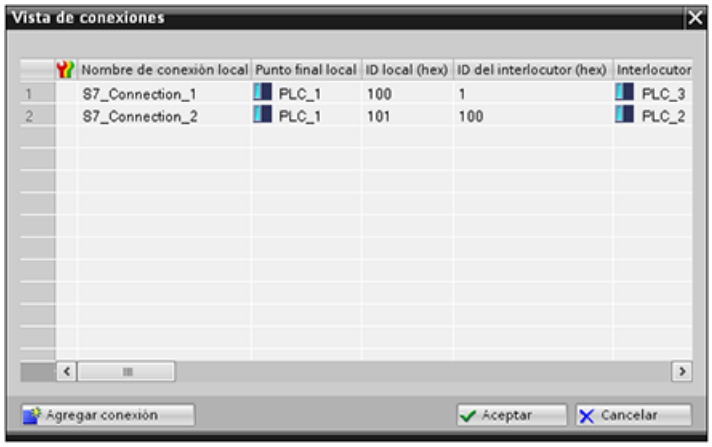
1. El usuario puede cambiar la ID actual directamente en el bloque GET/PUT. Si la ID nueva pertenece a una conexión ya existente, la conexión se modifica.
2. El usuario puede cambiar la ID actual directamente en el bloque GET/PUT, pero la ID nueva no existe. el sistema crea una nueva conexión S7.
3. El usuario puede cambiar la ID actual en el cuadro de diálogo "Vista de conexiones": la entrada del usuario se sincroniza con el parámetro de ID en el bloque GET/PUT correspondiente.

Nota

El parámetro "ID" del bloque GET/PUT no es un nombre de conexión, sino una expresión numérica que se escribe como el ejemplo siguiente: W#16#1

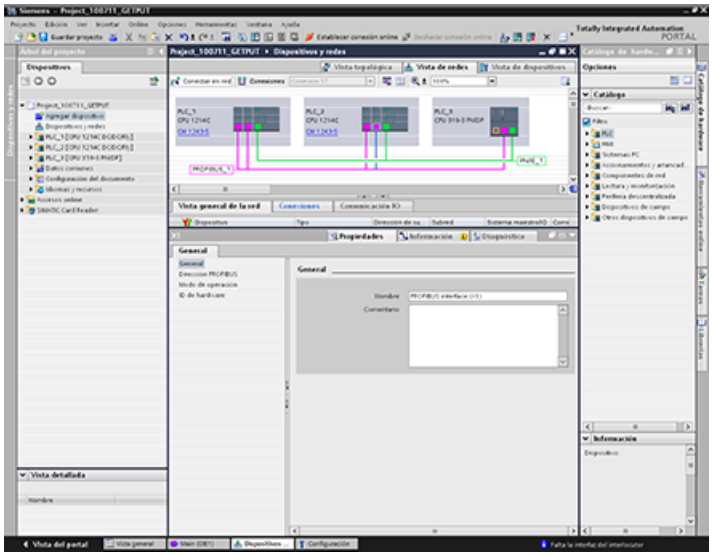
Parámetro de nombre de conexión

El nombre de la conexión se puede editar mediante un control de usuario especial, el cuadro de diálogo "Vista de conexiones". Este cuadro de diálogo ofrece todas las conexiones S7 disponibles que se pueden seleccionar como alternativa para la comunicación GET/PUT actual. El usuario puede crear una conexión completamente nueva en esta tabla. Haga clic en el botón a la derecha del campo "Nombre de conexión" para iniciar el cuadro de diálogo "Vista de conexiones".



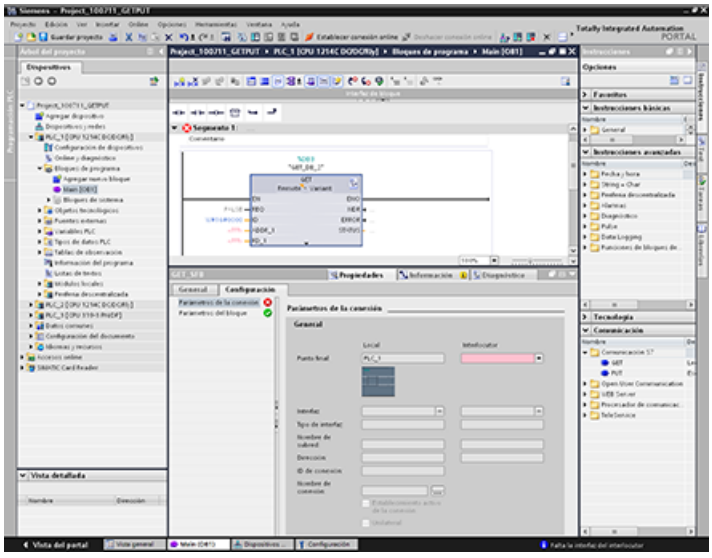
11.8.4.2 Configurar una conexión S7 de CPU a CPU

Dada la configuración de PLC_1, PLC_2 y PLC_3 mostrada en la figura siguiente, inserte bloques GET o PUT para "PLC_1".



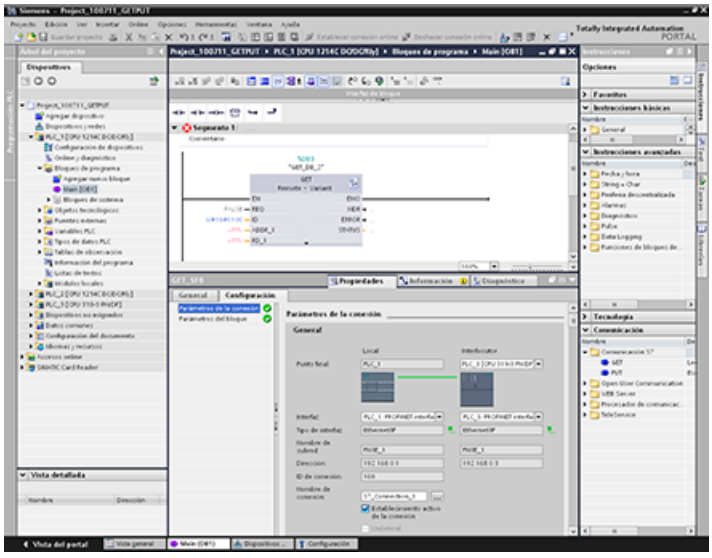
Para la instrucción GET o PUT, la ficha "Propiedades" se muestra automáticamente en la ventana de inspección con las siguientes selecciones de menú:

- "Configuración"
- "Parámetros de la conexión"



Configurar una conexión S7 PROFINET

Para el "Punto final del interlocutor", seleccione "PLC_3".



El sistema reacciona con los cambios siguientes:

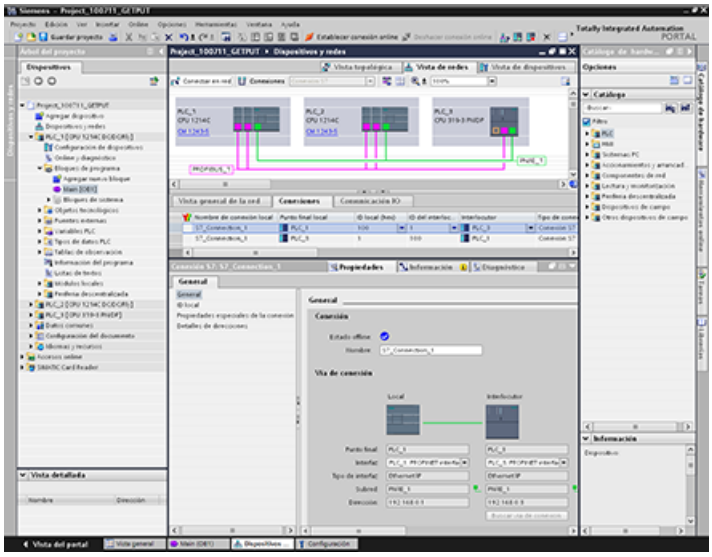
Tabla 11-74 Parámetro de conexión: valores generales

Parámetro		Definición
Parámetro de conexión: General	Punto final	"Punto final local" contiene "PLC_1" de solo lectura. El campo "Punto final del interlocutor" contiene "PLC_3[CPU319-3PN/DP]": • El color cambia de rojo a blanco. • Aparece la imagen del dispositivo "interlocutor". • Aparece una línea de conexión entre las imágenes de los dispositivos PLC_1 y PLC_3 (línea Ethernet verde).
	Interfaz	La "Interfaz local" contiene "CPU1214C DC/DC/DC, interfaz PROFINET (R0/S1)". La "Interfaz del interlocutor" contiene: "CPU319-3PN/DP, interfaz PROFINET (R0/S2)".
	Tipo de interfaz	El "Tipo de interfaz local" contiene "Ethernet/IP"; el control es de solo lectura. El "Tipo de interfaz del interlocutor" contiene "Ethernet/IP"; el control es de solo lectura. Las imágenes del tipo de interfaz se muestran a la derecha, junto al "Tipo de interfaz" local y del interlocutor (icono Ethernet verde).
	Nombre de subred	El "Nombre de subred local" contiene "PN/IE_1"; el control es de solo lectura. El "Nombre de subred del interlocutor" contiene "PN/IE_1"; el control es de solo lectura.
	Dirección	La "Dirección local" contiene la dirección IP local; el control es de solo lectura. La "Dirección del interlocutor" contiene la dirección IP del interlocutor; el control es de solo lectura.
	ID de conexión	"ID de conexión" contiene "100". En el editor de programas, en el [OB1] principal, el valor "ID de conexión" del bloque GET/PUT también contiene "100".
	Nombre de conexión	El "Nombre de conexión" contiene el nombre predeterminado de la conexión (por ejemplo, "S7_Connection_1"); el control está habilitado.
	Iniciativa local	Marcada y habilitada para seleccionar la CPU local como conexión activa.
	Unilateral	Solo lectura y sin marcar. Nota: Tanto "PLC_1" (una CPU S7-1200 1214CDC/DC/relé) como "PLC_3" (una CPU S7-300 319-3PN/DP) pueden actuar de servidor y cliente en una conexión GET/PUT PROFINET, lo que hace posible una conexión en ambos sentidos.

El icono GET/PUT del árbol Vista de propiedades también cambia de rojo a verde.

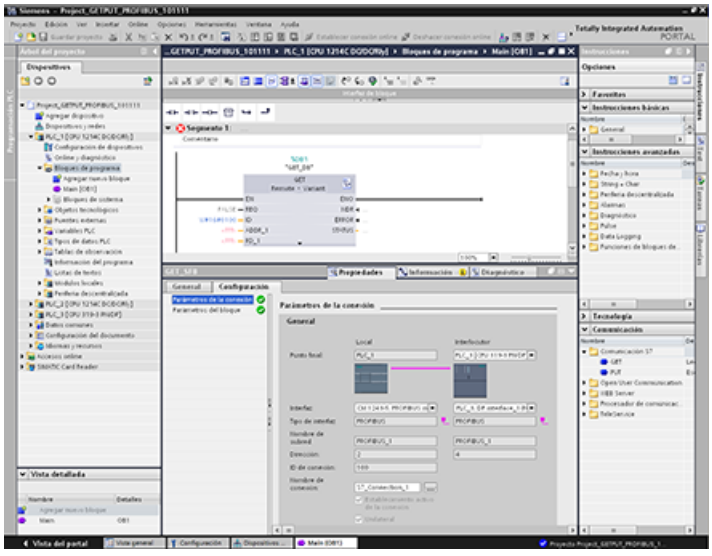
Conexión S7 PROFINET completada

En la "Vista de redes" se muestra una conexión S7 bilateral en la tabla "Conexiones" entre "PLC_1" y "PLC_3".



Configurar una conexión S7 PROFIBUS

Para el "Punto final del interlocutor", seleccione "PLC_3".



El sistema reacciona con los cambios siguientes:

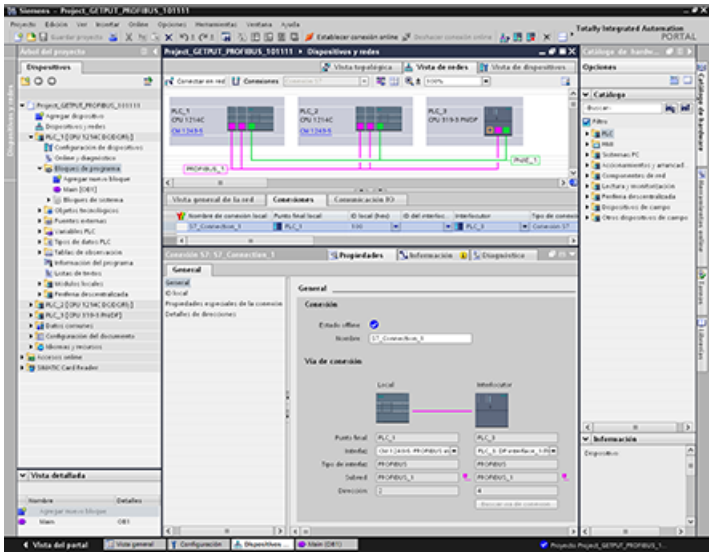
Tabla 11-75 Parámetro de conexión: valores generales

Parámetro		Definición
Parámetro de conexión: General	Punto final	"Punto final local" contiene "PLC_1" de solo lectura. El campo "Punto final del interlocutor" contiene "PLC_3[CPU319-3PN/DP]": • El color cambia de rojo a blanco. • Aparece la imagen del dispositivo "interlocutor". • Aparece una línea de conexión entre las imágenes de los dispositivos PLC_1 y PLC_3 (línea PROFIBUS púrpura).
	Interfaz	La "Interfaz local" contiene "CPU1214C DC/DC/DC, interfaz PROFIBUS (R0/S1)". La "Interfaz del interlocutor" contiene: "CPU319-3PN/DP, interfaz PROFIBUS (R0/S2)".
	Tipo de interfaz	El "Tipo de interfaz local" contiene "PROFIBUS"; el control es de solo lectura. El "Tipo de interfaz del interlocutor" contiene "PROFIBUS"; el control es de solo lectura. Las imágenes del tipo de interfaz se muestran a la derecha, junto al "Tipo de interfaz" local y del interlocutor (icono PROFIBUS púrpura).
	Nombre de subred	El "Nombre de subred local" contiene "PROFIBUS_1"; el control es de solo lectura. El "Nombre de subred del interlocutor" contiene "PROFIBUS_1"; el control es de solo lectura.
	Dirección	La "Dirección local" contiene la dirección IP local; el control es de solo lectura. La "Dirección del interlocutor" contiene la dirección IP del interlocutor; el control es de solo lectura.
	ID de conexión	"ID de conexión" contiene "100". En el editor de programas, en el [OB1] principal, el valor "ID de conexión" del bloque GET/PUT también contiene "100".
	Nombre de conexión	El "Nombre de conexión" contiene el nombre predeterminado de la conexión (por ejemplo, "S7_Connection_1"); el control está habilitado.
	Iniciativa local	De solo lectura, marcada y habilitada para seleccionar la CPU local como conexión activa.
	Unilateral	De solo lectura y marcada. Nota: "PLC_3" (una CPU S7-300 319-3PN/DP) solo puede actuar de servidor (no puede ser cliente) en una conexión GET/PUT PROFIBUS, lo que permite únicamente una conexión de sentido único.

El icono GET/PUT del árbol Vista de propiedades también cambia de rojo a verde.

Conexión S7 PROFIBUS completada

En la "Vista de redes" se muestra una conexión S7 unilateral en la tabla "Conexiones" entre "PLC_1" y "PLC_3".



11.9 Qué hacer cuando no se puede acceder a la CPU a través de la dirección IP

Si no se puede acceder a una CPU a través de la dirección IP, se puede ajustar una dirección IP de emergencia (temporalmente) para la CPU. La dirección IP de emergencia permite restablecer la comunicación con la CPU para descargar una configuración de dispositivo con una dirección IP válida.

Razones por las que puede ser necesaria una dirección IP de emergencia

La CPU puede no estar accesible si alguien ha descargado un proyecto con uno de los siguientes problemas:

- La dirección IP de la interfaz PROFINET de la CPU es un duplicado de otro dispositivo de la red.
- La subred no es correcta para la CPU.
- La máscara de subred hace que la CPU sea inaccesible.

En estos casos, la CPU deja de estar accesible desde STEP 7.

Asignación de una dirección IP de emergencia

Se puede asignar una dirección IP de emergencia en las siguientes condiciones:

- En la configuración de dispositivos de STEP 7 se ha seleccoinado "Ajustar dirección IP en el proyecto" para el protocolo IP.
- La CPU está en modo STOP.

En estas condiciones se puede utilizar una herramienta DCP para ajustar una dirección IP de emergencia para la dirección IP del dispositivo. La SIMATIC Automation Tool, por ejemplo, tiene un comando DCP Set IP address. Se puede ajustar una dirección IP de emergencia independientemente del nivel de protección (Página 163) de la CPU. Tras ajustar una dirección IP temporal con una herramienta DCP, el LED de mantenimiento de la CPU se enciende. El búfer de diagnóstico también incluye una entrada que indica que se ha habilitado una dirección de emergencia de una interfaz Ethernet.

Restauración de una dirección IP tras asignar una dirección IP de emergencia

El búfer de diagnóstico informa de cuándo se ha habilitado o deshabilitado una dirección IP de emergencia. La dirección IP de emergencia se puede inicializar con una desconexión/conexión (Power OFF-Power ON).

Después de haber asignado una dirección IP de emergencia se puede descargar un proyecto STEP 7 con una dirección IP válida para la CPU. Tras descargar el proyecto, ejecute una desconexión/conexión (Power OFF - Power ON) de la CPU.

11.10 Servidor OPC UA

Las CPU S7-1200 soportan el OPC UA Micro-Embedded Profile. Además, las CPU S7-1200 soportan la autenticación de usuario OPC UA, la seguridad de comunicación, subscripciones y la lectura y escritura de variables de programa. La funcionalidad OPC UA no indicada en el Micro Embedded Profile de OPC UA, o bien indicada de otro modo, no se soporta necesariamente. Las capacidades y limitaciones se describen en los siguientes apartados.

Consulte también

OPC Foundation (https://opcfoundation.org/)

11.10.1 Configuración del servidor OPC UA

El servidor OPC UA se puede configurar en las propiedades de hardware de la CPU, en TIA Portal.

Para configurar el servidor OPC UA, proceda del siguiente modo:

1. Seleccione la pestaña General en la vista de configuración de dispositivos.

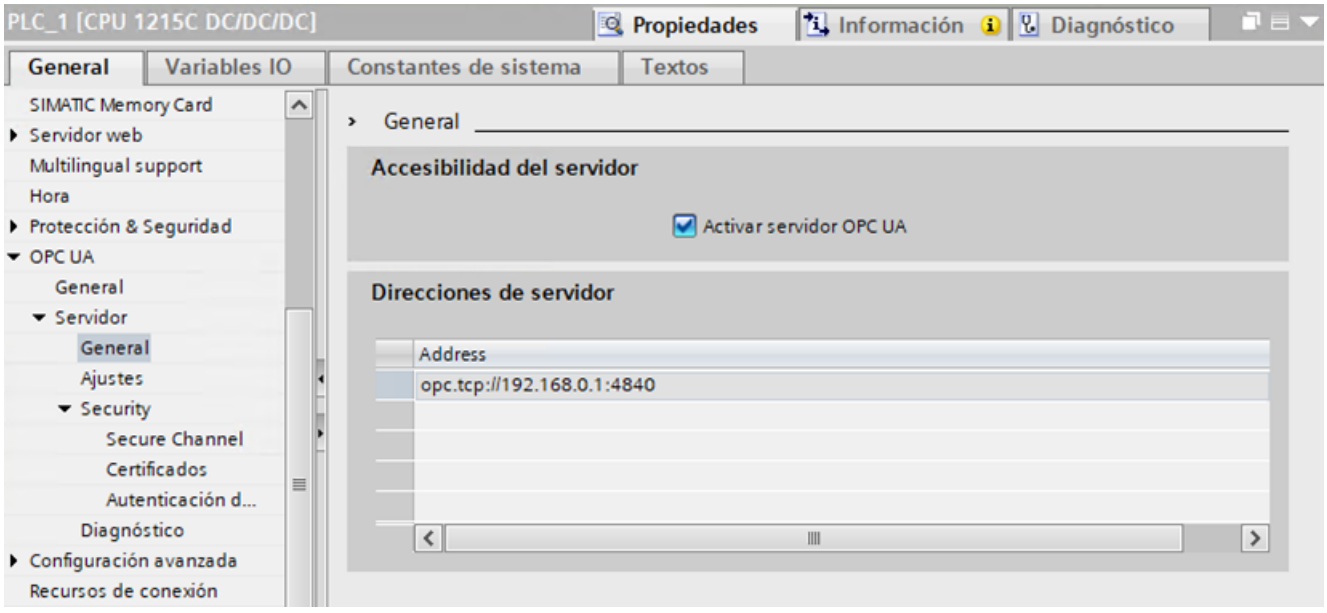
- En la ventana General, seleccione "OPC UA".
  - En la ventana OPC UA > Servidor se puede configurar lo siguiente:
    - Configuración general
    - Opciones
    - Seguridad
    - Diagnóstico

11.10.1.1 Activación del servidor OPC UA

El servidor OPC UA está desactivado por defecto. El servidor OPC UA debe activarse seleccionando la casilla de verificación "Activar servidor OPC UA" en las propiedades de hardware de la CPU.

Para activar el servidor OPC UA, proceda del siguiente modo:

1. Seleccione la pestaña General en la vista de configuración de dispositivos.
2. En la ventana General, seleccione "OPC UA".
3. En la ventana OPC UA > Servidor > General, active la casilla de verificación "Activar servidor OPC UA".



11.10.1.2 Comportamiento del servidor OPC UA durante el funcionamiento

Comportamiento del servidor OPC UA durante el funcionamiento

El servidor OPC UA de la CPU S7-1200 arranca cuando se activa el servidor y se descarga el proyecto en la CPU.

Comportamiento en modo STOP de la CPU

Un servidor OPC UA activado permanece operativo incluso si la CPU cambia a "STOP". El servidor OPC UA sigue respondiendo a las solicitudes de clientes OPC UA.

La respuesta del servidor en detalle:

- Si se solicitan los valores de variables PLC, se obtienen los valores que eran actuales antes de que la CPU cambiara o fuera puesta a "STOP".
- Si se escriben valores en el servidor OPC UA, el servidor OPC UA los acepta. No obstante, la CPU no procesa los valores porque el programa de usuario no se ejecuta en modo "STOP". Un cliente OPC UA puede sin embargo leer los valores escritos en STOP del servidor OPC UA de la CPU.

Carga de la CPU con el servidor OPC UA en funcionamiento

Si se carga una CPU con el servidor OPC UA en funcionamiento, puede que sea necesario parar y rearrancar el servidor dependiendo de los objetos cargados. En ese caso, las conexiones activas se interrumpen y deben restablecerse una vez rearrancado el servidor.

La duración del rearranque depende principalmente de los parámetros siguientes:

- El alcance de la estructura de datos
- El número de variables visibles en el espacio de direcciones OPC UA
- El ajuste de la definición de tipos de datos compatibles hacia abajo de acuerdo con la especificación OPC UA correspondiente a V1.03 (TypeDictionary activado)
- Ajustes para la carga de comunicación y el tiempo de ciclo mínimo (Página 153)

Con las versiones de firmware anteriores a V4.5, el servidor OPC UA se paraba con cada descarga en la CPU y, seguidamente, rearrancaba.

A partir de la versión de firmware V4.5, este comportamiento del servidor OPC UA se ha optimizado del siguiente modo:

- Cuando se descargan objetos en estado operativo STOP de la CPU, el servidor OPC UA todavía se sigue parando y rearrancando seguidamente. En ese caso, STEP 7 no muestra ninguna advertencia.
- Cuando se descargan objetos en estado operativo RUN de la CPU, el servidor OPC UA solo se para si los objetos descargados son relevantes para OPC UA. El servidor OPC UA rearranca tras una reinicialización debida a los datos OPC UA modificados.
  Antes de que los objetos relevantes para OPC UA se carguen en la CPU y se pare el servidor OPC UA, STEP 7 muestra una advertencia en la vista preliminar de la carga. A continuación puede decidir si el rearranque de un servidor es compatible con el proceso en curso o si prefiere cancelar la descarga. Estas advertencias solo se muestran si el servidor OPC UA está funcionando. Si el servidor OPC UA está desactivado, los datos OPC UA modificados no influyen en el proceso de descarga.

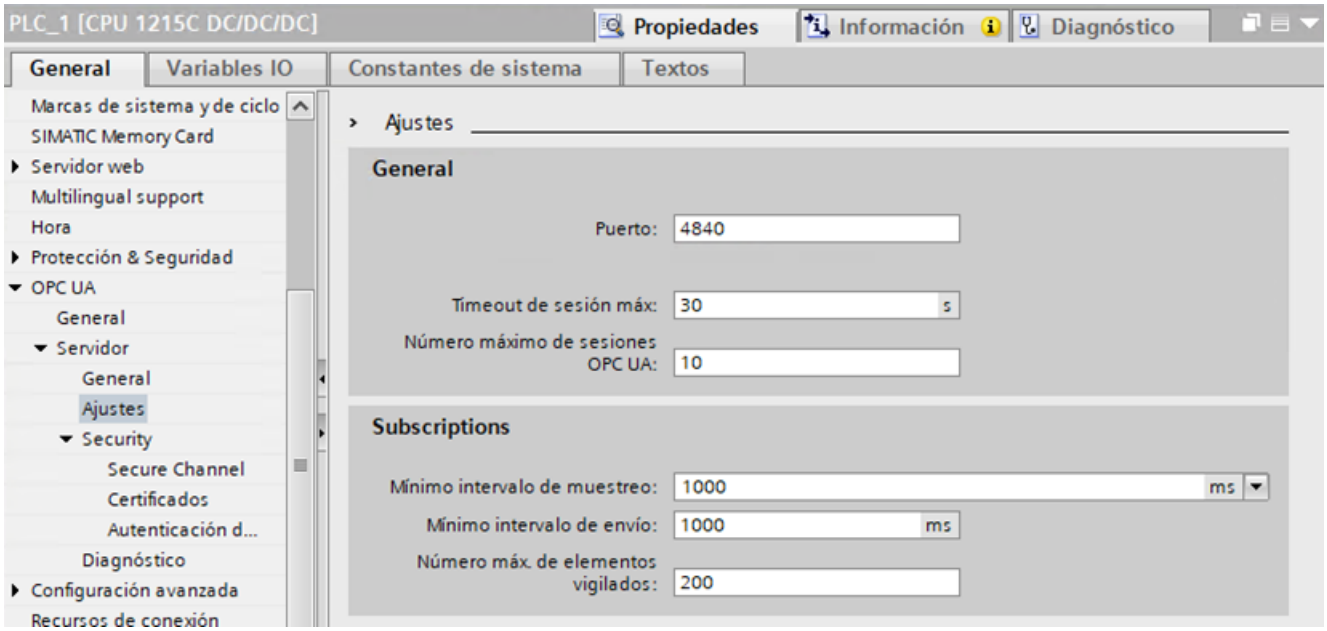
Ejemplos

- Solo desea agregar otro bloque lógico al programa.
  Los bloques de datos, entradas, salidas, marcas, temporizadores o contadores no se ven afectados.
  Reacción durante la carga: un servidor OPC UA en funcionamiento no se interrumpe.
- Desea cargar un módulo de datos nuevo y ha marcado el módulo de datos como no relevante para OPC UA.
  Reacción durante la carga: un servidor OPC UA en funcionamiento no se interrumpe.
- Desea sobrescribir un módulo de datos.
  Reacción durante la carga: Aparece una advertencia de que el servidor rearrancará.
  Explicación: STEP 7 no puede determinar si los cambios hacen referencia a datos relevantes para OPC UA o no.

Determinar el estado operativo de la CPU a través del servidor OPC UA

11.10.1.3 Configuración del servidor OPC UA

El cuadro de diálogo Opciones muestra los ajustes de OPC UA que pueden configurarse.



La tabla de ajustes del servidor OPC UA proporciona información adicional sobre los ajustes que se configuran para el servidor OPC UA. La tabla también describe las limitaciones del servidor para el firmware V4.5 de S7-1200 o posteriores.

Los valores predeterminados de cada columna trabajan con la carga por comunicación predeterminada de la CPU del 20 %. Si se especifican los parámetros de carga máxima, el

porcentaje de carga por comunicación de la CPU deberá incrementarse. Esto depende de la carga impuesta por otras actividades de comunicación (por ejemplo, PROFINET).

Tabla 11-76 Configuración del servidor OPC UA

Configuración del servidor	Mín.	Máx.	Valor predeterminado	¿Ajustable por el usuario?
Puertos de comunicación	1024	49151	4840	Sí
Timeout de sesión máx.	1 s	600 000 s	30 s	Sí
Sesiones simultáneas	1	10	10	Sí
Nodos máx.	100	2000		
Nodos en la interfaz del servidor[1]		1000		No
Espacios de nombres en total[2]		18		No
Métodos de servidor		10		No
Número máx. de DB[3] de instancia de métodos de servidor		20	20	No
Número máx. de argumentos de entrada de métodos de servidor		20	20	No
Número máx. de argumentos de salida de métodos de servidor		20	20	No
Limitaciones de suscripción				
Intervalo mín. de muestreo	100 ms, 250 ms, 500 ms, 1000 ms, 5000 ms, 10 000 ms		1000 ms	Sí
Intervalo mín. de publicación	200 ms	20.000.000 ms	1000 ms	Sí
Suscripciones simultáneas		50	50	No
Máx. suscripciones por sesión		5		No
Nodos vigilados como máx.	100	1000	200	Sí

[1] Este es el número máximo de nodos definidos por el usuario. Aquí se incluyen los nodos que se definen implícitamente en la interfaz del servidor. El número máximo no incluye los nodos definidos con la interfaz de servidor SIMATIC.

[2] Es el número total de espacios de nombres definidos para todas las interfaces de servidor y espacios de nombres de referencia.

[3] Es el número total de DB de instancia de métodos de servidor permitido para todas las interfaces de servidor y todos los métodos de servidor asociados. Esta limitación viene impuesta tanto por el sistema de ingeniería (en tiempo de compilación) como por el sistema de automatización (en la descarga).

11.10.1.4 Uso del S7-1200 como servidor OPC UA

Configuración de la interfaz de servidor OPC UA

Al usar interfaces de servidor OPC UA deben respetarse los límites de los objetos siguientes de acuerdo con el rendimiento de la CPU S7-1200.

Nota sobre los límites de configuración al usar las interfaces de servidor

- Número de interfaces de servidor
- Número de nodos OPC UA
- Si ha implementado métodos: número de métodos de servidor o instancias de métodos de servidor

Límites de configuración de las interfaces y métodos del servidor OPC UA

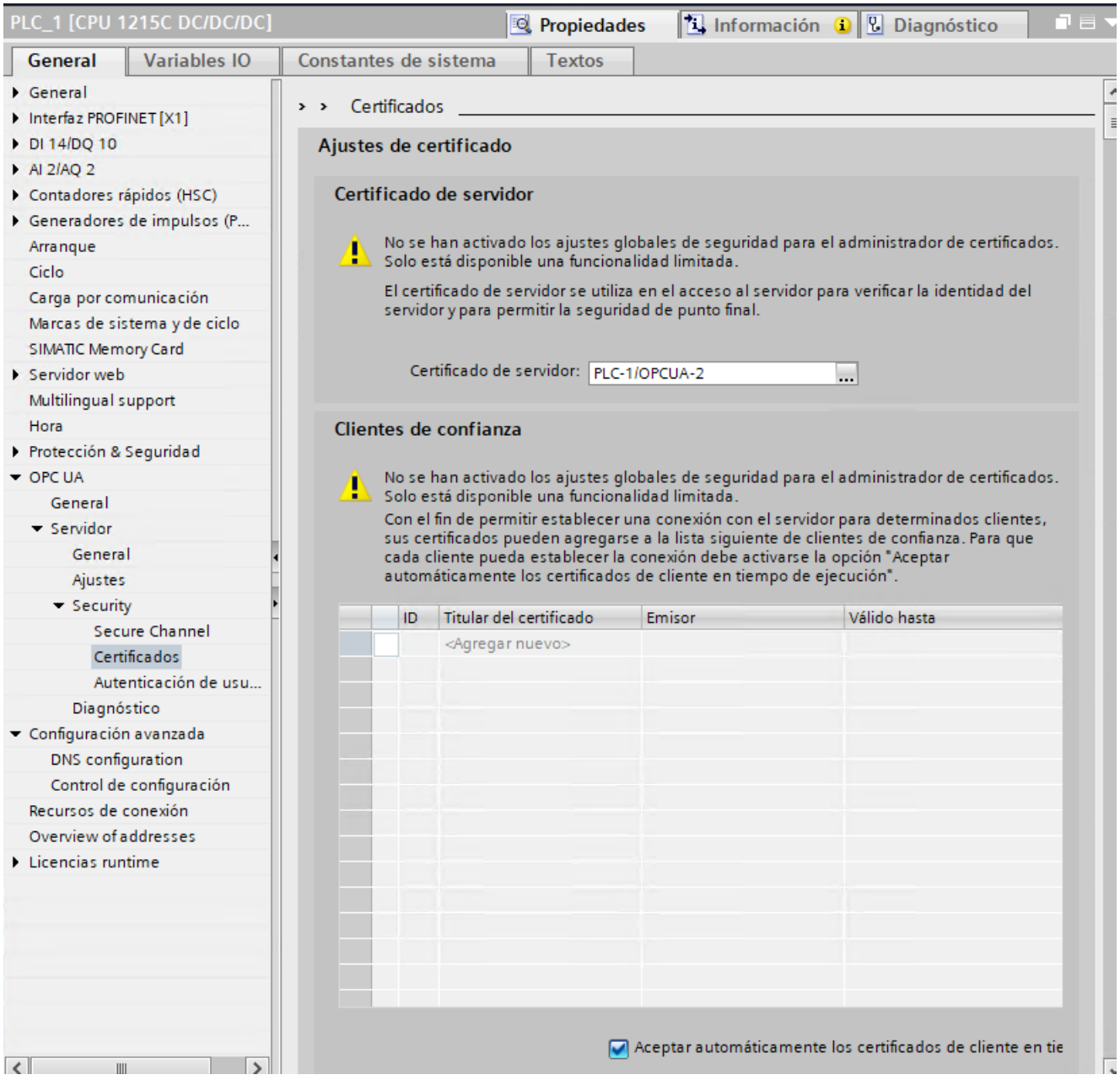
La tabla siguiente indica los límites de configuración de las CPU S7-1200; estos también deben tenerse en cuenta al compilar y cargar una configuración. Una infracción de los límites de configuración genera un mensaje de error.

Valor de especificación técnica	
CPU S7-1200	CPU 1211C DC/DC/DC 6ES7211-1AE40-0XB0 CPU 1211C AC/DC/Relay 6ES7211-1BE40-0XB0 CPU 1211C DC/DC/Relay 6ES7211-1HE40-0XB
	CPU 1212C DC/DC/DC 6ES7212-1AE40-0XB0 CPU 1212C AC/DC/Relay 6ES7212-1BE40-0XB0 CPU 1212C DC/DC/Relay 6ES7212-1HE40-0XB
	CPU 1214C DC/DC/DC 6ES7214-1AG40-0XB0 CPU 1214C AC/DC/Relay 6ES7214-1BG40-0XB0 CPU 1214C DC/DC/Relay 6ES7214-1HG40-0XB0
	CPU 1215C DC/DC/DC 6ES7215-1AG40-0XB0 CPU 1215C AC/DC/Relay 6ES7215-1BG40-0XB0 CPU 1215C DC/DC/Relay 6ES7215-1HG40-0XB0
	CPU 1217C DC/DC/DC 6ES7217-1AG40-0XB0
CPU S7-1200 de seguridad	CPU 1212FC DC/DC/DC 6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/Relay 6ES7212-1HF40-0XB0
	CPU 1214FC DC/DC/DC 6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/Relay 6ES7214-1HF40-0XB0
	CPU 1215FC DC/DC/DC 6ES7215-1AF40-0XB0
	CPU 1215FC DC/DC/Relay 6ES7215-1HF40-0XB0
Uso de especificaciones Companion importadas (modelos de información)	
Número máx. de interfaces de servidor OPC UA:	
• Tipo "Especificación Companion"	2
• Tipo "Espacio de nombres de referencia"	10
• Tipo "Interfaz de servidor"	2
Número máx. de nodos OPC UA en interfaces de servidor personalizadas	2000
Aprovisionamiento de métodos	
Número máx. de métodos de servidor utilizables o número máx. de instancias de métodos de servidor (instrucciones OPC_UA_Server-MethodPre, OPC_UA_Server- MethodPost)	20

11.10.2 Seguridad del servidor OPC UA

La seguridad de servidor solo es uno de los métodos de seguridad utilizados para asegurar tecnología. Otros métodos utilizados para asegurar tecnología incluyen la seguridad TIAP y la seguridad de PLC.

El servidor OPC UA requiere un certificado para activarse. TIA Portal genera automáticamente un certificado cuando se activa el servidor. Dicho certificado puede cambiarse en las propiedades del PLC.



.

Nota

Límite de certificados en S7-1200

S7-1200 tiene un límite de certificados del sistema de 64.

Todos los certificados se incluyen en este número (por ejemplo, certificados web, certificados OPC UA y certificados OUC).

Si hay más de 64 certificados, TIA Portal muestra un error que indica que se ha excedido el número máximo de 64 certificados. Habrá que eliminar algunos certificados de la configuración del PLC.

11.10.2.1 Directivas de seguridad soportadas

Existe la posibilidad de seleccionar una de las directivas de seguridad que soporta el servidor. La directiva de seguridad que haya seleccionado en runtime determina la seguridad de la comunicación entre cliente y servidor.

Para seleccionar la directiva de seguridad OPC UA, proceda del siguiente modo:

1. Seleccione la pestaña General en la vista de configuración de dispositivos.
2. En la ventana General, seleccione "OPC UA".
3. En la ventana OPC UA > Servidor > Seguridad, seleccione "Secure Channel".
4. Agregue la directiva de seguridad soportada en el campo "Certificado de servidor".

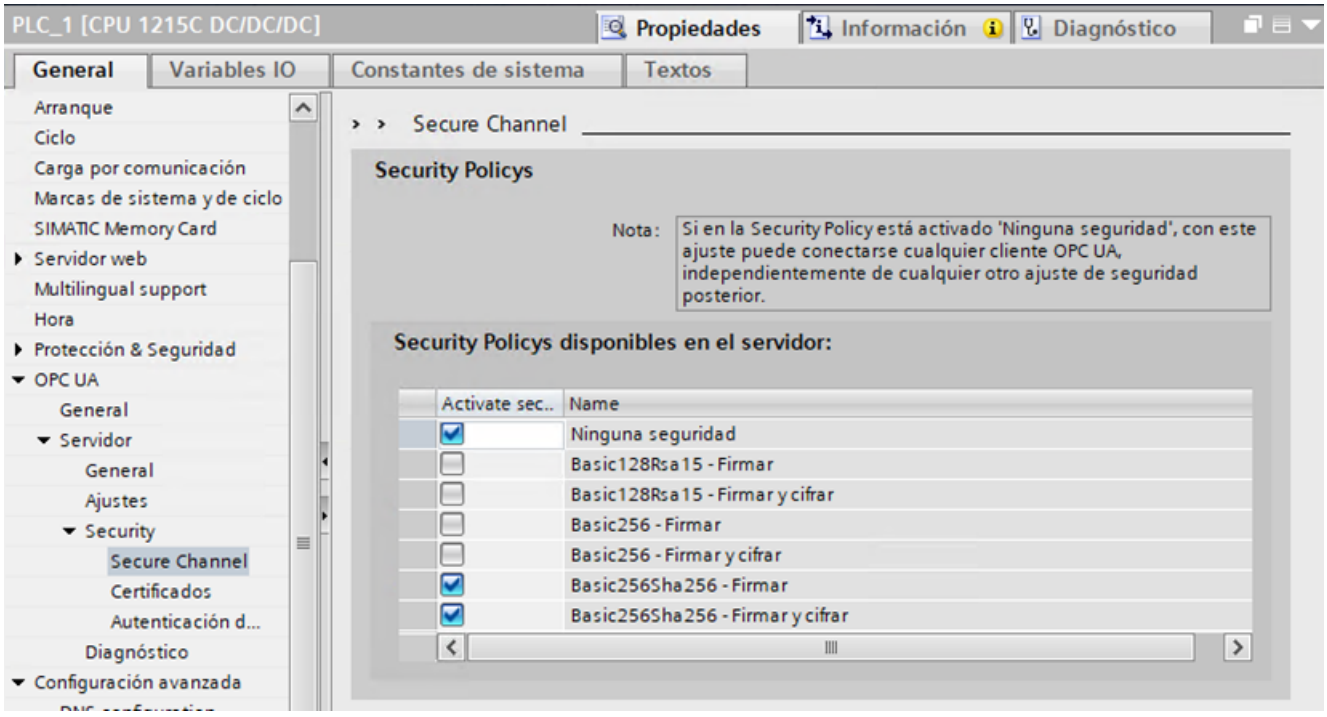


Tabla 11-77 Directivas de seguridad OPC UA soportadas por S7-1200

Directiva de seguridad	¿Activada por defecto?
Sin seguridad	Sí
Basic128Rsa15 – firmar	No
Basic128Rsa15 – firmar y cifrar	No
Basic256 – firmar	No
Basic256 – firmar y cifrar	No
Basic256Sha256 – firmar	Sí
Basic256Sha256 – firmar y cifrar	Sí

Nota

Establecer una conexión OPC UA segura

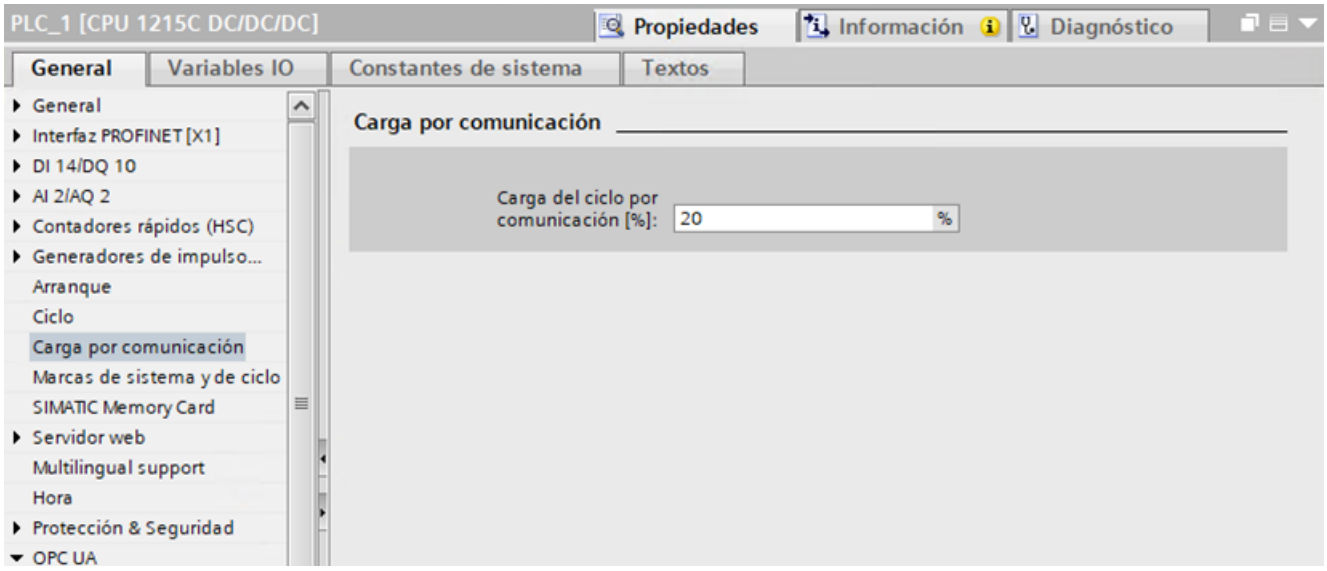
Las directivas de seguridad OPC UA más complejas pueden requerir un porcentaje de carga de comunicación mayor que el predeterminado para mantener las conexiones online con dispositivos externos, como TIA Portal, HMI, etc.

Así, por ejemplo, al establecer una conexión entre un cliente OPC UA y el servidor OPC UA del S7-1200, es posible que se interrumpa una conexión online de TIA Portal. Esto se debe a los complejos cálculos iniciales necesarios para establecer la conexión segura, lo que puede provocar un rebase de tiempo de la conexión online de TIA Portal.

Cabe destacar que la seguridad aumentada no afecta negativamente a las comunicación cliente/servidor OPC UA más allá del establecimiento inicial de la conexión.

Una solución a este problema consiste simplemente en restablecer la conexión online. Otra solución es aumentar el porcentaje de carga de comunicación predeterminado de la forma indicada a continuación.

No obstante, recuerde que el aumento de la carga de comunicación predeterminada aumentará proporcionalmente el tiempo de ciclo.

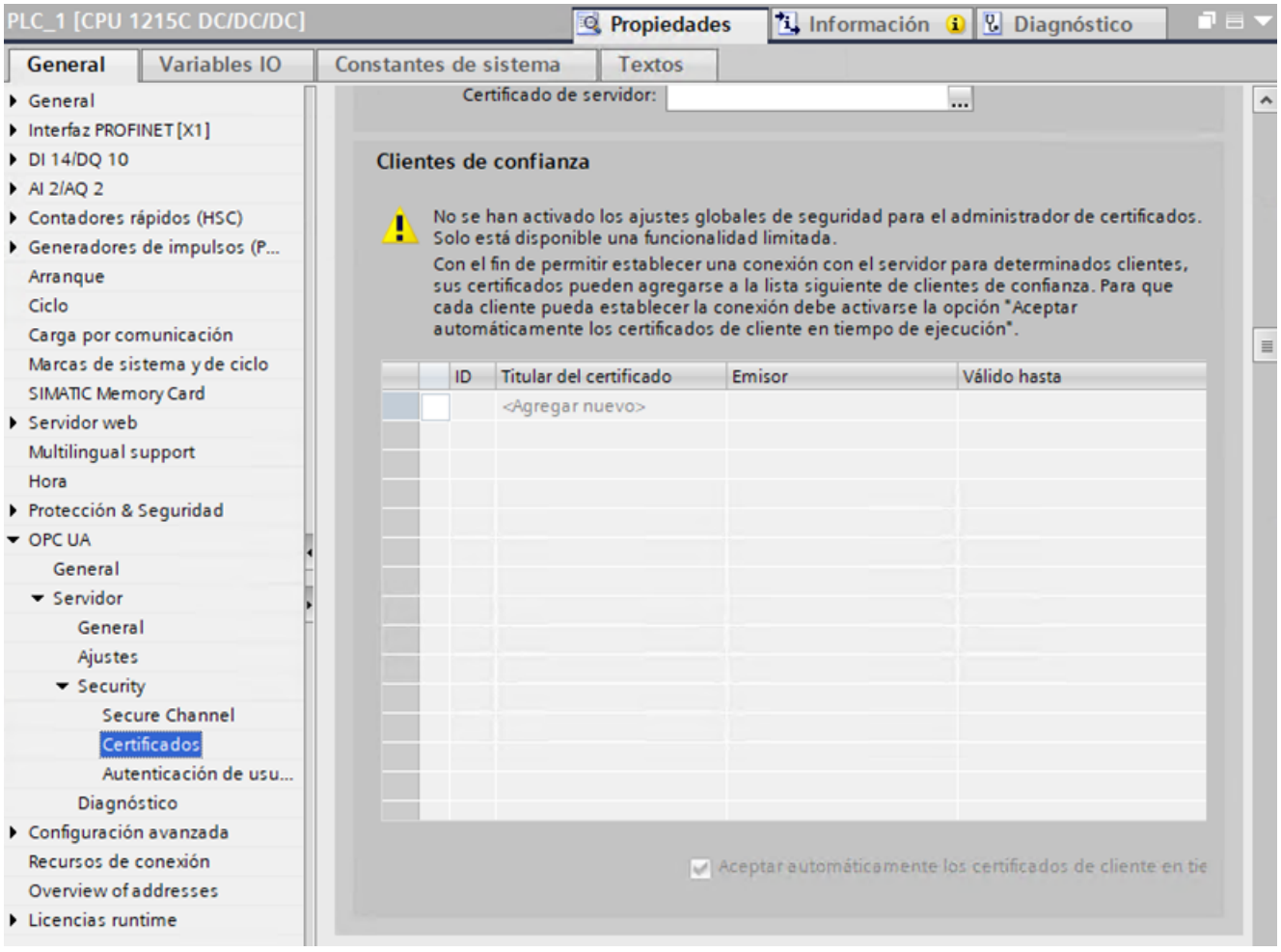


11.10.2.2 Clientes de confianza

El servidor OPC UA puede configurarse para permitir únicamente conexiones de clientes de confianza. En el ajuste predeterminado, el servidor está configurado para aceptar automáticamente certificados de cliente.

El usuario define la lista de clientes de confianza, que se identifican por sus certificados. Si está seleccionado, solo los clientes que presentan certificados de confianza en tiempo de ejecución tienen permiso para conectarse a un servidor.

Para especificar clientes de confianza, agregue sus certificados a la lista "Clientes de confianza" en TIA Portal, en "Propiedades de hardware > Seguridad OPC UA > Secure Channel > Clientes de confianza".



11.10.2.3 Autenticación de usuario

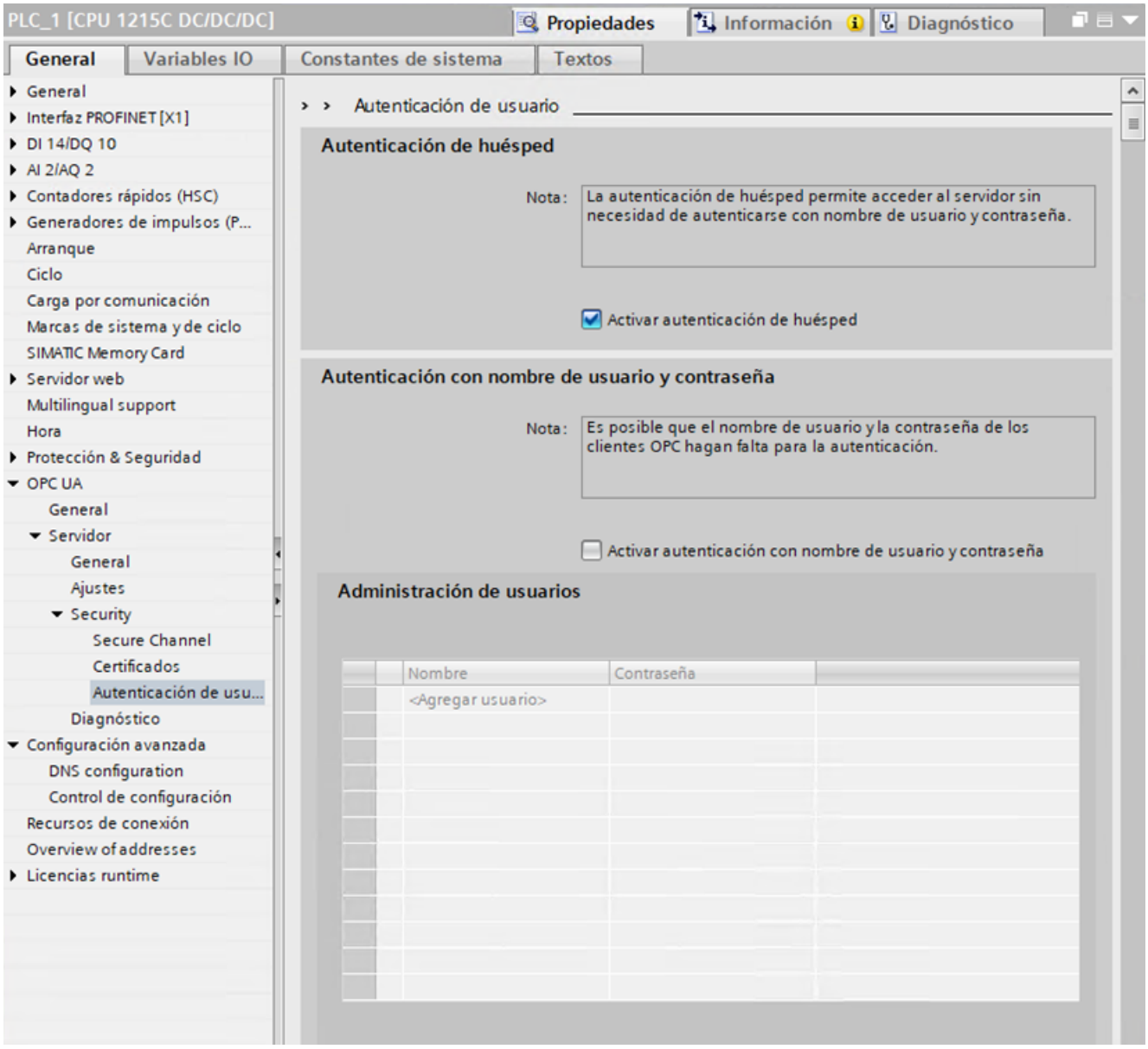
El S7-1200 soporta tanto la autenticación de huéspedes como la autenticación con nombre de usuario y contraseña. El ajuste predeterminado es la autenticación de huéspedes.

Si la autenticación de huéspedes está activada, el cliente no debe proporcionar un nombre de usuario ni una contraseña en el momento de conectarse.

Si la autenticación de huéspedes está desactivada, solo se permite conectar a clientes que proporcionan un nombre de usuario y una contraseña definidos previamente.

Los usuarios válidos para el servidor OPC UA se definen de dos formas:

- Definir usuarios válidos en las propiedades de hardware de la CPU.
- Si se ha activado la seguridad global para el proyecto, los usuarios válidos se identifican mediante las funciones especificadas para usuarios individuales del proyecto TIA Portal. Es posible configurar un máximo de 21 usuarios.



11.10.3 Interfaz de servidor OPC UA

El servidor OPC UA de S7-1200 soporta la interfaz de servidor SIMATIC estándar. Sin embargo, no soporta la "publicación" automática de variables de CPU y DB con esta selección. En su lugar deben definirse la estructura y el contenido de la interfaz de servidor en TIA Portal y, seguidamente, descargarse al PLC.

Para agregar una interfaz de servidor, proceda del siguiente modo:

1. En el árbol del proyecto haga clic en el nombre del PLC.
2. Seleccione "Comunicación OPC UA".
3. Seleccione "Interfaces de servidor".
4. "Agregar nueva interfaz de servidor".
5. Introduzca la información relevante para el servidor.
6. Descargue el servidor al PLC.

Al agregar una interfaz de servidor, tenga en cuenta que todas las variables disponibles están recogidas en los elementos OPC UA de la pantalla. Existe la posibilidad de arrastrar los elementos desde la ventana "Elementos OPC UA" a la ventana "Interfaz de servidor OPC UA". Hay una prueba de coherencia que verifica los contenidos de la interfaz de servicio. Además, la interfaz puede exportarse a un archivo XML.

11.10.3.1 Tipos de datos soportados

La OPC Foundation (https://opcfoundation.org/) define un conjunto de tipos de datos soportados que describen la estructura del atributo Value de variables y VariableTypes. El S7-1200 V4.5 soporta una parte de estos tipos de datos (Página 106), además de otros tipos definidos que se derivan de los primeros.

La tabla siguiente recoge los tipos de datos que soporta S7-1200 V4.5:

Tipo SIMATIC	Nombre de tipo OPC UA	Identificador de nodo
Bool	Boolean	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
DATE	UInt16	i=5
TOD	UInt32	i=7
TIME	Int32	i=6
DTL	Estructura	N/D

Recuerde que esta lista representa los tipos de nodos básicos soportados. No es una lista completa de los tipos de nodos soportados, pues muchos tipos de datos SIMATIC se basan en los tipos de nodos básicos. Todos los tipos de datos SIMATIC que se basen en un tipo de nodo básico también serán tipos de nodos soportados.

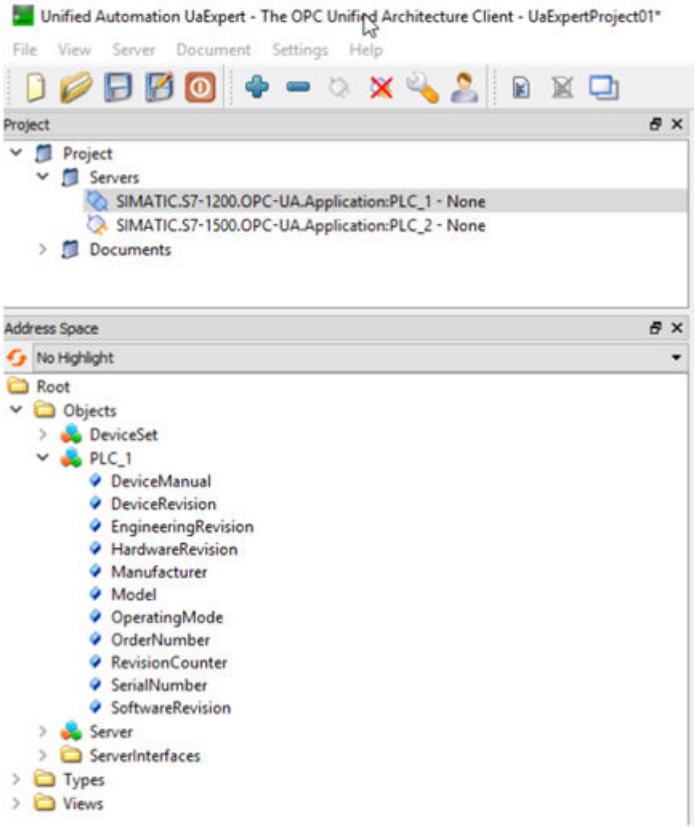
Las CPU S7-1200 V4.5 soportan métodos de servidor así como tipos de datos estructurados (estructuras y arrays).

Las CPU S7-1200 no soportan relaciones.

El S7-1200 acepta la descarga de un servidor con tipos no soportados. Sin embargo, el PLC devuelve un error si el cliente intenta leer o escribir en un nodo de un tipo no soportado.

11.10.3.2 Representación del PLC

La interfaz de servidor OPC UA proporciona nodos que representan el PLC con propiedades que describen el PLC. Están disponibles siempre que se activa el servidor OPC UA.



La información siguiente se ofrece en la interfaz de servidor SIMATIC estándar:

Atributo	**Valor**		
NodeId	Ns=SI;s=PLC		
BrowseName	SI:<PLC> Siendo <PLC> el nombre que el usuario ha asignado al PLC en su proyecto TIA Portal		
Referencias	**NodeClass**	**BrowseName**	**Comentario**

Atributo	Valor		
ComponentOf the DeviceSet Object			
OrganizedBy the Objects folder			
HasTypeDefinition	ObjectType	SimaticDeviceType	Derivado del tipo de dispositivo
HasProperty	Variable	DeviceManual	
HasProperty	Variable	DeviceRevision	
HasProperty	Variable	EngineeringRevision	
HasProperty	Variable	HardwareRevision	
HasProperty	Variable	Icon	
HasProperty	Variable	Manufacturer	
HasProperty	Variable	Model	
HasProperty	Variable	OperatingMode	
HasProperty	Variable	OrderNumber	
HasProperty	Variable	RevisionCounter	
HasProperty	Variable	SerialNumber	
HasProperty	Variable	SoftwareRevision	



11.10.3.3 Interfaces de servidor descargables

Los componentes de la interfaz de servidor OPC UA se crean y editan en TIA Portal. Las interfaces de servidor OPC UA se definen de dos formas:

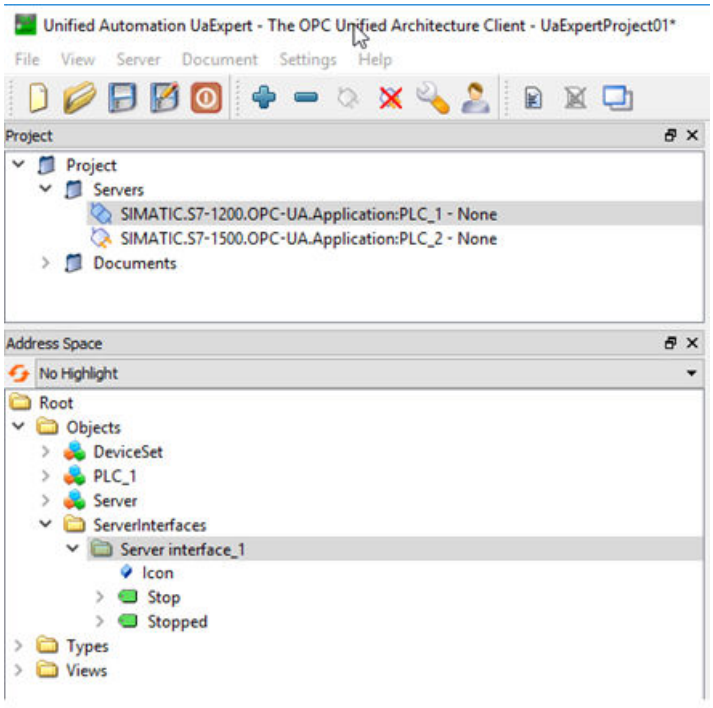
- Archivos XML creados externamente con la especificación Companion (conocidos como especificaciones Companion)
- Interfaz de servidor definida directamente en TIA Portal a partir de elementos de bloques de datos y variables globales incluidas en el programa

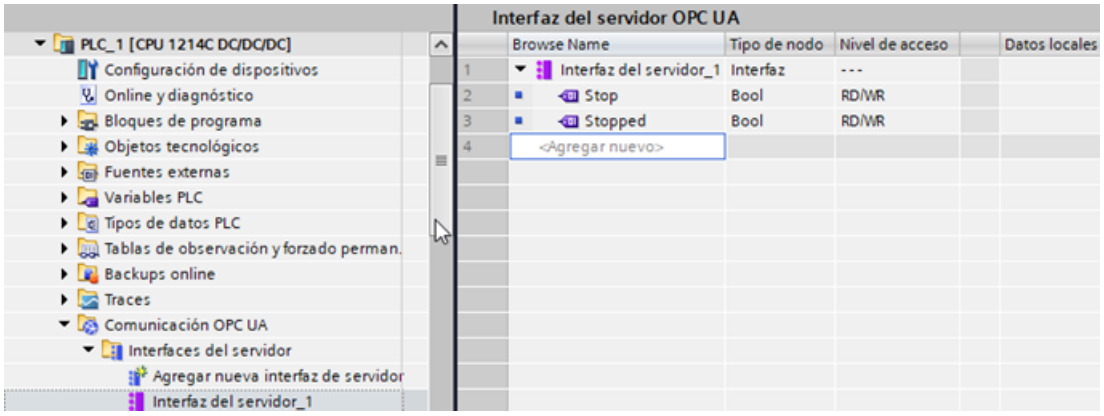
Cuando se descargan, definen la interfaz de servidor que es visible para un cliente OPC UA.

Se deben definir atributos TAG específicos para que TAG puede leerse/escribirse desde OPC UA.

En el cuadro de diálogo para crear un servidor es posible agregar o definir la interfaz de servicio. Para definir la interfaz de servidor OPC UA en TIA portal:

- Seleccione "Server interface", tipo: interfaz
- Seleccione "Especificación Companion", tipo: Especificación Companion
- Seleccione "Especificación Companion", tipo: espacio de nombre de referencia





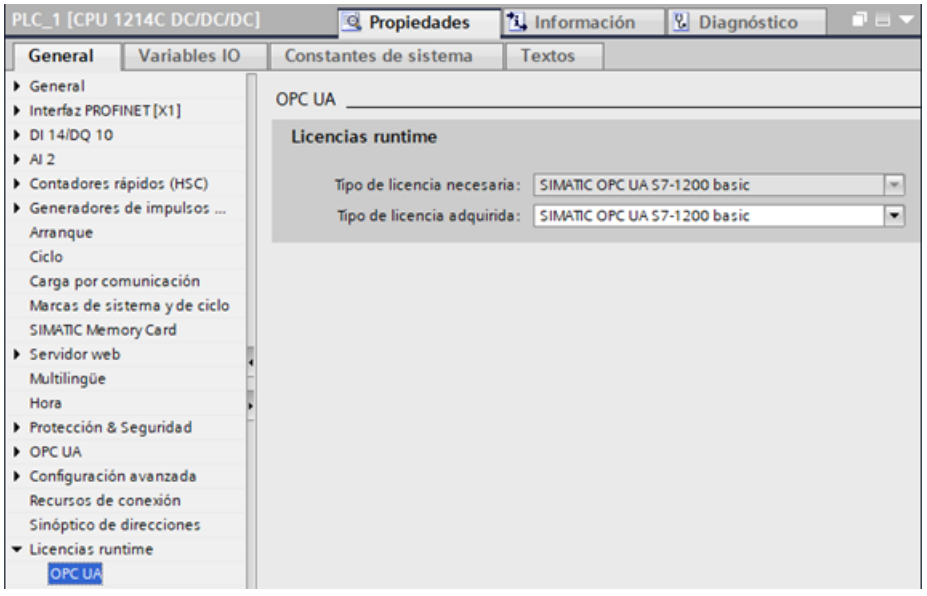
Licencias de runtime necesarias

Se requiere una licencia de runtime para ejecutar el servidor OPC UA para la CPU S7-1200. Están disponibles las licencias siguientes:

- SIMATIC OPC UA S7-1200 Basic DVD 6ES7823-0BA00-2BA0
- SIMATIC OPC UA S7-1200 Basic DL 6ES7823-0BE00-2BA0

El tipo de licencia necesario se indica en "Propiedades > General > Licencias runtime > OPC UA > Tipo de licencia necesario". Para confirmar la compra de la licencia necesaria, proceda del siguiente modo:

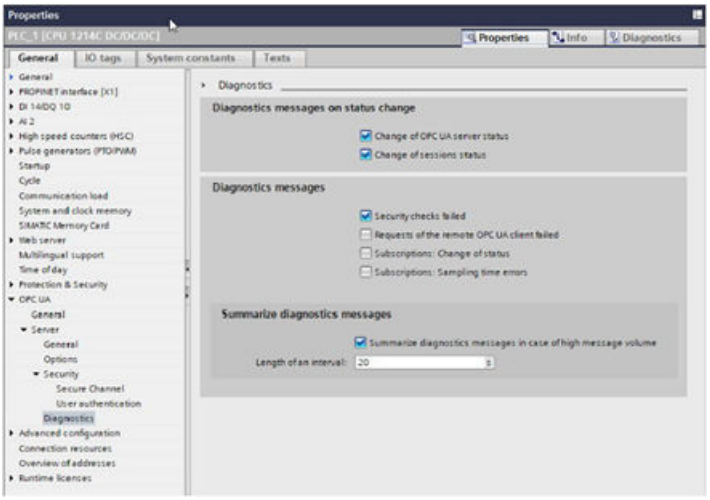
1. En las propiedades de la CPU, haga clic en "Licencias runtime > OPC UA".
2. Seleccione la licencia necesaria en la lista desplegable "Tipo de licencia adquirida".



11.10.4 Búfer de diagnóstico OPC UA

Con S7-1200 V4.5, el servidor OPC UA permite generar eventos para un componente que administre todos los avisos de diagnóstico de OPC UA y los incorpore al búfer de diagnóstico de la CPU S7-1200. Cada vez que un subcomponente de OPC UA emite un evento para este componente, el componente central comprueba si el aviso de diagnóstico se activa en la configuración y debe agregarse al búfer de diagnóstico.

El diagnóstico de OPC UA se configura en TIA Portal V17 y la configuración se descarga al PLC. A continuación, dispare los avisos de diagnóstico con un cliente OPC y compruebe las entradas del búfer de diagnóstico a través del cliente OPC.



Tal y como puede observarse, es posible seleccionar lo siguiente:

- Avisos de diagnóstico sobre cambios de estado
- Avisos de diagnóstico
- Agrupar avisos de diagnóstico

Avisos de diagnóstico sobre cambios de estado

No hay ningún aviso agrupado para tipos de aviso de diagnóstico controlados por las dos casillas de verificación "Cambio de estado del servidor OPC UA" y "Cambio del estado de sesión". No se espera una aparición muy frecuente de estos tipos de aviso. Cada uno de estos avisos tiene gran importancia para el uso de OPC UA. Significado de las casillas de verificación:

"Cambio de estado del servidor OPC UA" incluye los tipos de aviso siguientes:

- "El estado del servidor OPC UA ha cambiado a - Causa:"
- "El servidor OPC UA ha activado la directiva de seguridad más baja"

"Cambio de estado de la sesión" incluye el tipo de aviso

- "El estado del servidor OPC UA Server ha cambiado a - SessionID:"

Avisos de diagnóstico

Todos los tipos de aviso de esta categoría soportan el mecanismo de recopilación. Significado de las casillas de verificación:

- **"Error en las comprobaciones de seguridad"** incluye el tipo de aviso:
  "Error en las comprobaciones de seguridad del servidor OPC UA."
- **"Error en las solicitudes del cliente OPC UA remoto"** incluye:
  "Servidor OPC UA: uso indebido del servicio"
  "Servidor OPC UA: fallo del servicio con estado"
  "Servidor OPC UA: límite excedido del"
  "Servidor OPC UA: se ha producido un error en el estado del servidor"
  ➢ **"Suscripciones: Cambio de estado"** incluye "estado de la suscripción del servidor OPC UA modificado a ID de suscripción:"
  ➢ **"Suscripciones: errores de tiempo de muestreo"** incluye "no se pudo alcanzar la velocidad de muestreo del servidor OPC UA. Sobrecarga del ID de suscripción"

Agrupar avisos de diagnóstico

Esta parametrización solo es válida para los avisos de diagnóstico del servidor OPC UA. Es posible elegir diferentes configuraciones para los eventos de seguridad globales de la CPU.

Marque la casilla de verificación "Agrupar avisos de diagnóstico en caso de gran volumen de avisos" para activar el mecanismo de recopilación para todos los avisos de diagnóstico del servidor OPC UA. La duración del intervalo de recopilación puede especificarse en forma de cantidad de segundos, al contrario que en el cuadro de diálogo para eventos de seguridad globales de la CPU, donde también pueden especificarse en minutos y horas. Para avisos específicos solo se soportan los segundos como unidad de tiempo. El rango de valores es de 1 s a 7200 s.

Nota

Activar/desactivar el servidor OPC UA

La configuración del menú de diagnóstico OPC UA solo puede modificarse si el servidor OPC UA está activado.

Si el servidor OPC UA está desactivado, todos los ajustes se muestran atenuados y no pueden editarse.

11.10.4.1 Límites de OPC UA alcanzados

Ahora es posible saber si se ha alcanzado un límite del sistema. Sabrá que no pueden proporcionarse datos o cuándo el servidor o cliente no podrá usarse del modo acostumbrado.

Cuando se alcanzan límites de sistema específicos, se registra un aviso en el búfer de diagnóstico. Los límites pueden ser:

- El número máximo de subscripciones/elementos vigilados.
- El número máximo de elementos registrados.
- La velocidad de muestreo de subscripciones no puede alcanzarse.
- El número de clientes conectados.

Se crea una nueva entrada en el búfer de diagnóstico cuando se alcanza un límite determinado. El texto del aviso es:

Servidor OPC UA: se ha excedido el límite para <Nombre del límite>.

Existen textos predefinidos para los siguientes límites:

- Número de sesiones
- Número de elementos vigilados
- Número de subscripciones
- Número de nodos registrados
- Número de métodos de servidor
- Número de elementos vigilados por llamada
- Número de nodos por navegación
- Número de nodos por lectura
- Número de nodos por RegisterNodes
- Número de nodos por TranslateBrowsePathsToNodeIds
- Número de nodos por escritura
- Número de nodos por MethodCall
- Consumo de memoria

Nota

La solicitud de mantenimiento preventivo supera un límite configurado o un límite del sistema

Si una solicitud de mantenimiento preventivo supera un límite configurado o un límite del sistema, el servidor responderá con un error de servicio en la mayoría de los casos.

A continuación se muestran las aplicaciones siguientes:

Tabla 11-78 Aplicaciones para "Límites de OPC UA alcanzados"

Acción del usuario	Comportamiento previsto	Límite
El cliente crea más sesiones de las permitidas por HWCN	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el límite de <Límite>.	Sesiones
El cliente crea más elementos vigilados (MonitoredItems) de los permitidos por HWCN	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el límite de <Límite>.	Elementos vigilados
El cliente crea más subscripciones de las permitidas (depende del tipo de PLC)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	El cliente intenta conectarse con un testigo de identidad no soportado
El cliente registra más nodos de los permitidos por HWCN	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el límite de <Límite>.	Nodos registrados

Acción del usuario	Comportamiento previsto	Límite
El cliente lanza una respuesta BadOutOfMemory del servidor con un número elevado de sesiones, subscripciones o nodos registrados	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el límite de <Límite>.	Consumo de memoria
El cliente excede un límite de error práctico de un servicio (la cantidad de operaciones de una única solicitud supera el límite especificado por el servidor)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el límite de <Límite>.	Nodos por navegación Nodos por lectura Nodos por escritura Nodos por MethodCall Nodos por RegisterNodes Nodos por traducción

Aviso de diagnóstico para "Comportamiento por sobrecarga de subscripciones"

Con una subscripción OPC UA para diferentes elementos (variables), el servidor OPC UA de SIMATIC comprueba en intervalos predeterminados (intervalos de muestreo) si se produce algún cambio de valor en los elementos. Esta comprobación, el "muestreo", requiere un cierto tiempo, que es independiente del número y tipo de datos de los elementos. Una vez que el muestreo ha finalizado y hay una orden de publicación, el servidor envía los elementos al cliente. Si hay demasiados elementos en cola, puede producirse una "sobrecarga" de la pila de comunicación. La CPU no puede comprobar todos los elementos en el intervalo de muestreo especificado, por lo que debe pasar a la siguiente tarea. En ese caso, la CPU envía el código de estado "GoodOverload" (0x002F0000) por cada elemento, incluso aunque estos hayan sido comprobados. El significado del código de estado según IEC 61131-3 es: "El muestreo se ha ralentizado debido a limitaciones de recursos". El comportamiento se describe a continuación.



Si no es posible alcanzar una velocidad de muestreo específica, el cliente recibirá un valor con el estado "GoodOverload". Esto también genera una entrada "Sobrecarga de muestreo" en el búfer de diagnóstico:

"No se pudo alcanzar la velocidad de muestreo del servidor OPC UA. Sobrecarga del ID de suscripción"

El aviso solo se dispara si el estado del elemento vigilado cambia de "Good" a "GoodOverload".

11.10.4.2 Lectura remota del búfer de diagnóstico en OPC UA

Ahora existe la posibilidad de utilizar OPC UA para leer el búfer de diagnóstico de forma remota.

El servidor OPC UA proporciona un campo en el área de diagnóstico en el que solo está disponible la última entrada del búfer de diagnóstico. Es posible realizar una subscripción a ese campo y hacer un seguimiento de todos los avisos con este mecanismo.

Configure el diagnóstico de OPC UA en TIA Portal y descargue la configuración al PLC. A continuación, dispare los avisos de diagnóstico con un cliente OPC y compruebe las entradas del búfer de diagnóstico a través del cliente OPC.

11.10.4.3 Eventos de seguridad OPC UA

Ahora es posible conocer cuándo ocurre un evento de seguridad en el servidor OPC UA, lo que permite reaccionar cuando el servidor OPC UA es vulnerable o si se detecta algún ataque. Si se lanzan eventos de seguridad OPC UA específicos en el sistema de servidor/cliente OPC UA, se introducen avisos en el búfer de diagnóstico.

Un evento de seguridad puede ser, por ejemplo, que una sesión o algún intento de conexión se rechace debido a que los datos de autenticación son erróneos.

Los problemas de seguridad relacionados con el servidor OPC UA se notifican con este tipo de aviso:
"Servidor OPC UA: error en las comprobaciones de seguridad"

Los avisos se generan cuando se produce un error en una comprobación de seguridad y la operación no se ejecuta de la forma solicitada.

En la mayoría de los casos, el código de estado que devuelven otros componentes de seguridad, como OpenSSL, UMAC 711 o la pila OPC UA, se escribe en el aviso de diagnóstico.

Cada vez que el usuario recibe una respuesta negativa se genera un aviso. A continuación se muestran las aplicaciones más comunes:

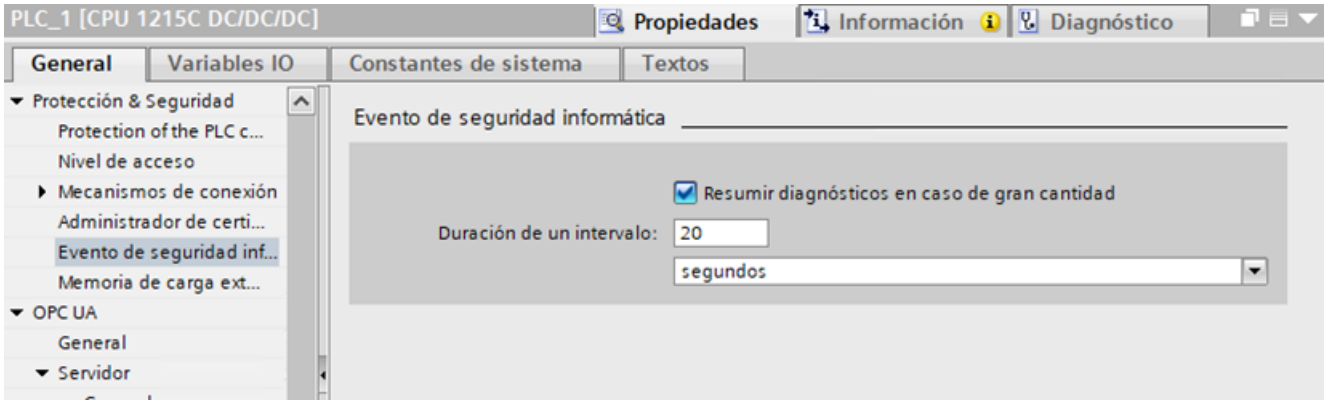
Acción del usuario	Comportamiento previsto	Código de estado
El cliente intenta conectarse con un modelo de seguridad no soportado	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	16#8054_0000 (BadSecurityModeRejected)
El cliente intenta conectarse con una directiva de seguridad no soportada	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	16#8055_0000 (BadSecurityPolicyRejected)
El cliente intenta conectarse con un testigo de identidad no soportado	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	El cliente intenta conectarse con un testigo de identidad no soportado
El cliente intenta conectarse con una autenticación incorrecta (nombre de usuario o contraseña incorrectos)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	16#8021_0000 (BadIdendityTokenRejected)
El cliente intenta conectarse con un certificado no válido (existen varias razones posibles que invalidan un certificado)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con el código de estado <Código de estado>.	Depende de la razón de la invalidez, p. ej. 16#8014_0000 (BadCertificateTimeInvalid) 16#801A_0000 (BadCertificateUntrusted)

Tipos de avisos de seguridad

Actualmente existen dos tipos de avisos de seguridad: avisos de seguridad para toda la CPU y avisos de seguridad de OPC UA.

Avisos de seguridad para toda la CPU

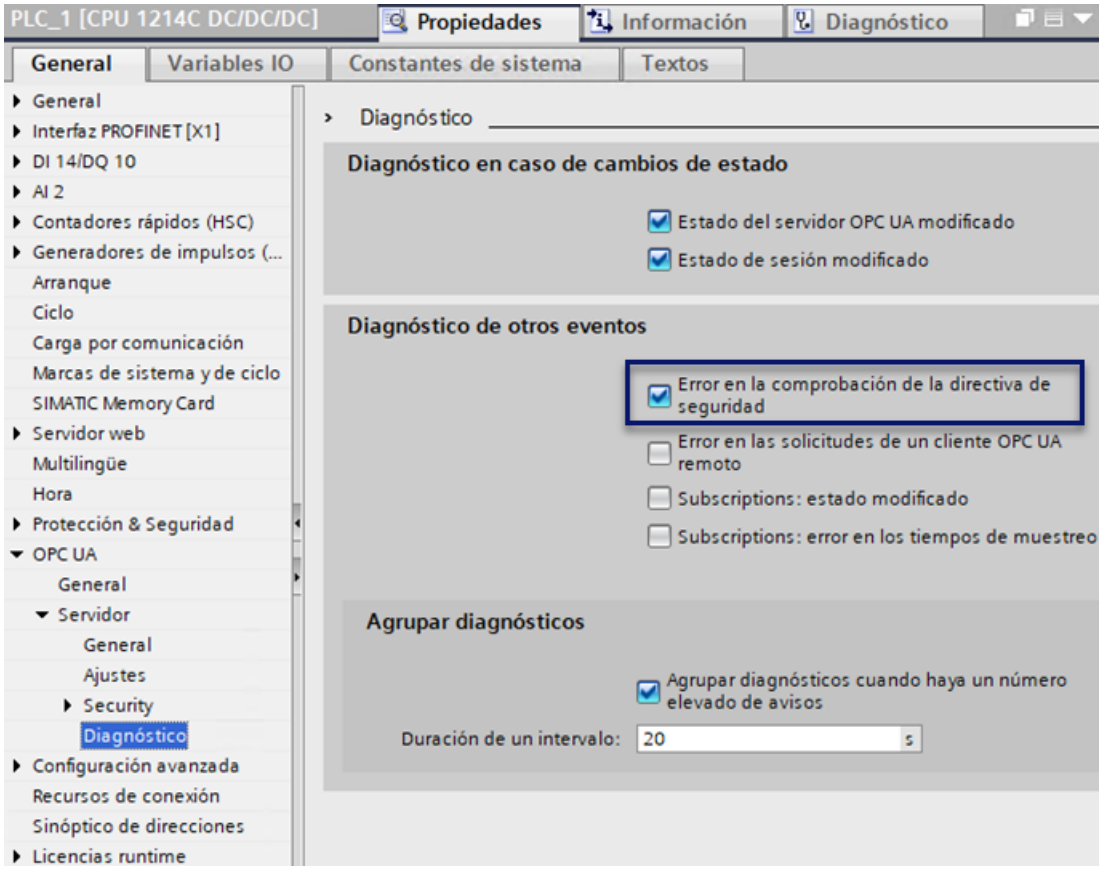
Los avisos de seguridad para toda la CPU son avisos de diagnóstico generados en escenarios definidos por el responsable de seguridad como críticos para toda la CPU. Este tipo de avisos utilizan un dominio de alarma especial. Esto permite a las HMI filtrarlos y tratarlos de forma diferenciada. En estos avisos de seguridad se trata de fallos de autenticación que pueden ocurrir cuando un usuario intenta iniciar sesión en el servidor web o el servidor OPC UA de la CPU. Hay un cuadro de diálogo de configuración en TIA Portal bajo el elemento "Protección y seguridad".



Seguridad de los avisos OPC UA

Los avisos de OPC UA relativos a la seguridad del servidor OPC UA son generados por el servidor OPC UA y utilizan el dominio de alarma o comprobaciones estándar durante la verificación de certificados.

La comprobación de seguridad relativa a OPC UA puede activarse y desactivarse en la vista de propiedades de TIA Portal (consulte a continuación).



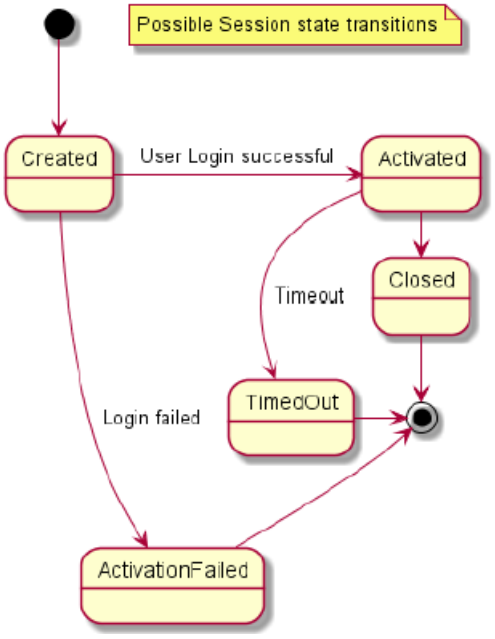
Información sobre la conexión de servidor OPC UA

Con el búfer de diagnóstico de OPC UA ahora es posible obtener información sobre los clientes conectados y ver el estado del servidor OPC UA. Se pueden realizar las siguientes acciones:

- Ver cuándo un cliente se conecta al servidor OPC UA
- Obtener un aviso cuando un cliente se desconecta del servidor OPC UA

Se registra un aviso de diagnóstico en el búfer de diagnóstico cuando cambia el estado de una sesión. Las posibles transiciones de estado de una sesión son las siguientes:

- Created
- Activated
- Closed
- TimedOut
- ActivationFailed



Se han implementado las aplicaciones siguientes:

Tabla 11-79 Aplicaciones implementadas para seguridad del servidor OPC UA activada

Acción del usuario	Comportamiento previsto	Directivas de seguridad
Se ha iniciado el servidor OPC UA (por ejemplo un ciclo de desconexión y reconexión)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con la directiva de seguridad más baja <Directiva de seguridad>.	Ninguna, Basic128Rsa15, Basic256, Basic256Sha256,

Tabla 11-80 Aplicaciones implementadas para cambios en el estado de sesión OPC UA

Condición previa	Acción del usuario	Comportamiento previsto	Estados
Servidor OPC UA funcionando, ningún cliente conectado	El cliente se conecta al servidor OPC UA y envía las credenciales correctas	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Created, Activated
Servidor OPC UA funcionando, ningún cliente conectado	El cliente se conecta al servidor OPC UA y envía credenciales incorrectas	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Created, Activation failed
Servidor OPC UA funcionando, cliente conectado	El cliente cierra la sesión correctamente	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Closed
Servidor OPC UA funcionando, cliente conectado	El cliente no envía más avisos al servidor hasta que la sesión finalice	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	TimedOut

Servidor OPC UA iniciado/detenido

En el búfer de diagnóstico de OPC UA ahora se puede ver cuándo se inició o paró un servidor, lo que permite conocer el estado global del servidor OPC UA. Cada vez que el servidor OPC UA se inicia o detiene se muestra una entrada en el búfer de diagnóstico. También es posible ver información adicional como, por ejemplo, si la interfaz genérica está encendida, o cuántas interfaces de servidor o espacios de nombres están activados.

Los posibles estados del servidor OPC UA son:

- Running
- Failed
- NoConfiguration
- Suspended
- Shutdown
- Test
- CommunicationFault
- Unknown
- Starting
- Restarting

Las razones para un cambio de estado son las siguientes: descarga/ciclo de desconexión-reconexión, llamada de instrucción desde el programa de usuario, solicitud remota. A continuación se muestran las posibles transiciones de estado del servidor.

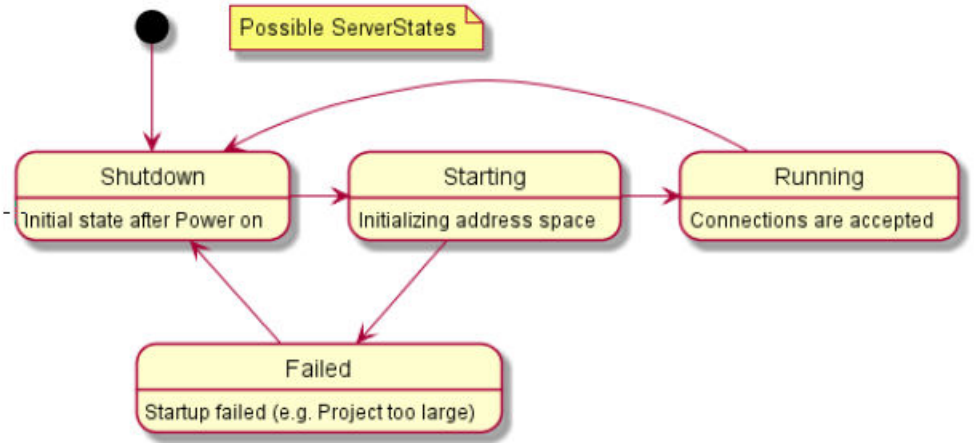


Tabla 11-81 Aplicaciones implementadas para cambios en el estado de sesión OPC UA

Condición previa	Acción del usuario	Comportamiento previsto	Estados
Servidor OPC UA en funcionamiento	Descarga de hardware con servidor OPC UA activado	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados"). Motivo: descarga / ciclo de desconexión-reconexión	Shutdown, Starting, Running
Servidor OPC UA detenido	Descarga de hardware con servidor OPC UA activado	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Created, Activated
Servidor OPC UA en funcionamiento	Descarga de hardware con el servidor OPC UA desactivado	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Created, Activation failed
Servidor OPC UA en funcionamiento	Descarga de hardware con el servidor OPC UA activado y typedictionary demasiado grande (demasiadas estructuras)	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	Closed
Servidor OPC UA detenido	Descarga de hardware con el servidor OPC UA activado y typedictionary demasiado grande	En el búfer de diagnóstico se agregan avisos de diagnóstico con cambios de estado (consulte la columna "Estados").	TimedOut

Nota

Descarga de software

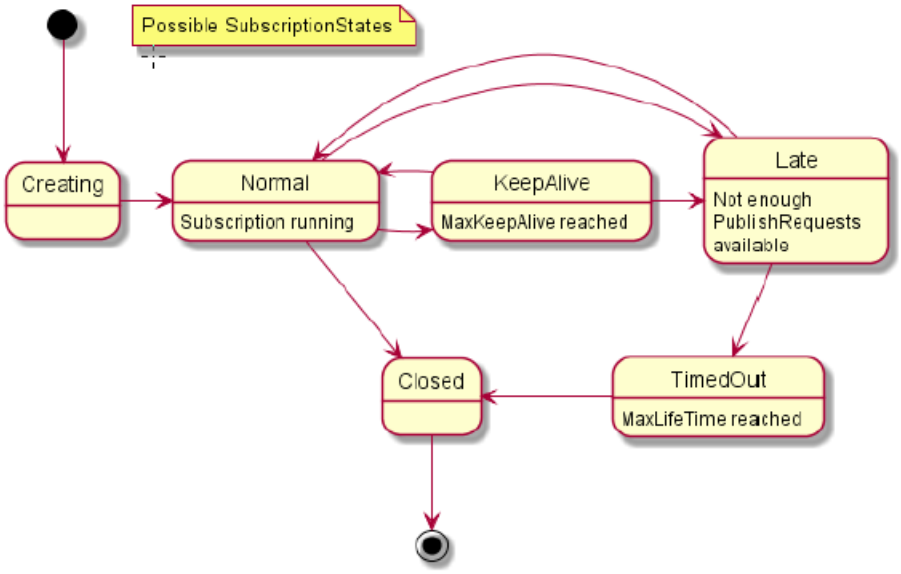
Una descarga de software también provocará un reinicio del servidor OPC UA.

Tiempo de sesión/suscripción OPC UA excedido

Ahora es posible saber cuándo ha concluido el tiempo de una sesión o subscripción OPC UA y conocer así qué sesiones o subscripciones continúan activas.

Cuando el estado de una sesión o suscripción OPC UA cambia, la información relativa a dichos eventos se escribe en el búfer de diagnóstico OPC UA. A continuación se indican los posibles estados de subscripción:

- Created
- Closed
- Normal
- Late
- KeepAlive
- TimedOut



Una subscripción activa suele cambiar entre los estados "Normal" y "KeepAlive" (cuando los valores vigilados cambian solo ocasionalmente). No se lanza ningún aviso para el estado "KeepAlive", ya que eso agregaría demasiados avisos al búfer de diagnóstico. Se han implementado las siguientes aplicaciones:

Tabla 11-82 Aplicaciones implementadas para seguridad del servidor OPC UA activada

Acción del usuario	Comportamiento previsto	Directivas de seguridad
Se ha iniciado el servidor OPC UA (por ejemplo un ciclo de desconexión y reconexión)	En el búfer de diagnóstico se agrega un aviso de diagnóstico con la directiva de seguridad más baja <Directiva de seguridad>.	Ninguna, Basic128Rsa15, Basic256, Basic256Sha256,

11.10.4.4 Uso incorrecto de OPC UA

Ahora es posible conocer los usos incorrectos del servidor OPC UA por parte del usuario o cliente.

Se detecta un "uso incorrecto" en aquellos casos en los que el cliente solicita datos o funcionalidades no destinados a ese uso. Si el cliente identifica que un servicio o solicitud no es válido antes de enviarlo, no debe enviarlo. De lo contrario se genera un aviso de diagnóstico.

La solicitud de un ID de nodo no válido puede identificarse examinando, por lo que la solicitud de un nodo no válido genera un aviso de tipo "Uso incorrecto". Cuando se leen atributos opcionales y se exceden subscripciones o límites de sesión del servidor, el cliente solo puede identificar el problema mediante "intentos", por lo que no se genera ningún aviso de uso incorrecto.

Nota: el espacio de nombres "http://opcfoundation.org/UA/" (ns=0) es un espacio de nombres especial gestionado por la OPC Foundation (o el SDK) y las funciones de diagnóstico son muy limitadas. No todos los "usos incorrectos" de este espacio de nombres generarán un aviso (por ejemplo, registro de un nodo desconocido).

El aviso de diagnóstico *"Servidor OPC UA: Uso incorrecto del servicio <Nombre del servicio> en el ID de sesión <ID de sesión>"* se registra en el búfer de diagnóstico cada vez que se detecta un

uso incorrecto de uno de los siguientes servicios. Solo se tienen en cuenta los servicios soportados por el servidor OPC UA para S7-1200.

- FindServers
- GetEndpoints
- FindServersOnNetwork
- CreateSession
- ActivateSession
- CloseSession
- Cancel
- Browse
- BrowseNext
- TranslateBrowsePathsToNodeIds
- RegisterNodes
- UnregisterNodes
- Write
- Read
- Call
- CreateMonitoredItems
- ModifyMonitoredItems
- DeleteMonitoredItems
- SetMonitoringMode
- SetTriggering
- CreateSubscription
- ModifySubscription
- DeleteSubscription
- Publish
- Republish
- SetPublishingMode
- OpenSecureChannel
- CloseSecureChannel

11.10.4.5 Sinopsis de avisos de OPC UA

Ejemplo de mensajes de diagnóstico del búfer de diagnóstico OPC UA para uso erróneo del servicio. Consulte el ejemplo siguiente.

Tabla 11-83 Sinopsis de mensajes para OPC UA de uso erróneo del servicio

Aviso	Evento único	Evento agrupado
Uso erróneo de OPC UA		
Definición	'@2W%t#7W@: Uso erróneo del servicio @4W%t#7W@ en ID de sesión @5X%u@\n	'@2W%t#7W@: Uso erróneo de un servicio de lectura
Ejemplo	Ejemplo para servidor OPC UA: Uso erróneo del servicio Read en ID de sesión 12345678	Servidor OPC UA: Uso erróneo de un servicio. – Aviso agrupado. - Mensaje agrupado para 3 ocurrencias durante los últimos 20 segundos

11.10.5 Llamada de métodos OPC UA

11.10.5.1 Información importante sobre métodos de servidor

Preparar el programa de usuario para métodos de servidor

En el servidor OPC UA de una CPU S7-1200 (a partir de firmware V4.5) tiene la posibilidad de proporcionar métodos a través del programa de usuario. Estos métodos pueden ser utilizados por clientes OPC UA, por ejemplo, para iniciar una tarea de fabricación mediante la llamada de método de la CPU S7-1200.

Los métodos OPC UA, una implementación de "Llamadas de procedimiento remoto", ponen a disposición un mecanismo eficaz para interacciones entre distintos nodos de comunicación. El mecanismo proporciona tanto confirmación de tareas como valores de respuesta de modo que ya no sea necesario programar mecanismos de handshake.

Los métodos OPC UA permiten transferir datos de forma coherente sin bits de disparo/ handshaking, por ejemplo, o activar acciones específicas en el controlador.

¿Cómo funciona un método OPC UA?

En principio, un método OPC UA funciona como un bloque de función con protección de know-how llamado por un cliente OPC UA externo en runtime.

El cliente OPC UA solo "ve" las entradas y salidas definidas. El contenido del bloque de función, el método o algoritmo, permanecen ocultos para el cliente OPC UA externo. El cliente OPC UA recibe respuesta sobre la correcta ejecución y valores devueltos por el bloque de función (método), o bien un mensaje de error si la ejecución no ha sido correcta.

Como programador, el usuario tiene pleno control y responsabilidad del contexto del programa en el que se ejecuta el método OPC UA.

Reglas para programar un método y el comportamiento en runtime

- Asegúrese de que los valores devueltos por el método OPC UA sean coherentes con los valores de entrada proporcionados por el cliente OPC UA.
- Respete las reglas de asignación de nombres y la estructura de parámetros, así como los tipos de datos permitidos (ver descripción de las instrucciones OPC UA).
- Comportamiento en runtime: el servidor OPC UA acepta una llamada por instancia La instancia del método no está disponible para otros clientes OPC UA hasta que la llamada es procesada por el programa de usuario o se haya cancelado tras el tiempo de espera.

El procedimiento básico para implementar un programa de usuario como método de servidor se expone a continuación.

Implementar un método de servidor

Un programa (bloque de función) para implementar un método de servidor se estructura del siguiente modo:

1. **Consultar la llamada del método de servidor con OPC_UA_ServerMethodPre**
   Primero llame la instrucción "OPC_UA_ServerMethodPre" en su programa de usuario (en su método de servidor).
   Esta instrucción tiene las siguientes tareas:
   - Con esta instrucción el usuario pregunta al servidor OPC UA de la CPU si su método de servidor ha sido llamado por un cliente OPC UA.
   - Si el método se ha llamado y el método de servidor tiene parámetros de entrada, su servidor ahora recibe los parámetros de entrada. Los parámetros de entrada del método de servidor vienen del cliente OPC UA.
2. **Editar el método de servidor**
   En esta sección del método de servidor se proporciona el programa de usuario.
   Tiene las misma opciones que en cualquier programa de usuario (por ejemplo, acceso a otros bloques de función o bloques de datos globales).
   Si el método de servidor utiliza parámetros de entrada, estos estarán a su disposición.
   Esta sección del método de servidor solo debe ejecutarse si un cliente OPC UA ha llamado el método de servidor.
   Tras ejecutar correctamente el método, ajuste los parámetros de salida del método de servidor (si el método tiene parámetros de salida).
3. **Responder al método de servidor con OPC_UA_ServerMethodPost**
   Para completar el método de servidor, llame la instrucción "OPC_UA_ServerMethodPost".
   Utilice los parámetros para notificar a la instrucción "OPC_UA_ServerMethodPost" si el programa de usuario se ha procesado o no.
   Si el programa de usuario se ha ejecutado correctamente, el servidor OPC UA es notificado a través de los parámetros pertinentes. A continuación, el servidor OPC UA envía los parámetros de salida del método de servidor al cliente OPC UA.

Llame siempre las instrucciones "OPC_UA_ServerMethodPre" y "OPC_UA_ServerMethodPost" como par independientemente de si el programa de usuario lo procesan ambas instrucciones o se continúa en el siguiente ciclo.

Encontrará un ejemplo de implementación del método de servidor en la Ayuda en pantalla de STEP 7.

Implementar el método de servidor

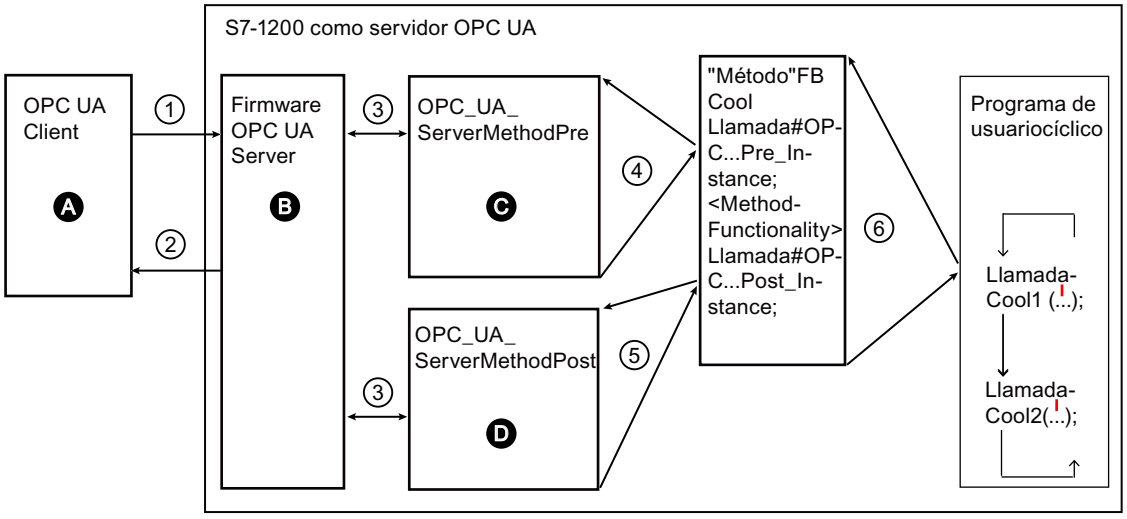
El siguiente esquema muestra cómo un cliente OPC UA (A) llama el método de servidor "Cool":

La CPU ejecuta la instancia "Cool1" del método de servidor "Cool" en el programa de usuario cíclico ⑥.

La CPU utiliza primero la instrucción "OPC_UA_ServerMethodPre" para consultar ④ si un cliente OPC UA

ha llamado el método de servidor "Cool" ①.

- Si el método de servidor no se ha llamado, la ejecución del programa retorna directamente al programa de usuario cíclico a través de ④ y ⑥. La CPU reanuda el programa de usuario cíclico tras "Cool1".
- Si el método de servidor se ha llamado, esta información se devuelve al método de servidor "Cool" a través de ④. La funcionalidad disponible se ejecuta ahora en el método de servidor "Cool", ver "<Method Functionality>" en el esquema.
  El método de servidor utiliza entonces la instrucción "OPC_UA_ServerMethodPost" ⑤ para notificar al firmware (B) que la instrucción se ha ejecutado ③. El firmware devuelve esta información a través de ② al cliente OPC UA invocante (A). La CPU reanuda el programa de usuario cíclico tras "Cool1".



A Llamada del método de servidor y gestión de la información "Done" (método completo).

① Llamada asíncrona del método de servidor.

② Información "Done" asíncrona del método invocado (método completo).

B Espera de llamadas de cliente OPC UA, gestión de llamadas en cola, reenvío de información "Done" del programa de usuario cíclico al cliente OPC UA.

③ Transferencia de datos del servidor OPC UA a las instancias de método del programa de usuario y viceversa.

C Comprobación de si el método se ha llamado. En caso afirmativo, reenvío de datos de entrada del servidor OPC UA a la instancia de método del programa de usuario y respuesta a la instancia de método de que el método se ha llamado ("called").

④ Llamada síncrona de la instrucción OPC_UA_ServerMethodPre como multiinstancia indicando la ubicación de los datos de entrada del servidor OPC UA. El valor de retorno indica si el método ha sido llamado o no por el cliente OPC UA. ⑤ Comprobación de si el método se ha completado o si sigue activo ("busy").

⑤ Comprobación de si el método se ha completado.

D En caso afirmativo, los datos de salida de la instancia de método se reenvían al servidor OPC UA y se notifica a la instancia de método que el método se ha completado. El servidor OPC UA recibe la notificación.

⑥ Llamada del FB de método (en este caso: FB Cool) con la instancia requerida y los parámetros del proceso.

Información sobre instrucciones de servidor

"OPC_UA_ServerMethodPre" y "OPC_UA_ServerMethodPost" se describen detalladamente en la Ayuda en pantalla de STEP 7 en Instrucciones > Comunicación > OPC UA > Servidor OPC UA.

La figura siguiente muestra un método de llamada de interfaz formateado correctamente.

OpenDoor

	Name	Datentyp	Offset	Defaultwert
1	Input			
2	<Hinzufügen>			
3	Output			
4	<Hinzufügen>			
5	InOut			
6	doorLocked	Bool	0.0	TRUE
7	Static			
8	UAMethod_OutParameters	"UDT_OpenDoorOutArguments"	2.0	
9	UAMethod_InParameters	"UDT_OpenDoorInArguments"	4.0	
10	Method_Result	DWord	6.0	16#0
11	Method_Finished	Bool	10.0	false
12	Started	Bool	10.1	false
13	Error_Message	WString	12.0	WSTRING#''
14	OPC_UA_ServerMethodPre_Instance	OPC_UA_ServerMethodPre		
15	OPC_UA_ServerMethodPost_Instance	OPC_UA_ServerMethodPost		
16	Temp			
17	Method_Called	Bool	0.0	
18	Pre_Done	Bool	0.1	
19	Pre_Error	Bool	0.2	
20	Post_Done	Bool	0.3	
21	Post_Error	Bool	0.4	

11.10.5.2 Condiciones específicas del uso de métodos de servidor

Tipos de datos permitidos

Si proporciona métodos de servidor, observe la regla siguiente:

- Asigne los tipos de datos como figura a continuación (tipo de datos SIMATIC - tipo de datos OPC UA). Otras asignaciones no están permitidas.
- STEP 7 no comprueba el cumplimiento de esta regla y no impide una asignación incorrecta. El usuario es responsable de que los tipos de datos seleccionados y asignados cumplan esta regla.

También puede utilizar los tipos de datos enumerados, por ejemplo, como elementos de estructuras/arrays/UDT para parámetros de entrada y salida de métodos de servidor creados por el usuario (UAMethod_InParameters y UAMethod_OutParameters).

Tipo de datos SIMATIC	**Tipo de datos OPC UA**
BOOL	Boolean
SINT	SByte
INT	INT16
DINT	INT32
USINT	Byte
UINT	UINT16
UDINT	UINT32
REAL	Float
LREAL	Double
WSTRING	String
DINT	Enumeración (codificación Int32) y todos los tipos de datos derivados

Servidor web

12

El servidor web para el S7-1200 ofrece acceso mediante página web a datos de la CPU y datos de proceso.

Puede acceder a las páginas web de S7-1200 desde un PC o un dispositivo móvil. Para dispositivos con pantallas pequeñas, el servidor web soporta una serie de páginas básicas (Página 858).

Se utiliza un navegador web para acceder a la dirección IP de la CPU S7-1200 o a la dirección IP de un módulo CP (procesador de comunicaciones) apto para servidor web (Página 853) en el rack local con la CPU para establecer la conexión. El S7-1200 soporta varias conexión concurrentes.



Nota

Varias conexiones simultáneas del servidor web

El servidor web del S7-1200 permite 30 conexiones simultáneas (presuponiendo que se dispone de suficientes conexiones dinámicas). Las instancias abiertas del navegador pueden ocupar entre 2 y 8 conexiones cada una. El servidor web permite un máximo de 7 usuarios registrados, pero Siemens recomienda mantener el número de usuarios simultáneos lo más reducido posible. Una media de 7 usuarios a la vez suele ser aceptable con cargas de trabajo normales.

Páginas web estándar

El S7-1200 incluye páginas web estándar (Página 857) a las que se accede desde el navegador web del PC (Página 851) o desde un dispositivo móvil (Página 852):

- Introducción (Página 862) - punto de acceso a las páginas web estándar
- Página de inicio (Página 863) - información general sobre la CPU
- Diagnóstico (Página 864) - información detallada sobre la CPU incluyendo la referencia, el número de serie y el número de versión, la protección del programa y el uso de memoria
- Búfer de diagnóstico (Página 866) - el búfer de diagnóstico
- Información del módulo (Página 867) - información sobre los módulos locales y remotos y la posibilidad de actualizar el firmware de los módulos
- Comunicación (Página 872) - información sobre las direcciones de red, propiedades físicas de las interfaces de comunicación, estadísticas, parámetros así como un resumen de las conexiones e información de diagnóstico
- Estado de variables (Página 875) - E/S y variables de la CPU, accesibles con la dirección o el nombre de variable PLC

- Tablas de observación (Página 877) - tablas de observación que se han configurado en STEP 7
- Copia de seguridad online (Página 879) - posibilidad de realizar una copia de seguridad de una CPU online o de restaurar una copia de seguridad online realizada previamente
- Registros de datos (Página 882)- permite ver una lista de todos los registros de datos del PLC, descargar un registro de datos del PLC al equipo, recuperar (consultar) un registro de datos del PLC y buscar y borrar un registro de datos del PLC
- Archivos de usuario (Página 884)- permite ver una lista de los archivos de usuario en el PLC, descargar un archivo de usuario del PLC al equipo, cargar un archivo de usuario del equipo al PLC y borrar un archivo de usuario del PLC
- Páginas de usuario (Página 891): crear páginas web personalizadas para acceder a los datos de la CPU
- Navegador de archivos (Página 889) - explorador de archivos almacenados internamente en la CPU o en una SIMATIC Memory Card, como por ejemplo registros y recetas.
- Inicio de sesión (Página 859) - inicia sesión como un usuario distinto o cierra sesión.

Estas páginas está incluidas en la CPU S7-1200 y están disponibles en inglés, alemán, francés, español, italiano y chino simplificado. Se necesita configuración adicional en TIA Portal para ver avisos de diagnóstico PLC (capítulo 15). Todas las páginas, excepto las de introducción e inicio, requieren para su visualización derechos de usuario (Página 848) adicionales que se configuran en STEP 7.

Páginas web definidas por el usuario

El S7-1200 también ofrece ayuda para crear páginas web definidas por el usuario que pueden acceder a los datos de la CPU. Es posible desarrollar estas páginas con el software de diseño HTML de su elección, incluyendo comandos "AWP" (Automation Web Programming) predefinidos en su código HTML para acceder a los datos de la CPU. Consulte el apartado Páginas web definidas por el usuario (Página 891) para obtener información detallada sobre el desarrollo de páginas web definidas por el usuario y sobre la configuración y programación correspondiente en STEP 7.

Se accede a las páginas web definidas por el usuario desde un PC o dispositivo móvil partiendo de las páginas web estándar o básicas. También es posible configurar una de las páginas web definidas por el usuario para que sea la página de entrada (Página 908) al servidor web.

Web API

La CPU S7-1200 también proporciona una Web API (Página 931), que es una interfaz que permite leer y escribir datos de proceso.

Requisitos de navegadores web

Siemens ha probado las páginas estándar del servidor web y verifica el soporte de los siguientes navegadores web:

- Internet Explorer 11
- Microsoft Edge V44
- Microsoft Edge Chromium Based V86
- Mozilla Firefox V64
- Opera V58
- Google Chrome V75
- Navegador Android para Android Pie V9
- Mobile Chrome para Android Pie V9
- Mobile Safari y Chrome para iOS V13

Cuando se utiliza el control de navegador HTML en un proyecto WinCC, el servidor web soporta los siguientes paneles HMI de Siemens para las páginas estándar:

- Basic Panels
  - Gen 2 KTP400 a KTP1200
- Comfort Panels
  - TP700 a TP2200
  - KP400 a KP1500
  - KTP400
  - TP700 Comfort Outdoor
- Mobile Panels
  - Gen 2 KTP700[F], KTP900[F]
- Unified Comfort Panels
  - Basic Panel (2.ª gen.)
  - Mobile Panel (2.ª gen.)

En la sección Limitaciones (Página 932) aparecen las restricciones que rigen para navegadores web que pueden interferir con la visualización de páginas web definidas por el usuario.

Rendimiento del servidor web

Muchos factores pueden afectar el rendimiento del servidor web. La CPU S7-1200 y la programadora deben compartir tiempo con otras tareas que consumen recursos y tiempo de procesamiento. Si el rendimiento del servidor web es deficiente, pruebe a realizar estos ajustes para mejorarlo:

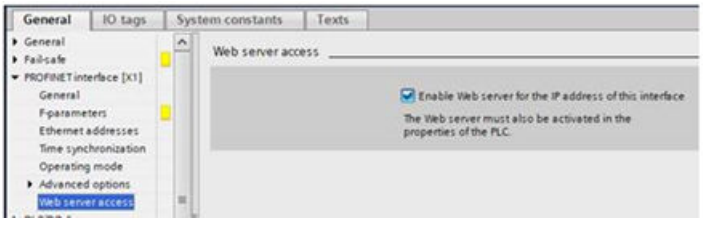
- Aumente la carga por comunicación (Página 91) en el PLC del 20% al 50%.
- Configure un tiempo mínimo de ciclo (Página 91). Ajustando un tiempo mínimo de ciclo se incrementa el tiempo de comunicación entre la CPU S7-1200 y la programadora.
- Utilice la interfaz Ethernet de la CPU S7-1200 en lugar de un módulo CP (Página 853) para acceder al servidor web.

12.1 Habilitar el servidor web

El servidor web se habilita en STEP 7 desde la configuración de dispositivos de la CPU con la que se desea establecer la conexión.

Para habilitar el servidor web, proceda del siguiente modo:

1. Seleccione la CPU en la vista de la configuración de dispositivos.
2. En la ventana de inspección, seleccione "Servidor web" desde las propiedades de la CPU.
3. Seleccione la casilla de verificación para "Activar servidor web en todos los módulos de este dispositivo".
4. Para una mayor seguridad, asegúrese de que "Permitir acceso solo vía HTTPS" está seleccionado si requiere un acceso seguro al servidor web.
5. Si se selecciona "Activar actualización automática" para "Actualización automática", las páginas web estándar se actualizan cada diez segundos. También puede introducir un intervalo de actualización personalizado en segundos en el campo "Intervalo de actualización".
   - PLC PROFINET: Asegúrese de que la propiedad "Activar el servidor web para la dirección IP de esta interfaz" está activada en el PLC para que el servidor web sea accesible a través de la interfaz PLC PROFINET.
   - Puerto WanCP PROFINET: Si utiliza un dispositivo WanCP, asegúrese de que la propiedad "Activar el servidor web para la dirección IP de esta interfaz" está marcada para que el servidor web sea accesible a través del puerto WanCP PROFINET en particular.



⚠ ADVERTENCIA

Acceso no autorizado a la CPU mediante el servidor web

Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU (Página 163) protegidos por contraseña e ID de usuario de servidor web (Página 848) con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.
- Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida.

Después de descargar la configuración del dispositivo es posible utilizar las páginas web estándar para acceder a las páginas de introducción e inicio de la CPU. Para acceder a páginas adicionales hay que configurar uno o más usuarios de servidor web (Página 848).

Si ha creado y habilitado páginas web definidas por el usuario (Página 891), podrá acceder a estas desde el menú de navegación de las páginas web estándar o básicas.

Nota

Cambio de dispositivo: Sustituir una CPU V3.0 por una CPU V4.x

Si se reemplaza una CPU V3.0 existente por una CPU V4.x (Página 1448) y se convierte el proyecto V3.0 a un proyecto V4.x, debe tenerse en cuenta que STEP 7 y la CPU V4.x mantienen los ajustes del servidor web para

- "Activar servidor web en todos los módulos de este dispositivo"
- "Permitir acceso solo con HTTPS"

Nota

Si se está ejecutando una "Carga en RUN" (Página 1224), las páginas web estándar y personalizadas no actualizan valores de datos ni permiten escribir valores de datos hasta que no haya finalizado la carga. El servidor web rechaza los intentos de escribir valores de datos mientras hay una carga en curso.

12.2 Configuración de usuarios de servidor web

Puede configurar usuarios con varios niveles de privilegios para acceder a la CPU mediante el servidor web.

Para configurar usuarios de servidor web y sus privilegios asociados, proceda del siguiente modo:

1. Seleccione la CPU en la vista de la configuración de dispositivos.
2. En la ventana de inspección, seleccione "Servidor web" desde las propiedades de la CPU y habilite el servidor web (Página 846).
3. Seleccione "Administración de usuarios" en las propiedades del servidor web.
4. Introduzca nombres de usuario, niveles de acceso y contraseñas para los datos de inicio de sesión que desee proporcionar.

Una vez que haya cargado la configuración en la CPU, solo los usuarios autorizados podrán acceder a las funciones del servidor web para las que tienen privilegios.

Niveles de acceso al servidor web

STEP 7 proporciona un usuario predeterminado llamado "Cualquiera" sin contraseña. Por defecto, este usuario no dispone de privilegios adicionales y solo puede visualizar las páginas web estándar de Inicio (Página 863) e Introducción (Página 862). Sin embargo, puede configurar derechos adicionales tanto para el usuario "Todos" como para otros usuarios:

- Consultar diagnóstico
- Leer variables
- Escribir variables
- Leer estados de variables

- Escribir estados de variables
- Abrir páginas web definidas por el usuario
- Escribir en páginas web definidas por el usuario
- Leer archivos
- Escribir o borrar archivos
- Cambiar el estado operativo
- Hacer parpadear LED
- Actualizar el firmware
- Realizar una copia de seguridad de la CPU
- Restaurar una CPU
- Cambiar parámetros del sistema
- Cambiar parámetros de la aplicación

Si se ha especificado que una de las páginas web definidas por el usuario sea la página de entrada (Página 908) al servidor web, el usuario "Todos" debe tener el derecho "Abrir páginas web definidas por el usuario".

⚠ ADVERTENCIA

Acceso no autorizado a la CPU mediante el servidor web

Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU (Página 163) protegidos por contraseña e ID de usuario de servidor web con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.
- Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida.

Nota

Actualice el cifrado de contraseña para el sustitución de dispositivos por V4.5 (o superior)

Tras sustituir un dispositivo por uno de la versión V4.5 (o superior) debe actualizarse el cifrado de contraseña de usuario del servidor web. Desde la configuración de dispositivos de la CPU en TIA Portal, haga clic en el botón "Actualizar cifrado de contraseña" en la administración de usuarios del servidor web.

12.3 Acceso a las páginas web desde un PC

Es posible acceder a la página web estándar del S7-1200 desde un PC o desde un dispositivo móvil por medio de la dirección IP de la CPU S7-1200 o la dirección IP de cualquier CP apto para servidor web (Página 853) en el rack local.

Para acceder a las páginas web estándar del S7-1200, proceda del siguiente modo:

1. Asegúrese de que el S7-1200 y el PC se encuentren dentro de una red Ethernet común o que estén interconectados directamente con un cable Ethernet estándar.
2. Abra un navegador web e introduzca la URL "http://ww.xx.yy.zz", siendo "ww.xx.yy.zz" la dirección IP de la CPU S7-1200 o la dirección IP de un CP en el rack local.

El navegador web abre la página web estándar de introducción (Página 862) o la página HTML predeterminada de las páginas web definidas por el usuario si se ha configurado una para que sea la página de entrada (Página 908).

Nota

Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida. Asimismo hay que tener en cuenta todas las restricciones (Página 932) impuestas por el entorno web o el sistema operativo.

Acceso a páginas web estándar introduciendo la URL de la página

Se puede acceder a una página web estándar específica desde la URL de la página. Para ello, introduzca la URL en el formato "https://ww.xx.yy.zz/<página>.html", donde "ww.xx.yy.zz" corresponde a la dirección IP de la CPU S7-1200 o a la dirección IP de un CP del rack local:

- https://ww.xx.yy.zz/start.html - Página de inicio (Página 863) con información general sobre la CPU
- https://ww.xx.yy.zz/identification.html - Información de identificación (Página 864) sobre la CPU incluyendo la referencia, el número de serie y el número de versión; ahora se denomina Página de diagnóstico
- https://ww.xx.yy.zz/module.html - Información sobre los módulos del rack local y la capacidad de actualizar el firmware (Página 867)
- https://ww.xx.yy.zz/communication.html - Información de comunicación (Página 872) sobre las direcciones de red, las propiedades físicas de las interfaces de comunicación y las estadísticas de comunicación
- https://ww.xx.yy.zz/diagnostic.html - El búfer de diagnóstico (Página 866)
- https://ww.xx.yy.zz/variable.html - Variables de CPU y E/S (Página 875), accesibles por dirección, nombre de variable PLC o nombre de variable de bloque de datos
- https://ww.xx.yy.zz/watch.html - Tablas de observación (Página 877)
- https://ww.xx.yy.zz/filebrowser.html - Explorador que permite acceder a archivos de registro de datos o archivos de receta (Página 889) almacenados internamente en la CPU o en una Memory Card

- https://ww.xx.yy.zz/index.html - Página de introducción (Página 862) para acceder a las páginas web estándar
- https://ww.xx.yy.zz/login.html - Página para iniciar sesión (Página 859) si ningún usuario ha iniciado sesión; de lo contrario, la página está vacía.

Por ejemplo, si se introduce "https//ww.xx.yy.zz/communication.html", el navegador mostrará la página de comunicación.

Nota

Tenga en cuenta que todas las páginas web estándar que no se listan específicamente arriba (p. ej. la página de copia de seguridad online (Página 879)) no tienen una URL de acceso directo.

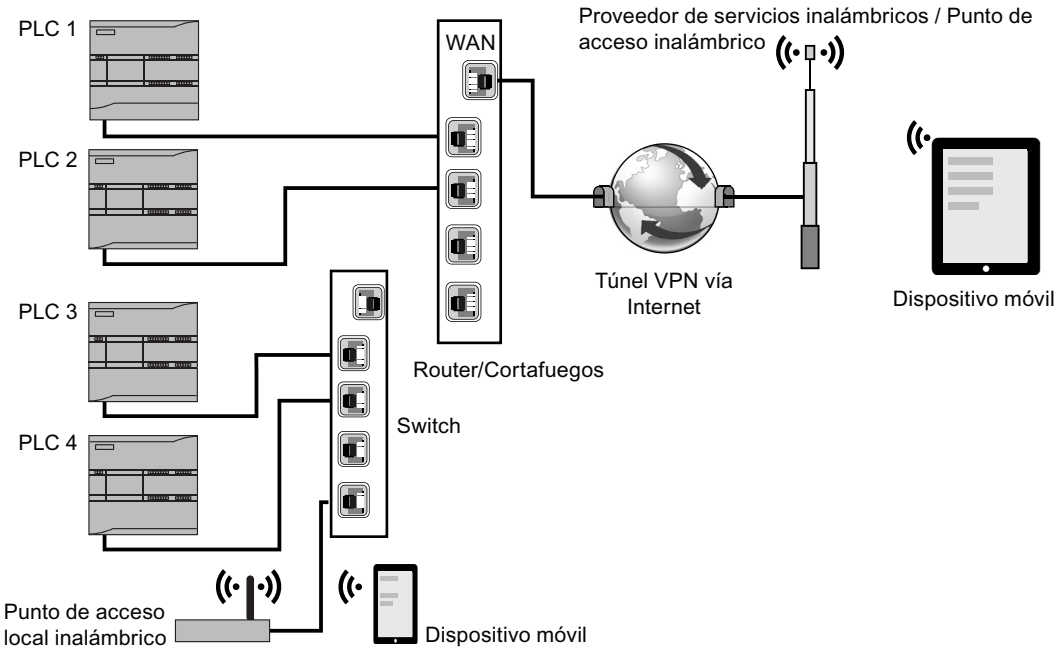
Acceso seguro

Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida. Solicite y use https:// en vez de http:// para acceder de modo seguro (Página 846) a las páginas web estándar. Al conectar al S7-1200 con https://, el sitio web cifra la sesión con un certificado digital. El servidor web transmite los datos de forma segura y nadie puede acceder a ellos para verlos. Por lo general, aparece un aviso de seguridad indicando que confirmando con "Sí" se puede acceder a las páginas web estándar. Para evitar que la advertencia de seguridad aparezca en cada acceso seguro, es posible importar el certificado de software de Siemens a su navegador web (Página 854).

12.4 Acceso a las páginas web desde un dispositivo móvil

Para acceder a un S7-1200 desde un dispositivo móvil, conecte el PLC a una red que conecte a Internet o a un punto de acceso inalámbrico local. Use una red privada virtual (VPN = Virtual Private Network) para conectar un dispositivo móvil al servidor web del PLC S7-1200. Puede usar redireccionamiento del puerto del router inalámbrico para mapear la dirección IP del PLC a una dirección que permita a un dispositivo móvil acceder a él desde Internet. Para configurar el redireccionamiento de puerto, siga las instrucciones para la configuración del software de su router. Puede conectar tantos PLC y dispositivos de conmutación como admita su router.

Sin redireccionamiento de puerto, puede conectarse a un PLC, pero solo localmente dentro del alcance de la señal inalámbrica.



En este ejemplo, un dispositivo móvil que está dentro del alcance del punto de acceso inalámbrico puede conectarse al PLC 3 y PLC 4 a partir de sus direcciones IP. Desde Internet, fuera del alcance inalámbrico local, un dispositivo móvil puede conectarse a PLC 1 y PLC 2 con la dirección del puerto redireccionada para cada PLC.

Para acceder a las páginas web estándar debe tener acceso a un servicio de telefonía móvil o a un punto de acceso inalámbrico. Para acceder a un PLC desde Internet, introduzca la dirección redireccionada del puerto en el navegador web de su dispositivo móvil para acceder al PLC, por ejemplo http://ww.xx.yy.zz:pppp o https:/ww.xx.yy.zz:pppp, en la que ww.xx.yy.zz es la dirección del router y pppp es la asignación de puerto de un PLC específico.

Para un acceso local mediante un punto de acceso local inalámbrico, introduzca la dirección IP de la CPU S7-1200 o un CP apto para servidor web (Página 853) en el rack local:

- http://ww.xx.yy.zz o https://ww.xx.yy.zz para acceder a las páginas web estándar (Página 857)
- http://ww.xx.yy.zz/basic o https://ww.xx.yy.zz/basic para acceder a las páginas web básicas (Página 858)

Para una mayor seguridad, configure el servidor web de modo que solo sea accesible mediante acceso seguro (HTTPS) (Página 846).

12.5 Usar un módulo CP para acceder a páginas web

Independientemente de si se accede al servidor web desde un PC o un dispositivo móvil, es posible conectarse a la página web estándar a través de uno de los módulos CP siguientes si se ha configurado en STEP 7 y se ha instalado en el rack local o la CPU S7-1200:

- CP 1242-7 GPRS V2
- CP 1243-1

- CP 1243-7 LTE-EU
- CP 1243-7 LTE-US
- CP 1243-8 IRC

Use la página web estándar de inicio (Página 863) para acceder a las páginas web a través de estos módulos CP. La página de inicio muestra todos los módulos CP configurados e instalados que hay en el rack local, pero solo es posible acceder a páginas web a partir de los que aparecen listados en la parte superior.

Nota

Acceso a páginas web estándar cuando los CPs aptos para servidores web están en el rack local

Es posible que observe retardos de hasta uno y dos minutos al conectarse a las páginas web estándar de S7-1200 cuando los CPs aptos para servidor web están en el rack local. Si las páginas no están disponibles o recibe errores solo debe esperar uno o dos minutos y actualizar la página.

12.6 Descargar e instalar un certificado de seguridad

El certificado de seguridad predeterminado de Siemens se puede descargar en las opciones de Internet.

Con el certificado se puede omitir la verificación de seguridad introduciendo https://ww.xx.yy.zz en el navegador web, siendo "ww.xx.yy.zz" la dirección IP del dispositivo. Si utiliza una URL http:// en vez de una URL https://, entonces no es necesario cargar e instalar el certificado.

A partir de STEP 7 V15 SP1 con compatibilidad con CPU S7-1200 V4.3 es posible crear certificados en la configuración de dispositivos de una CPU S7-1200. Esta función está disponible en la configuración general del dispositivo, bajo "Protección y seguridad > Administrador de certificados". Encontrará instrucciones sobre el administrador de certificados y cómo crear certificados globales y locales específicos de la CPU en el sistema de información de STEP 7.

A partir de STEP 7 V17 con compatibilidad con CPU S7-1200 V4.5 también es posible crear certificados para el servidor web de una CPU S7-1200. Puede configurar la seguridad del servidor web en "Servidor web > Seguridad" en la configuración de dispositivos de la CPU.

Nota

Límite de certificados en S7-1200

S7-1200 tiene un límite de certificados del sistema de 64.

Todos los certificados se incluyen en este número (por ejemplo, certificados web, certificados OPC UA y certificados OUC).

Si el certificado de servidor web ha sido firmado por una autoridad certificadora (CA) y está presente en TIA Portal, entonces el servidor web utilizará 2 certificados (uno para el certificado de servidor web y otro para el certificado de CA descargado).

Si hay más de 64 certificados, TIA Portal muestra un error que indica que se ha excedido el número máximo de 64 certificados. Habrá que eliminar algunos certificados de la configuración del PLC.

Descargar el certificado

Utilice el enlace "Descargar certificado" de la página de introducción (Página 862) para descargar el certificado de seguridad Siemens en su PC. El procedimiento para descargar e importar varía en función del navegador web utilizado.

Importar el certificado en Internet Explorer

1. Haga clic en el enlace "Descargar certificado" de la página de introducción.
2. En el siguiente cuadro de diálogo, haga clic en "Abrir" para abrir el archivo.
3. En el cuadro de diálogo "Certificado", haga clic en el botón "Instalar certificado" para iniciar el asistente de importación de certificados.
4. Haga clic en "Siguiente" en el cuadro de diálogo "Asistente para importar certificados" para ajustar la ruta de almacenamiento de los certificados.
5. Elija "Colocar todos los certificados en el siguiente almacén" y haga clic en el botón "Examinar".
6. Desde el cuadro de diálogo "Seleccionar almacén de certificados", elija "Entidades de certificación raíz de terceros" y haga clic en Aceptar.
7. Haga clic en "Siguiente" y después en "Finalizar" para completar el Asistente para importar certificados.

Importar el certificado en Mozilla Firefox

1. Haga clic en el link "download certificate" en la página de introducción.
2. Cuando aparezca la pregunta correspondiente, haga clic en Aceptar para confiar en la S7-1200 Controller Family.

En versiones anteriores de Mozilla Firefox, tras hacer clic en "download certificate" hay que guardar el archivo y ejecutar el asistente:

1. Haga clic en "Save file" en el cuadro de diálogo desde el que se ha abierto el certificado. Entonces aparece el cuadro de diálogo "Downloads".
2. En el cuadro de diálogo "Downloads", haga doble clic en "MiniWebCA_Cer.crt" o en el nombre del certificado que ha creado. Si ha intentado realizar la carga más de una vez, aparecerán varias copias. Simplemente haga doble clic en una de las entradas del certificado repetido.
3. Haga clic en "OK" si se solicita abrir un archivo ejecutable.
4. Haga clic en "Open" en el cuadro de diálogo "Open File - Security Warning", en caso de que aparezca. Entonces aparece el cuadro de diálogo "Certificate".
5. En el cuadro de diálogo "Certificate", haga clic en el botón "Install Certificate".
6. Siga las instrucciones de los cuadros de diálogo del asistente de importación de certificados dejando que el sistema operativo seleccione automáticamente la ubicación del certificado.
7. Si aparece el cuadro de diálogo "Security Warning", haga clic en "Yes" para confirmar la instalación del certificado.

Otros navegadores

Siga las convenciones de su navegador web para importar e instalar el certificado Siemens.

Una vez que se haya instalado el certificado de seguridad "S7-1200 Controller Family" de Siemens en las opciones de Internet para el contenido del navegador web, ya no será necesario confirmar un aviso de seguridad al acceder al servidor web con https:// ww.xx.yy.zz.

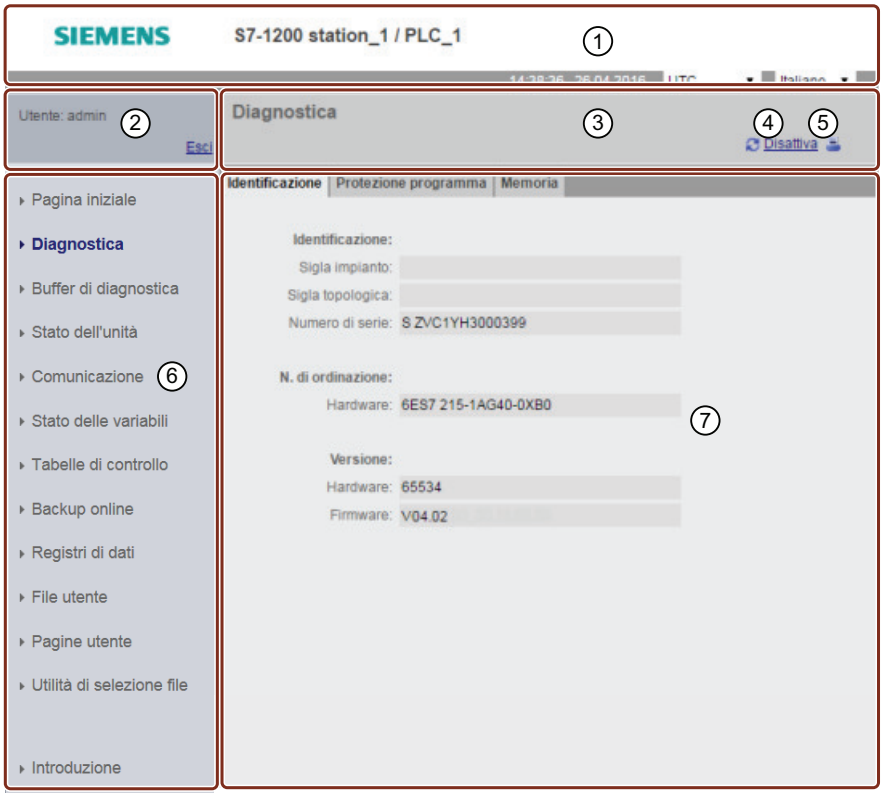
Nota

El certificado de seguridad permanece constante aunque se rearranque la CPU. Si se cambia la dirección IP del dispositivo, hay que descargar un nuevo certificado para utilizar un navegador que no sea Internet Explorer o Mozilla Firefox.

12.7 Páginas web estándar

12.7.1 Representación de las páginas web estándar

Todas las páginas web estándar de S7-1200 tienen una estructura común con enlaces de navegación y controles de página. Con independencia de si está viendo la página en un PC o en un dispositivo móvil, cada página tendrá la misma área de contenido, pero los controles de la estructura y la navegación variarán en función del tamaño y la resolución de la pantalla del dispositivo. En un PC estándar o un dispositivo móvil de gran tamaño, la estructura de una página web estándar se muestra del siguiente modo:



① Un encabezado del servidor web dotado de selector para visualizar la hora local del PLC o la hora UTC y un selector para el idioma del display (Página 153)

② Iniciar o cerrar sesión

③ Encabezado estándar de la página web con el nombre de la página visualizada. Este ejemplo muestra la página Diagnóstico > Identificación de la CPU. Algunas de las páginas web estándar, como la de información de módulo, también muestran aquí una ruta de navegación si se puede acceder a varias pantallas de ese tipo.

④ Icono de actualización: para páginas con actualización automática, activa o desactiva la función de actualización automática; en páginas sin actualización automática actualiza la página con datos actuales

⑤ Icono de impresión: prepara y visualiza una versión de impresión de la información disponible en la página visualizada

⑥ Área de navegación para pasar a otra página
⑦ Área de contenido de la página web estándar específica visualizada. Este ejemplo muestra la página de diagnóstico.

Nota

Páginas web estándar del módulo CP

Determinados módulos CP (Página 853) disponen de páginas web estándar cuya apariencia y funcionalidad son similares a las de las páginas web estándar de la CPU S7-1200. Consulte la documentación del CP para obtener las descripciones de las páginas web estándar del CP.

12.7.2 Páginas básicas

El servidor web ofrece páginas básicas destinadas a ser utilizadas en dispositivos móviles. Se accede a las páginas básicas utilizando la dirección IP del dispositivo y agregando "basic" a la URL: http://ww.xx.yy.zz/basic o https://ww.xx.yy.zz/basic

Las páginas básicas tienen un aspecto similar a las páginas estándar, aunque con algunas diferencias. Las páginas omiten el área de navegación, de inicio de sesión y de encabezado, e incluyen botones para avanzar y retroceder en las páginas web. Las páginas básicas también incluyen un botón de inicio que redirecciona a una página de navegación. Asimismo, puede usar los controles de navegación que se suministran con el dispositivo móvil para navegar. Así, por ejemplo, la página básica de diagnóstico tiene el aspecto siguiente en orientación vertical:

La resolución mínima para visualizar una página básica es de 240 x 240 píxeles.



Tenga en cuenta que las ilustraciones de páginas web estándar de este capítulo representan el aspecto de las páginas web del PC estándar. La mayoría de páginas web estándar tienen páginas básicas equivalentes.

12.7.3 Inicio de sesión y privilegios de usuario

Cada página web estándar del PC proporciona una ventana de inicio de sesión encima del panel de navegación. Por razones de espacio, las páginas web básicas proporcionan una página de inicio de sesión distinta. El S7-1200 soporta varios nombres de usuario con varios niveles de acceso (privilegios):

- Consultar diagnóstico
- Leer variables
- Escribir variables
- Leer estados de variables
- Escribir estados de variables
- Abrir páginas web del usuario
- Escribir en páginas web del usuario
- Leer archivos
- Escribir o borrar archivos
- Cambiar el estado operativo
- Hacer parpadear LED
- Actualizar el firmware
- Crear una copia de seguridad del PLC
- Restaurar el PLC con la copia de seguridad del firmware
- Acceder a parámetros (F-Admin) solo para CPU S7-1200 de seguridad

Las funciones de usuario, los niveles de acceso asociados (privilegios) y las contraseñas se configuran (Página 848) en las propiedades de la administración de usuarios del servidor de la configuración de dispositivos de STEP 7 de la CPU.

Inicio de sesión

STEP 7 proporciona un usuario predeterminado llamado "Cualquiera" sin contraseña. Por defecto, este usuario no dispone de privilegios adicionales y solo puede visualizar las páginas web estándar de Inicio (Página 863) e Introducción (Página 862). Sin embargo, puede conceder privilegios adicionales al usuario "Cualquiera" así como a otros usuarios que configure:

⚠ ADVERTENCIA

Acceso no autorizado a la CPU mediante el servidor web

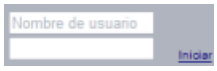
Los usuarios con derechos de pleno acceso o de pleno acceso (incl. seguridad positiva) a la CPU pueden leer y escribir variables PLC. Independientemente del nivel de acceso a la CPU, los usuarios de servidor web pueden tener derechos para leer y escribir variables PLC. El acceso no autorizado a la CPU o el cambio de variables PLC a valores no válidos puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

Los usuarios autorizados pueden realizar cambios en el estado operativo, escribir datos de PLC y actualizar el firmware. Siemens recomienda observar las siguientes consignas de seguridad:

- Niveles de acceso a la CPU (Página 163) protegidos por contraseña e ID de usuario de servidor web (Página 848) con contraseñas seguras.
- Las contraseñas seguras tienen como mínimo doce caracteres, no son triviales ni pueden adivinarse fácilmente e incluyen como mínimo tres de los siguientes tipos de caracteres:
  - mayúsculas
  - minúsculas
  - números
  - caracteres especiales
- Una contraseña trivial es una contraseña que puede adivinarse fácilmente. Generalmente se basa en características del usuario, como el nombre de una mascota, el apellido o la compañía en la que trabaja. Ejemplo: Siemens1$, Junio2015 o Qwerty1234.
- La mejor manera de crear una contraseña segura pero fácil de recordar es incluir el uso de frases breves sin sentido y mezclar varias palabras aleatoriamente. Ejemplo: PC;Casa#R3d
- Permitir el acceso al servidor web solo con el protocolo HTTPS.
- No amplíe los derechos mínimos predeterminados del usuario "Todos" del servidor web.
- Realice comprobaciones de errores y de rango de las variables usadas en la lógica del programa, puesto que los usuarios de páginas web pueden cambiar variables PLC a valores no válidos.
- Utilice una red privada virtual (VPN = Virtual Private Network) para conectar al PLC S7-1200 un servidor web desde una ubicación externa a la red protegida.

Para ejecutar ciertas acciones como cambiar el estado operativo del controlador, escribir valores en la memoria o actualizar el firmware de la CPU debe contar con los privilegios necesarios. Tenga en cuenta que si ha establecido el nivel de protección de la CPU (Página 163) en "Sin

acceso (protección completa)", el usuario "Cualquiera" no tendrá autorización para acceder al servidor web, independientemente de los ajustes de derechos de usuario del servidor web.



El cuadro de inicio de sesión se encuentra cerca de la esquina izquierda superior de cada página web estándar cuando se visualiza desde un PC o un dispositivo móvil ancho.



La página de inicio de sesión es una página aparte en dispositivos móviles pequeños que visualizan las páginas básicas. Se puede seleccionar desde la página de inicio.

Para iniciar sesión, proceda del siguiente modo:

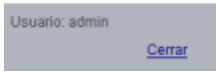
1. Introduzca el nombre de usuario en el campo Nombre de usuario.
2. Introduzca la contraseña de usuario en el campo Contraseña.

La sesión iniciada expirará después de treinta minutos de inactividad. Si la página que está cargada se actualiza constantemente, el tiempo activo de la sesión iniciada se pone en cero, lo que impedirá que expire.

Nota

Si se producen problemas al iniciar sesión, descargue el certificado de seguridad de Siemens (Página 854) desde la página de introducción (Página 862). A continuación podrá iniciar una sesión sin errores.

Cierre de sesión





Para cerrar sesión, haga clic en el enlace "Cerrar" desde cualquier página cuando la esté viendo desde un PC o un dispositivo móvil ancho.

Desde las páginas básicas, navegue hasta la página de inicio/cierre de sesión de la página de inicio y pulse el botón "Cerrar sesión".

Después de cerrar sesión solo es posible acceder a páginas web estándar y visualizarlas de acuerdo con los privilegios del usuario "Cualquiera". Cada una de las descripciones de las páginas web estándar especifica los privilegios necesarios para la página en cuestión.

Nota

Cierre de sesión antes de cerrar el servidor web

Si ha iniciado sesión en el servidor web, asegúrese de cerrarla antes de cerrar el navegador web. El servidor web soporta un máximo de siete sesiones activas al mismo tiempo.

12.7.4 Introducción

La página de introducción es la pantalla de bienvenida de las páginas web estándar del S7-1200.



Desde esta página, haga clic en "Intro" para acceder a las páginas web estándar S7-1200. En la parte superior de la pantalla hay enlaces de sitios web afines de Siemens, así como un enlace

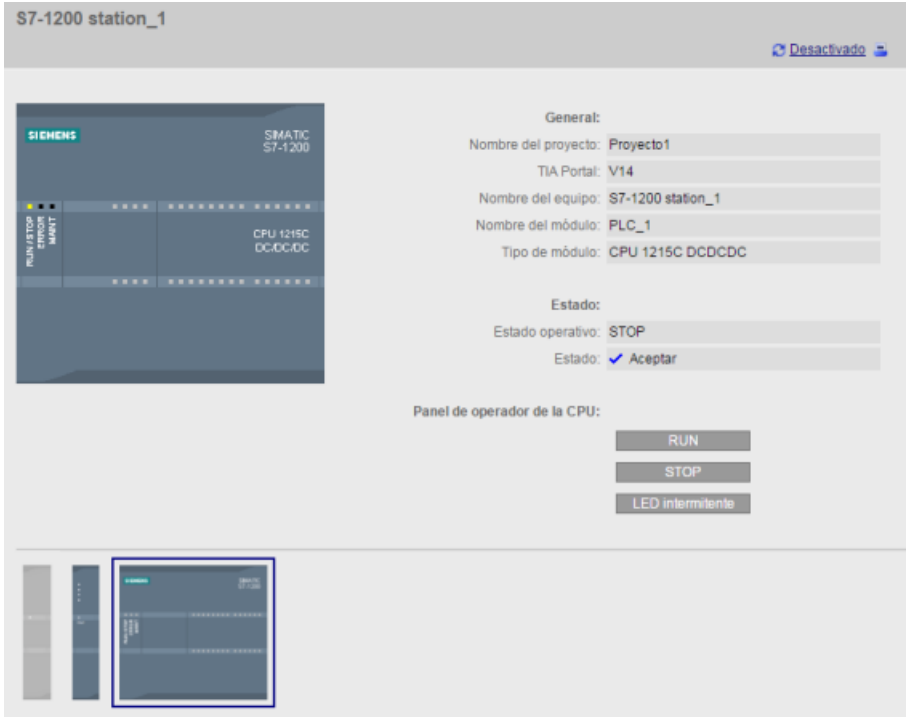
para cargar el certificado de seguridad de Siemens (Página 854). Existe la posibilidad de saltar la página de introducción en futuros accesos al servidor web.

12.7.5 Arranque

La página de inicio muestra una representación de la CPU o el CP con el que se ha realizado la conexión e indica la información general sobre el dispositivo y la versión del TIA Portal que se han utilizado para descargar el proyecto a la CPU. Para la CPU, puede usar los botones para cambiar el estado operativo y para que parpadeen los LED, si ha iniciado sesión (Página 859) con el derecho (Página 848) "Cambiar estado operativo".

La parte inferior de la pantalla es visible si han configurado e instalado módulos CP aptos para servidor web (Página 853) en el rack local con la CPU S7-1200. Es posible moverse por encima y hacer clic en un módulo CP apto para servidor web para acceder a las páginas web estándar. Consulte la documentación para su módulo PC para obtener más información sobre las páginas web del módulo CP. El nombre del módulo CP se ve cuando se pasa el puntero del ratón por encima de él.

El servidor web también muestra cualquier otro módulo CM y CP del rack local, pero no es posible hacer clic en ellos porque no contienen páginas web. El aspecto de estos CMs o CPs está en gris claro (atenuados), lo que indica que son módulos solo visibles pero que no se puede hacer clic en ellos.



Tenga en cuenta que las CPUs S7-1200 de seguridad positiva muestran en esta página datos adicionales relacionados con la seguridad funcional.

12.7.6 Diagnóstico

La página de diagnóstico muestra características que identifican la CPU, ajustes de configuración para la protección de know-how y el uso de la memoria de carga, trabajo y remanente:

La página consta de tres fichas:

- Identificación: características para identificar el módulo e información sobre la planta y la ubicación de STEP 7
- Protección del programa: estado de la protección de know-how y vinculación a CPU, que puede ser útil para planificar los repuestos, así como ajustes de configuración de STEP 7 para permitir o impedir la copia de la memoria de carga interna en la memoria de carga externa (SIMATIC Memory Card).
- Memoria: uso de la memoria de carga, trabajo y remanente

Para las CPU F hay una ficha Fail-safe adicional.

Para ver la página Identificación se necesita el derecho (Página 848) de "Consultar diagnóstico".

Ficha Identificación



Ficha Protección del programa

La ficha de protección del programa incluye la información siguiente:

- Protección de know-how (Página 167): Indica si se ha configurado la protección de know-how para alguno de los bloques de programa en STEP 7
- Vinculación (Página 168): Indica si el programa se ha vinculado a la CPU o a la SIMATIC Memory Card
- Copia del programa en la Memory Card (Página 167): indica si se ha activado la posibilidad de copiar el programa de la memoria de carga interna a la memoria de carga externa (SIMATIC Memory Card)



Ficha Memoria



Ficha Fail-safe

Encontrará información sobre la ficha Fail-safe de la página de diagnóstico en Manual S7-1200 Functional Safety (https://support.industry.siemens.com/cs/ww/es/view/104547552/en).

12.7.7 Búfer de diagnóstico

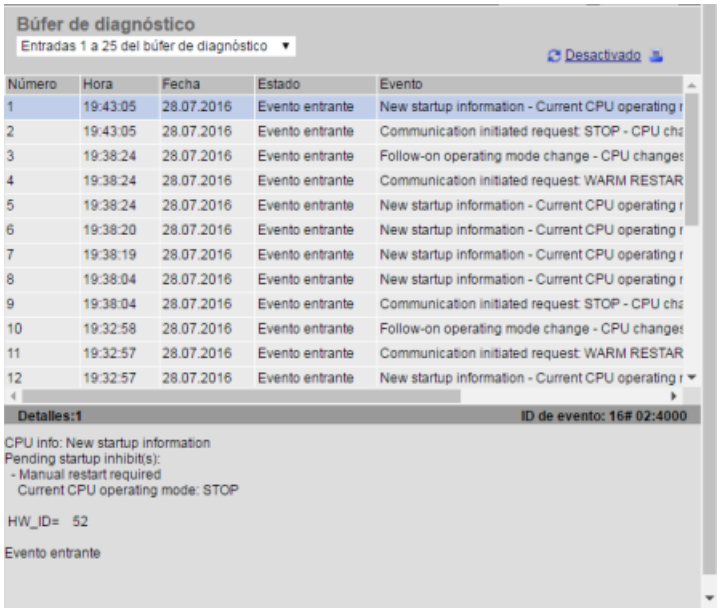
La página Búfer de diagnóstico muestra los eventos de diagnóstico. El evento más nuevo es el número 1 en la parte superior. El más antiguo es el número 50. En el selector de la izquierda es posible seleccionar el rango de entradas del búfer de diagnóstico que debe mostrarse, 1-25 o 26-50. En el selector de la derecha es posible seleccionar si desea ver las horas en formato UTC o local del PLC. En la parte superior de la página se muestran las entradas de diagnóstico con la hora y fecha de cuándo se ha producido el evento.

Se puede seleccionar cualquier entrada de diagnóstico individual y visualizar información detallada de la misma en la parte inferior de la página. Tenga en cuenta que el idioma de visualización de las entradas del búfer de diagnóstico depende de los ajustes de la configuración del dispositivo para el soporte multilingüe (Página 158).

Debe configurar los idiomas utilizados para textos de diagnóstico en el proyecto TIA Portal descargado en el PLC. Esta configuración está disponible en las propiedades del PLC en "Multilingüe". Cada idioma descargado debe estar asociado a un idioma de servidor web soportado. El PLC está limitado a 2 idiomas descargados.



La página del búfer de diagnóstico también incluye un botón para guardar el búfer de diagnóstico en un archivo CSV. En el ajuste predeterminado, el servidor web guarda el archivo en formato separado por comas en ASLog.csv, dentro de la carpeta de descargas. El archivo contiene el búfer de diagnóstico completo en el momento de guardar. El archivo puede guardarse las veces que se quiera y es posible guardar varios archivos. El archivo del búfer de diagnóstico se abre con Microsoft Excel o con cualquier editor de textos.



Para ver la página del búfer de diagnóstico se necesita el privilegio (Página 848) de "consulta de diagnóstico".

12.7.8 Información del módulo

La página de información del módulo:

- Ofrece información sobre todos los módulos en el rack local. En la sección superior de la pantalla se muestra un resumen de los módulos en base a la configuración de dispositivos de STEP 7 y en la sección inferior se muestra la información de estado, identificación y firmware del módulo seleccionado en base al correspondiente módulo conectado.
- Ofrece la posibilidad de realizar una actualización del firmware.
- Ofrece información sobre sistemas de periferia descentralizada.

Para ver la página Información del módulo se necesita el privilegio (Página 848) de "Consultar diagnóstico".

Información del módulo: Pestaña Estado

La pestaña Estado ubicada en el área inferior de la página de información del módulo muestra una descripción del estado actual del módulo seleccionado en el área superior. Si la sección está vacía, el módulo no tiene ningún estado de diagnóstico pendiente.



Iconos de estado de los módulos

Para cada módulo, la columna de estado de la sección superior muestra un icono que indica el estado de dicho módulo:

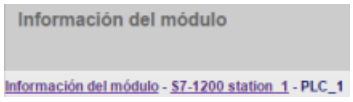
Icono	Significado
	No hay fallos
	Desactivado
	Mantenimiento necesario
	Mantenimiento solicitado
	Error
	La CPU no puede acceder al módulo o dispositivo (para dispositivos distintos de la CPU)
	La CPU ha establecido una conexión con el dispositivo, pero el estado del módulo es desconocido (para dispositivos distintos de la CPU)
	Los datos de entrada y salida no están disponibles porque el submódulo ha bloqueado sus canales E/S (para dispositivos distintos de la CPU)

Navegación jerárquica

Se puede seleccionar un enlace en la sección superior para navegar hasta la información de ese módulo en particular. Los módulos con submódulos tienen enlaces para cada submódulo. El tipo de información que se visualiza varía en función del módulo seleccionado. Así, por ejemplo, el cuadro de diálogo de información del módulo muestra inicialmente el nombre de la estación S7-1200, un indicador de estado y un comentario. Si se despliega hasta la CPU, la información del módulo muestra el nombre de las entradas y salidas digitales y analógicas que tiene este modelo de CPU, información de direccionamiento para las E/S, indicadores de estado, números de slot y comentarios.

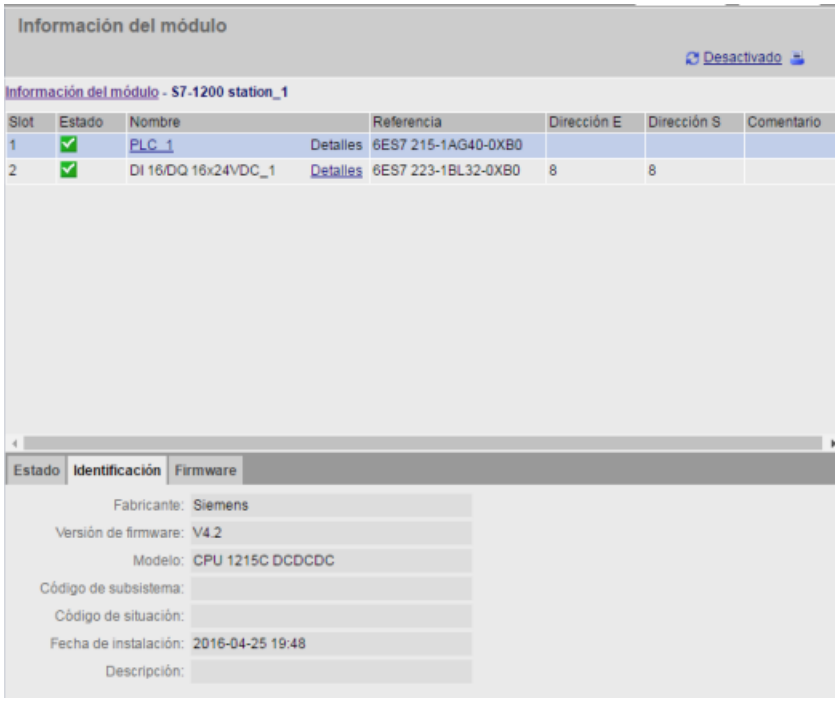
Slot	Estado	Nombre		Referencia	Dirección E	Dirección S	Comentario
1.1	✔	DI14/DQ10_1	Detalles	6ES7 214-1AG40-0XB0	0	0	
1.2	✔	AI2_1	Detalles	6ES7 214-1AG40-0XB0	64	---	
1.16	✔	HSC_1	Detalles	6ES7 214-1AG40-0XB0	1000	---	
1.17	✔	HSC_2	Detalles	6ES7 214-1AG40-0XB0	1004	---	
1.18	✔	HSC_3	Detalles	6ES7 214-1AG40-0XB0	1008	---	
1.19	✔	HSC_4	Detalles	6ES7 214-1AG40-0XB0	1012	---	
1.20	✔	HSC_5	Detalles	6ES7 214-1AG40-0XB0	1016	---	
1.21	✔	HSC_6	Detalles	6ES7 214-1AG40-0XB0	1020	---	
1.32	✔	Pulse_1	Detalles	6ES7 214-1AG40-0XB0	---	1000	

Al navegar, la página de información del módulo muestra la ruta que se ha seguido. Se puede hacer clic en cualquier enlace de esta ruta para regresar a un nivel superior.



Información del módulo: Pestaña Identificación

La pestaña Identificación muestra los datos de identificación y mantenimiento (I&M) del módulo seleccionado.



Tenga en cuenta que si se hace clic en la parte superior de un módulo F de E/S, la parte inferior tendrá una pestaña denominada Seguridad. En esta pestaña es posible especificar datos relacionados con el módulo seleccionado, tal como se describe en Manual de seguridad funcional S7-1200 (https://support.industry.siemens.com/cs/ww/es/view/104547552/en).

Información del módulo: Pestaña Firmware

En la pestaña Firmware de la página de información del módulo figura información sobre el firmware del módulo seleccionado. Si tiene el privilegio (Página 848) "Actualizar el firmware", también puede actualizar el firmware de la CPU u otros módulos del rack local que permitan la actualización de firmware o los módulos PROFINET IO. Para módulos remotos se ve la información del firmware, pero no es posible actualizarlo.

Nota

Para actualizar el firmware de la CPU solo se pueden actualizar CPU S7-1200 de la versión 3.0 o posteriores.



Actualización del firmware

Para actualizar el firmware, la CPU debe estar en modo STOP. Cuando la CPU está en estado operativo STOP, haga clic en el botón Examinar para ir hasta un archivo de firmware y seleccionarlo. Las actualizaciones de firmware están disponibles en la página web de Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/).

Durante la actualización, la página muestra un mensaje que indica que la actualización está en curso. Una vez que ha finalizado la actualización, la página mostrará la referencia y el número de versión del firmware actualizado. Si ha actualizado el firmware para la CPU o una Signal Board, el servidor web reiniciará la CPU.

También es posible actualizar el firmware utilizando uno de los métodos siguientes:

- Las herramientas online y de diagnóstico de STEP 7 (Página 1204)
- Una SIMATIC Memory Card (Página 132)
- La SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/es/view/98161300)

Nota

Posibles problemas al ejecutar una actualización del firmware desde el servidor web.

En caso de interrumpirse la comunicación durante la actualización del firmware desde el servidor web, el navegador web utilizado puede visualizar un mensaje, en el que se le pregunta si desea abandonar la página actual o bien permanecer en ella. Para evitar posibles problemas, seleccione la opción de permanecer en la página actual.

Si se cierra el navegador web mientras se está actualizando el firmware desde el servidor web, no se podrá cambiar el modo de operación de la CPU al modo RUN. Si esto ocurre, es preciso apagar y encender nuevamente la CPU para poder pasar la CPU al modo RUN.

Función de rearranque suave

La función de servidor web "Rearranque suave" está disponible como parte del proceso de actualización del firmware de la versión V4.5 y superior de la CPU S7-1200. Esta función solo

puede utilizarse durante el proceso de actualización del firmware. No es posible invocar la función "Rearranque suave" en ningún otro momento.

12.7.9 Comunicación

La página de comunicación muestra los parámetros de la CPU conectada, estadísticas de comunicación, recursos e información sobre conexiones.

Para ver la página Comunicación se necesita el derecho de "Consultar diagnóstico".

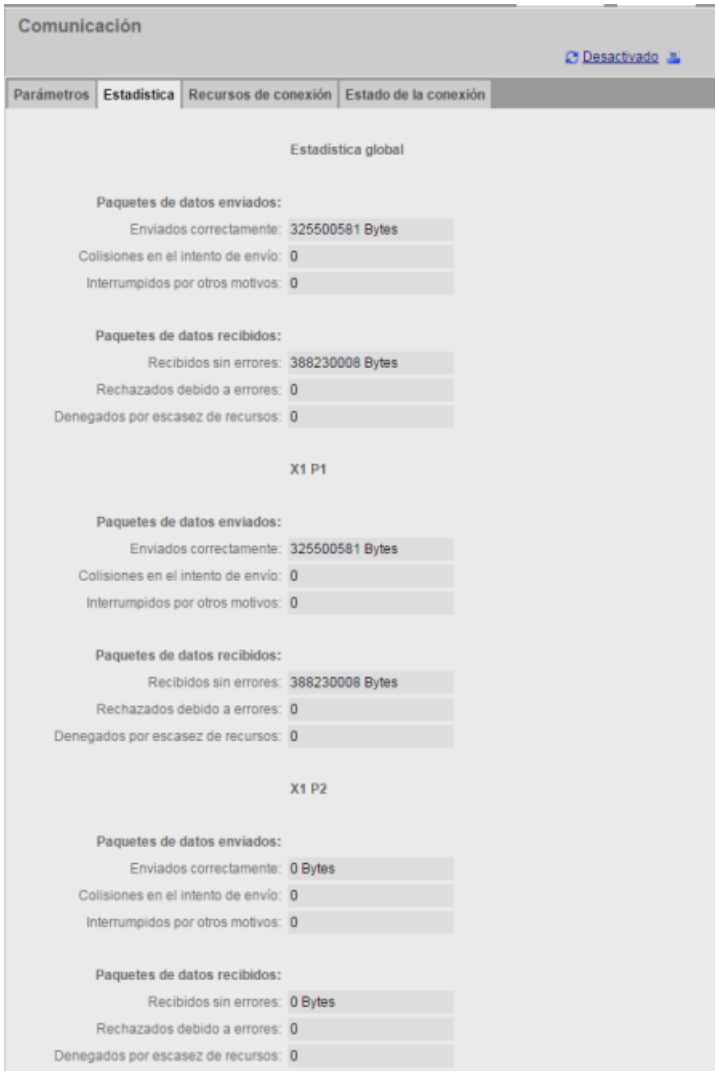
Ficha Parámetro

La ficha Parámetro muestra la dirección MAC de la CPU, la dirección IP y los ajustes IP de la CPU, además de las propiedades físicas:



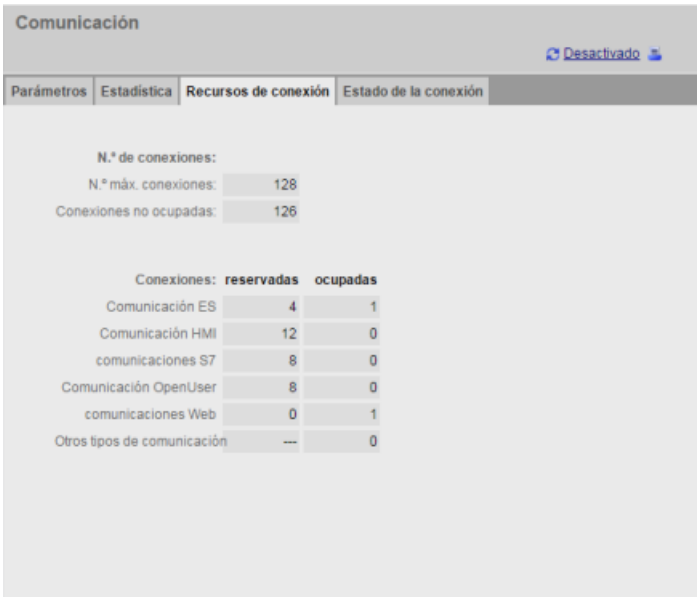
Ficha Estadísticas

La ficha Estadísticas muestra las estadísticas de comunicación entrante y saliente:



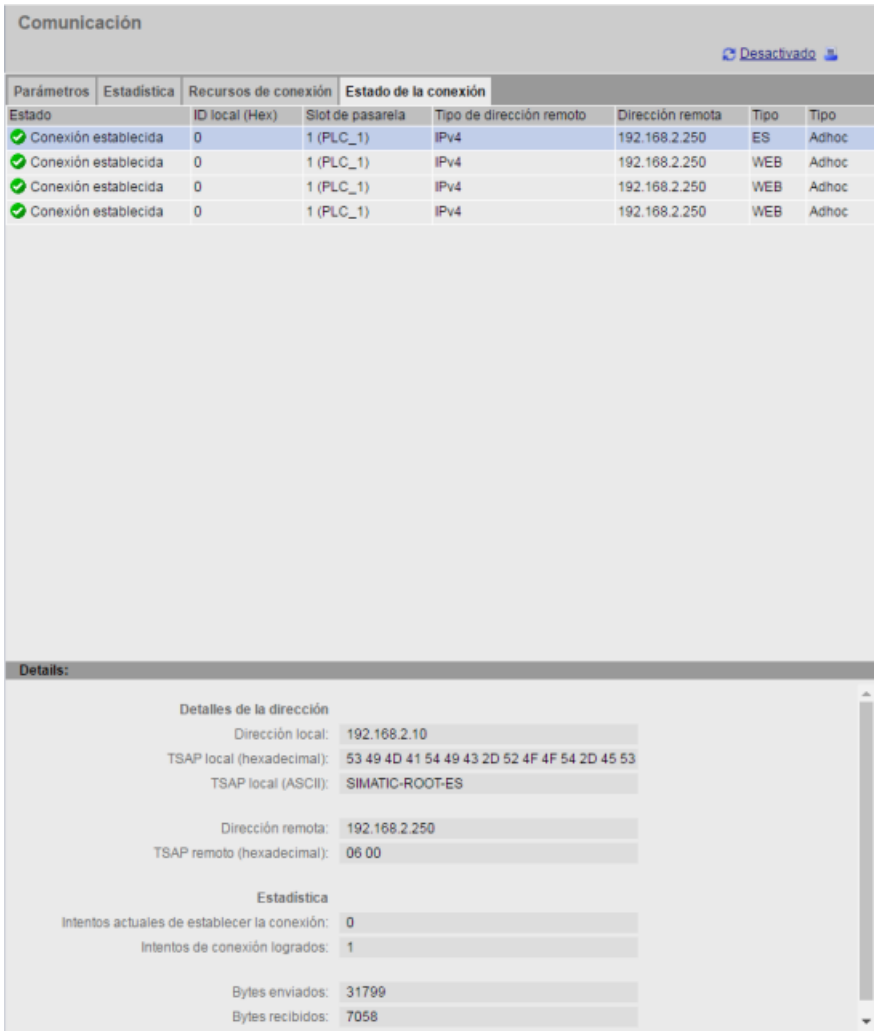
Ficha Recursos de conexión

La ficha Recursos de conexión muestra información sobre el número total de recursos de conexión y cómo están asignados a los diferentes tipos de comunicación:



Ficha Estado de conexión

La ficha Conexiones muestra las conexiones de la CPU y los detalles de la conexión seleccionada.

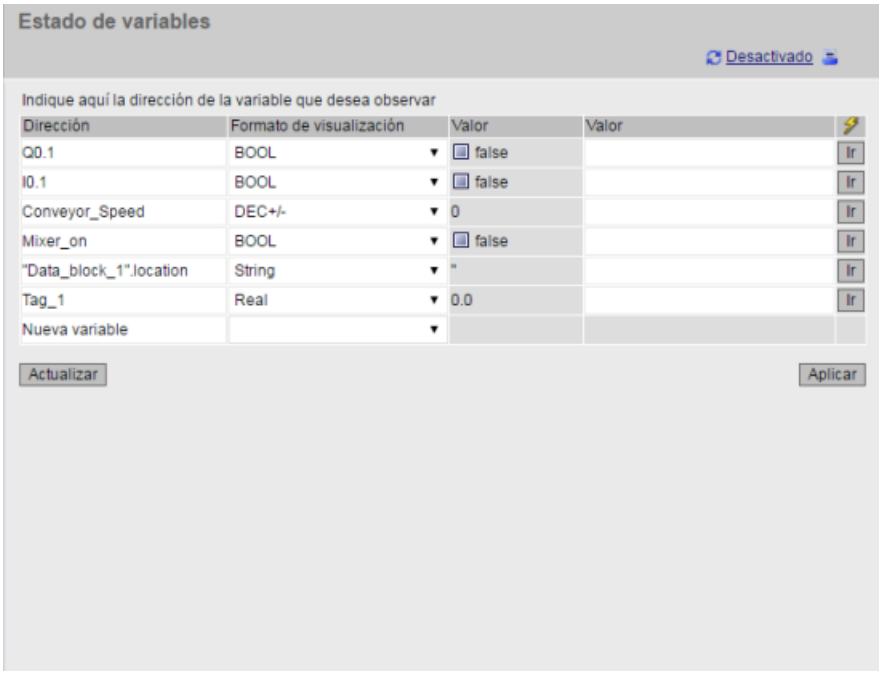


12.7.10 Estado de variable

La página Estado de variable permite ver cualquier dato de memoria o E/S en la CPU. Se puede introducir una dirección directa (como %I0.0), un nombre de variable PLC o una variable de un bloque de datos determinado. Para variables de bloque de datos se pone el nombre del bloque entre comillas dobles. Para cada valor de observación se puede seleccionar un formato de visualización de datos. Se puede seguir introduciendo y especificando tantos valores como se desee dentro de los límites de la página. Los valores de observación se visualizan automáticamente. Puede hacer clic en el botón "Actualizar" en cualquier momento para actualizar todos los valores de observación. Si ha activado la actualización automática en STEP 7 (Página 846), puede hacer clic en el icono "Off" en el área superior derecha de la página para desactivarla. Cuando la actualización automática está desactivada se puede hacer clic en "On" para activarla nuevamente.

Para ver la página Estado de variables se necesita el derecho "Leer estado de variable".

Si inicia sesión como usuario con el derecho (Página 859) "Escribir estado de variable", también puede modificar los valores de datos. Introduzca los valores que desee ajustar en el campo "Forzar valor" apropiado. Haga clic en el botón "Ir a" junto a un valor para escribirlo en la CPU. También es posible introducir varios valores y escribirlos conjuntamente en la CPU, para ello haga clic en "Aplicar". Los botones y las etiquetas de columna para hacer modificaciones solo aparecen si tiene el derecho "Escribir estado de variable".



Si sale de la página Estado de variables y regresa, la página Estado de variables no conserva las entradas. Es posible marcar la página y regresar al marcador para ver las mismas entradas. Si no se marca la página, deberá introducir nuevamente las variables.

Para valores que se observan o fuerzan con frecuencia, es aconsejable utilizar una tabla de observación (Página 877) en su lugar.

Nota

Al utilizar la página Estado de variables, tenga en cuenta lo siguiente:

- Delimite todas las modificaciones de la cadena con comillas simples.
- La página Estado de variables puede observar y forzar variables que contengan alguno de los caracteres siguientes: &, <, (, +, ,(coma), ., [,], $, o %, siempre que el nombre de la variable se ponga entre comillas dobles, por ejemplo "Reloj_2.5Hz".
- Para observar o modificar un solo campo de una variable DTL, incluya el campo en la dirección, por ejemplo "Data_block_1".DTL_tag.Año. Introduzca un valor entero para el valor de forzado de acuerdo con el tipo de datos del campo específico de la DTL. Así, por ejemplo, el campo Año es un UInt.
- En cada página se pueden introducir como máximo 50 variables.
- Si el nombre de una variable presenta caracteres especiales que no se admiten como entrada en la página Estado de variables, existe la posibilidad de poner el nombre de la variable entre comillas dobles. En la mayoría de los casos, la página reconocerá entonces el nombre de la variable.

Consulte también

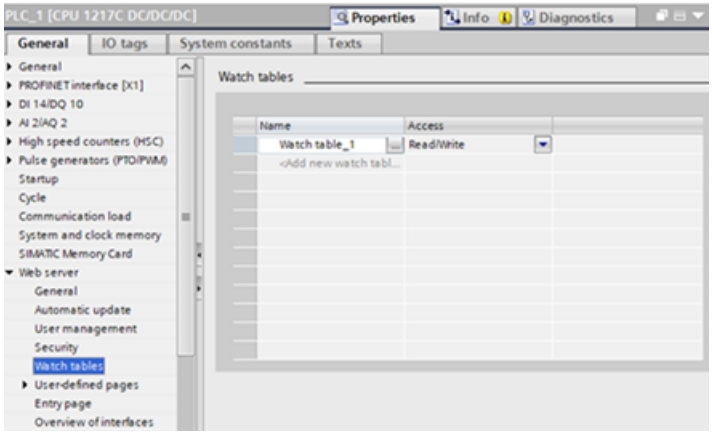
Reglas para introducir nombres de variables y valores (Página 934)

12.7.11 Tablas de observación

El servidor web permite acceder a tablas de observación que se han configurado en STEP 7 y se han descargado en la CPU. Las tablas de observación con 50 entradas o menos ofrecen el mejor rendimiento en el servidor web.

Configuración de STEP 7 para seleccionar tablas de observación para el servidor web

Desde la configuración de dispositivos de la CPU en STEP 7 es posible agregar las tablas de observación que el servidor web debe poder visualizar. Para cada tabla de observación que se seleccione en la lista de tablas existentes deben seleccionarse también los derechos de lectura o lectura/escritura. Una vez descargadas en la CPU, las tablas de observación que tienen el derecho de lectura solo pueden visualizarse, pero si se ha seleccionado el derecho de lectura/escritura es posible observar y forzar las variables de las tablas de observación.



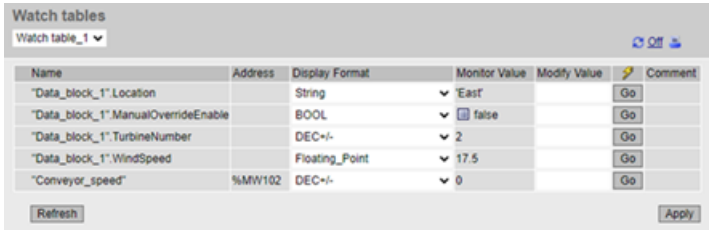
Una vez finalizada la configuración de tablas de observación en la sección de configuración de dispositivos del servidor web, descargue la configuración hardware en la CPU.

Visualizar tablas de observación desde el servidor web

Desde el servidor web, si se dispone del derecho (Página 848) "Leer variables" se pueden seleccionar las "tablas de observación" en el menú de navegación para acceder a las tablas de observación que se han configurado y descargado en la CPU. Si se ha descargado más de una tabla de observación, en la lista desplegable puede seleccionarse la que debe visualizarse. El servidor web muestra la tabla de observación que se ha creado en STEP 7 y los valores actuales de acuerdo con el formato de visualización. Es posible modificar el formato de visualización, pero cuando se regresa a la página de tablas de observación, el servidor web recupera los formatos de visualización predeterminados de la tabla de observación de STEP 7.

Forzar variables de tablas de observación desde el servidor web

Si se ha descargado una tabla de observación con el nivel de acceso "lectura/escritura" y se ha iniciado sesión en el servidor web con el derecho (Página 848) "Escribir variables", también es posible forzar valores de variables del mismo modo que se hace en una tabla de observación de STEP 7. Existe la posibilidad de forzar valores de variables individuales y hacer clic en "Ir a" para forzar únicamente ese valor o bien pueden introducirse varios valores y hacer clic en "Aplicar" para forzarlos todos a la vez.



Nota

Ventajas de las tablas de observación para forzar variables

Para que un usuario pueda forzar variables y variables de bloques de datos en la CPU desde una tabla de observación, hay que configurar la tabla de observación en las propiedades del servidor web dentro de la configuración de dispositivos de STEP 7 y hay que tener acceso de lectura/escritura. De este modo, es posible restringir las variables que puede forzar un usuario con el derecho "Escribir variables" únicamente a aquellas variables que hay en las tablas de observación configuradas en el servidor web.

La página Estado de variable (Página 875), en cambio, permite a todos los usuarios con el derecho "Escribir estado de variable" escribir en cualquier variable o variable de bloque de datos de la CPU.

Una configuración cuidadosa de los derechos de administración de usuarios (Página 848) del servidor web permite proteger el acceso a los datos del PLC.

Consulte también

Reglas para introducir nombres de variables y valores (Página 934)

12.7.12 Copia de seguridad online

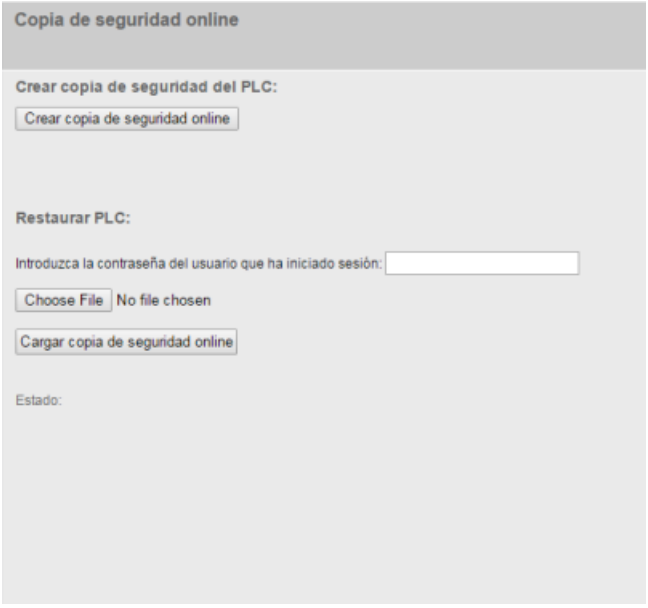
La página web estándar Copia de seguridad online permite realizar una copia de seguridad del proyecto de STEP 7 para el PLC online y también restaurar una copa de seguridad del PLC realizada previamente. Antes de crear o restaurar una copia de seguridad, ponga el PLC en estado operativo STOP y cancele todas las comunicaciones con el PLC, como el acceso HMI y el acceso al servidor web. Si la CPU no está en estado STOP, las funciones de copia de seguridad y restauración piden confirmación para poner la CPU a STOP antes de continuar.

Si se ha accedido a la página de copia de seguridad online desde uno de los módulos CP activados para web, es posible realizar una copia de seguridad pero no restaurarla.

Nota

También pueden llevarse a cabo operaciones de copia de seguridad y restauración desde STEP 7 (Página 1238). Consulte estos temas para obtener una descripción completa de los datos que pueden asegurarse y restaurarse. La SIMATIC Automation Tool (SAT) también ofrece la función de copia de seguridad y restauración.

Cuando se hacen copias de seguridad de archivos desde el servidor web, el PC o dispositivo guarda los archivos de copia de seguridad en la carpeta predeterminada para descargas. Cuando se hacen copias de seguridad desde STEP 7, STEP 7 guarda los archivos dentro del proyecto de STEP 7. No es posible restaurar archivos de copia de seguridad de STEP 7 desde el servidor web ni tampoco a la inversa. Sin embargo, sí es posible guardar archivos de copia de seguridad de STEP 7 directamente en la carpeta de descargas del PC o dispositivo. De hacerse así, después podrán restaurarse estos archivos desde el servidor web.



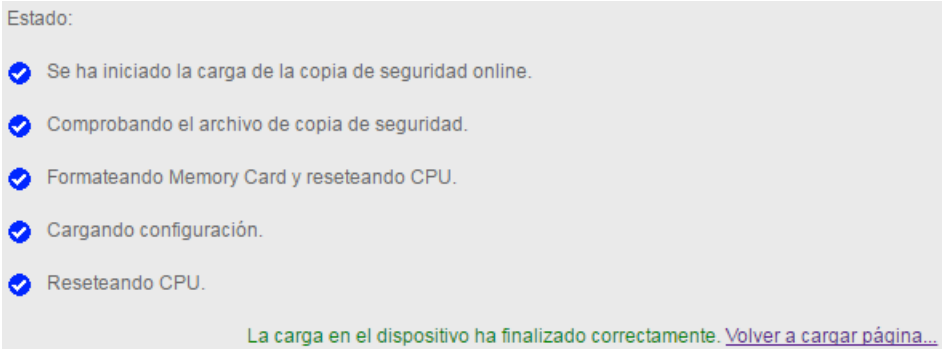
Crear copia de seguridad del PLC

Desde la sección Crear copia de seguridad del PLC de la página, haga clic en el botón "Crear copia de seguridad online" para hacer una copia de seguridad del proyecto almacenado actualmente en el PLC. Esta función requiere el derecho (Página 848) de usuario "Copia de seguridad CPU". Si la CPU está en RUN y debe pasar al estado operativo STOP, también se necesita el derecho "Cambiar estado operativo". El PC o dispositivo guardará el archivo de copia de seguridad en la ubicación predeterminada para descargas. En función de los ajustes del navegador y el dispositivo, es posible que se le pida que guarde el archivo.

Restaurar PLC

Desde la sección Restaurar PLC de la página, introduzca la contraseña de usuario del servidor web y haga clic en el botón "Examinar" o "Seleccionar archivo" (en función del navegador) para seleccionar un archivo de copia de seguridad creado previamente. Haga clic en el botón "Cargar copia de seguridad online" y confirme la pregunta de si el archivo debe cargarse en el PLC conectado. Esta página requiere el derecho (Página 848) de usuario "Restaurar CPU". Si la CPU está en RUN y debe pasar al estado operativo STOP, también se necesita el derecho "Cambiar estado operativo".

Cuando la operación de restauración está en marcha se muestran una serie de avisos de progreso y debe volver a introducir sus datos de usuario y contraseña. Después de que todos los pasos del proceso hayan finalizado correctamente, se muestran los siguientes indicadores de finalización y un enlace para volver a cargar la página:



⚠ ADVERTENCIA

Restaurar copias de seguridad con contenido desconocido

Si se restauran copias de seguridad con contenido desconocido pueden producirse daños y lesiones graves en caso de funcionamientos incorrectos o errores de programa.

Además, si se restaura una copia de seguridad que no tiene activado el servidor web en la configuración de dispositivos de la CPU, no será posible acceder a la CPU desde el servidor web.

Asegúrese de que la copia de seguridad contiene una configuración con contenido conocido.

Nota

Restaurar una copia de seguridad con una dirección IP diferente para la CPU

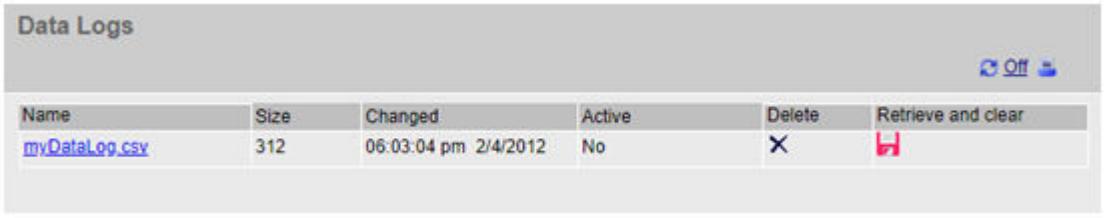
Si se intenta restaurar una copia de seguridad en la que la dirección IP de la CPU es distinta de la dirección IP de la CPU actual, el servidor web no podrá mostrar el aviso de que la restauración ha finalizado. Si pasan más de cinco minutos y se sigue visualizando el aviso "Reseteando la CPU", introduzca la nueva dirección IP que corresponda a la dirección del archivo de copia de seguridad. Ahora la CPU tiene esta dirección y se puede recuperar el acceso al servidor web.

12.7.13 Página Registros de datos

La página Registros de datos permite interactuar con registros de datos del programa STEP 7. Es posible:

- Ver una lista de todos los registros de datos del PLC
- Descargar un registro de datos del PLC al equipo
- Borrar un registro de datos del PLC
- Recuperar y borrar correctamente un registro de datos

Los registros de datos se muestran en orden alfabético ascendente, sin tener en cuenta las mayúsculas y minúsculas. La lista de registros de datos se pagina en incrementos de 50 registros de datos.



Nota

Administración de registros de datos

No guarde más de 1000 registros en un sistema de archivos. Si se excede este número, existe la posibilidad de que el servidor web no tenga recursos suficientes en la CPU para mostrar los registros.

Si resulta que la página web Registros de datos no puede visualizar los registros, deberá poner la CPU en estado STOP para ver y borrar registros.

Administre los registros para garantizar que solo se guarden aquellos que deben mantenerse y no exceda el número de 1000 registros.

Trabajar con un registro de datos en Excel

El archivo de registro es de formato separado por comas (CSV) para sistemas británicos y estadounidenses. Para abrir el archivo con Excel en sistemas que no son ni estadounidenses ni británicos, debe importar el archivo a Excel aplicando una configuración específica.

Estado activo

La columna "Activo" en la página de registros de datos indica "Sí" cuando hay un bloque de control de registros de datos en la CPU asociada a dicho archivo, y "No" cuando no lo hay. Es posible poner el estado activo a "Sí" incluso si el programa STEP 7 no está interactuando con el archivo.

Si el programa STEP 7 tiene un registro abierto o está escribiendo en un registro, el servidor web no puede borrar, descargar ni recuperar y borrar el archivo de registro. Además, mientras el servidor web está descargando un registro mediante descarga o mediante Recuperar y borrar,

no es posible realizar otra operación con registros hasta que la descarga ha finalizado o se ha cancelado. El servidor web muestra un mensaje de error "Aplicación ocupada".

Descarga de un archivo de registro

Un archivo de registro se descarga haciendo clic en el nombre del archivo. El servidor web muestra un mensaje de error si el archivo ya no existe o si hay otra descarga en curso. El mensaje de error permanece en la página hasta que se inicia una operación para volver a cargar la página Registros de datos. Las operaciones siguientes hacen que el servidor web vuelva a cargar la página Registros de datos:

- Actualizar la página de registros de datos saliendo de ella y volviendo a ella
- Modificar la selección de paginación actual de los registros de datos
- Borrar correctamente un registro de datos
- Recuperar y borrar correctamente un registro de datos

Nota

Mensaje de error de registros de datos

La funcionalidad de actualización automática del servidor web no elimina el mensaje de error de la página.

Los escenarios siguientes generan mensajes de error en la página Registros de datos:

Operación	Condición de fallo	Mensaje de error
Descargar Recuperar y borrar	• El archivo no existe • Nombre de archivo no válido • Método de solicitud HTTP no válido • La SMC está protegida contra escritura (solo Recuperar y borrar)	Error al descargar el archivo
Borrar	• Método de solicitud HTTP no válido • Nombre de archivo no válido • El archivo no existe	Error al borrar el archivo
Borrar	• La SMC está protegida contra escritura	Error al borrar el archivo: la Memory Card está protegida contra escritura
Descargar Borrar Recuperar y borrar	• No tiene los permisos necesarios • El método de solicitud HTTP no es POST o GET • Parámetro ACTION no válido en URL	Operación de archivo no permitida
Descargar Borrar Recuperar y borrar	• Campo de referente HTTP no válido o ausente	Operación de archivo no permitida: no hay referente

Operación	Condición de fallo	Mensaje de error
Descargar Borrar Recuperar y borrar	• El programa de usuario mantiene abierto el archivo de registro • Se está descargando un registro de datos	Aplicación ocupada
Descargar Borrar Recuperar y borrar	• Error interno de PLC imprevisto	Error interno

Borrado de un registro de datos

Un registro de datos se borra haciendo clic en el icono  de la columna Borrar para un registro concreto. Para borrar el registro de datos, confirme la operación de borrado en el cuadro de diálogo Borrar.

Recuperar y borrar un registro de datos

Para abrir un registro de datos y eliminar todas las entradas, haga clic en el icono Recuperar y borrar . Para recuperar (consultar) el archivo de registro y vaciar su contenido, confirme la operación en el cuadro de diálogo Recuperar y borrar.

Después de confirmar la operación, el servidor web permite descargar el contenido del archivo de registro. El cuadro de diálogo Guardar archivo permite elegir si el registro debe guardarse o no. Después de la selección, el servidor web vacía el contenido del archivo de registro sin borrarlo. No es posible cancelar el borrado del registro de datos desde el cuadro de diálogo Guardar archivo. Solo es posible cancelar la operación desde la confirmación inicial del cuadro de diálogo Recuperar y borrar.

Si se produce un error durante el proceso de recuperación y borrado, no se borran datos del registro y se muestra un mensaje de error. Si el registro de datos tiene un "No" en la columna "Activo", la operación de recuperación y borrado no borra el contenido porque el programa STEP 7 no tiene ninguna referencia al archivo. En este caso hay que borrar manualmente el archivo para eliminar los datos.

Consulte también

Servidor web (Página 843)

12.7.14 Archivos de usuario

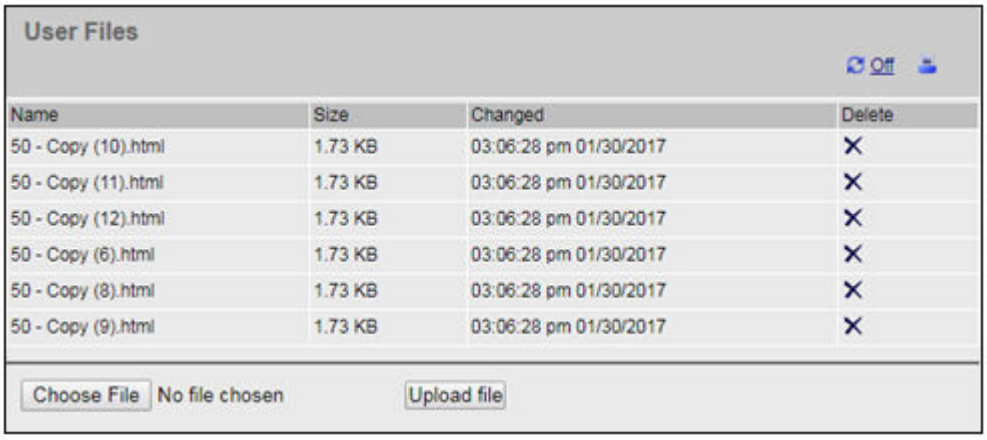
Manejo de archivos de usuario

La página Archivos de usuario permite acceder a archivos ubicados en la Memory Card (memoria de carga externa).

El tipo de acceso al archivo de usuario depende de los privilegios de usuario. Cualquier usuario que tenga privilegios para "leer archivos" puede ver los archivos y carpetas con la página Archivos de usuario. Si tiene el privilegio para "escribir/borrar archivos", también podrá:

- Ver una lista de todos los archivos de usuario del PLC
- Descargar un archivo de usuario del PLC al equipo
- Cargar un archivo de usuario del equipo al PLC
- Borrar un archivo de usuario del PLC

Para ver los archivos de usuario, haga clic en Archivos de usuario desde la página de navegación principal.



La lista de archivos incluye el tamaño actual del archivo y la fecha de la última modificación. La lista de archivos se llena desde el directorio "Archivos de usuario" en la raíz de la SMC.

Nota

Administración de archivos de usuario

No guarde más de 1000 archivos de usuario en la SMC. Si hay más archivos, el servidor web solo permitirá ver los 1000 primeros archivos. Los archivos se muestran en orden alfabético ascendente, sin tener en cuenta las mayúsculas y minúsculas.

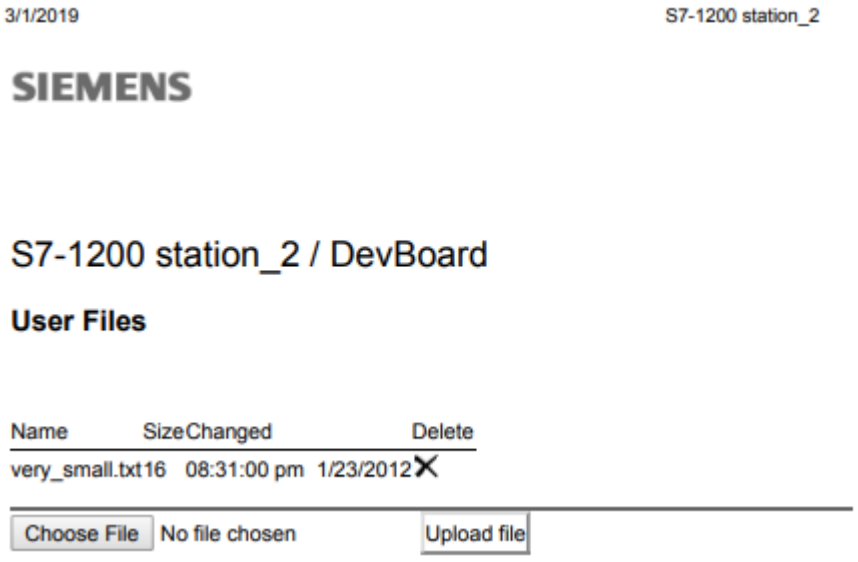
Paginación de la lista de archivos de usuario

La lista de archivos de usuario se pagina en incrementos de 50 elementos. Una casilla desplegable permite seleccionar el rango del elemento que desea ver.



Impresión de la lista de archivos de usuario

La lista de archivos de usuario puede imprimirse en el PLC mediante el icono "Imprimir" de la página web PLC:



Errores de archivos de usuario

La página Archivos de usuario muestra un mensaje de error si una operación no puede finalizar correctamente:



El mensaje de error permanece en la página hasta que se inicia una operación para volver a cargar la página Archivos de usuario. Las operaciones siguientes hacen que el servidor web vuelva a cargar la página Archivos de usuario:

- Actualizar la página Archivos de usuario saliendo de ella y volviendo a ella
- Modificar la selección de paginación actual de los archivos de usuario
- Borrar correctamente un archivo de usuario
- Actualizar correctamente un archivo de usuario

Nota

Mensaje de error en archivos de usuario

La funcionalidad de actualización automática del servidor web no elimina un mensaje de error de la página Archivos de usuario.

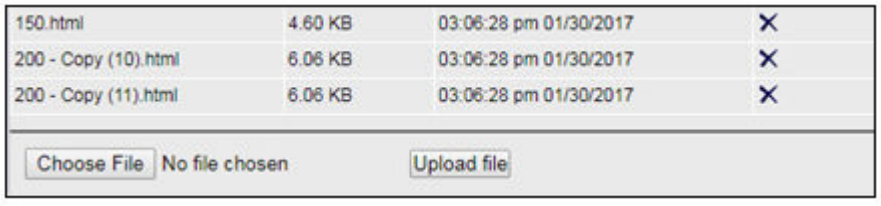
Los escenarios siguientes generan mensajes de error en la página Archivos de usuario:

Operación	Condición de fallo	Mensaje de error
Descargar archivo de usuario	• El archivo no existe • Error interno de PLC	Error al descargar el archivo
Descargar archivo de usuario Cargar archivo de usuario Borrar archivo de usuario	• Permisos insuficientes para la operación solicitada • Método de solicitud HTTP o parámetro de acción no válido	Operación de archivo no permitida
Descargar archivo de usuario Cargar archivo de usuario Borrar archivo de usuario	• Campo de referente HTTP no válido o ausente	Operación de archivo no permitida - no hay referente
Borrar archivo de usuario	• La Memory Card está protegida contra escritura	Operación de archivo no permitida
Descargar Borrar Recuperar y borrar	• El archivo no existe • Error interno de PLC	Error al borrar el archivo - la Memory Card está protegida contra escritura

Operación	Condición de fallo	Mensaje de error
Cargar archivo de usuario	• El nombre de archivo no es válido o está ausente • Error interno de PLC	Error al cargar el archivo
Cargar archivo de usuario	• El nombre de archivo ya existe en el PLC	Error al cargar el archivo - el nombre ya existe
Cargar archivo de usuario	• La SMC está llena	Error al cargar el archivo - la Memory Card está llena
Cargar archivo de usuario	• Nombre de archivo no válido	Error al cargar el archivo - carácter no válido en el nombre de archivo
Cargar archivo de usuario	• El archivo es demasiado grande para el sistema de archivos del SM	Error al cargar el archivo - el archivo es demasiado grande
Cargar archivo de usuario	• La SMC está protegida contra escritura	Error al cargar el archivo - la Memory Card está protegida contra escritura

Carga de un archivo de usuario

Existe la posibilidad de cargar un archivo de usuario seleccionándolo en el formulario de carga de archivos de usuario:



Para cargar un archivo de usuario, seleccione el archivo y haga clic en "Cargar archivo" para empezar a transferir el archivo del equipo al PLC. Antes de que la transferencia comience, deben cumplirse los siguientes requisitos:

Requisitos	Mensaje
Debe haber un archivo seleccionado	Introduzca un nombre de archivo
El nombre de archivo debe ser válido	Los caracteres permitidos son a-z, 0-9, (){}[]$!=~ (espacio)
El archivo debe tener menos de 2 GB de tamaño (límite del sistema de archivos)	Error al cargar el archivo - el archivo es demasiado grande

Consulte también

Servidor web (Página 843)

12.7.15 Data Log UserFiles API

Data Log UserFiles API

Para archivos de usuario y registros de datos del S7-1200 hay disponible una función API denominada Data Log User Files. Encontrará más información en el *manual S7-1500, ET 200SP, ET 200pro Servidor web*.

12.7.16 File Browser

La página Navegador de archivos permite acceder a archivos en la memoria de carga interna de la CPU o en la Memory Card (memoria de carga externa). La página del navegador de archivos muestra la carpeta raíz de la memoria de carga, que contiene las carpetas "Recetas" y "Registros de datos", pero también muestra cualquier otra carpeta que haya podido crear, si utiliza una Memory Card.

El tipo de acceso a archivos que tiene para archivos y carpetas depende de sus privilegios (Página 848) de usuario. Cualquier usuario que tenga privilegios para "leer archivos" puede ver los archivos y carpetas con el navegador de archivos. No es posible borrar la carpeta Recetas o Registros de datos, sean cuales sean sus privilegios de inicio de sesión, pero si se han creado carpetas personalizadas en la Memory Card, podrán borrarse si se ha iniciado sesión como usuario con privilegios para "escribir/borrar archivos".

Haga clic en una carpeta para acceder a los archivos específicos de la carpeta.



Archivos de recetas

La carpeta de recetas muestra todos los archivos de recetas que existen en la memoria de carga. Los archivos de recetas también están en formato CSV y puede abrirlos en Microsoft Excel o cualquier otro programa. Es preciso modificar los privilegios para borrar, modificar y guardar, renombrar o cargar los archivos de recetas.

Actualización de archivos y actualización automática de páginas

Si comienza a cargar un archivo, el proceso de carga continuará mientras se mantenga en la página del navegador de archivos. Si habilita la actualización automática para actualizar las páginas del servidor web cada diez segundos, cada vez que se actualiza la página se visualiza el progreso incremental del proceso de carga del archivo. Por ejemplo, si está cargando un archivo de 2 MB, se pueden ver actualizaciones que muestren el tamaño del archivo en bytes a 2500, 5000, 10000, 15000 y 20000 según va progresando el proceso.

Si abandona la página del navegador de archivos antes de finalizar el proceso de carga, el servidor web borrará el archivo incompleto.

Información adicional

Nota

Convenciones para nombres de archivo

Para que el servidor web funcione con archivos de registro y de receta, los caracteres de los nombres de archivo deben pertenecer al juego de caracteres ASCII, exceptuando los caracteres \ / : * ? " < > | y el espacio.

Si los nombres no cumplen las convenciones de nombre, el servidor web puede mostrar errores en operaciones como la carga o eliminación de archivos, así como al cambiar el nombre de archivos. En ese caso puede ser necesario utilizar un lector de tarjetas y el Explorador de Windows para cambiar el nombre de los archivos que se encontraban en la memoria de carga externa.

Para obtener información sobre la programación con instrucciones de registros e importación (Página 485) y exportación (Página 483) de recetas, consulte el capítulo Recetas y registros de datos (Página 478).

Consulte también

Importar registros de datos en formato CSV a versiones de Microsoft Excel que no son estadounidenses ni británicas (Página 935)

12.8 Páginas web definidas por el usuario

El servidor web de S7-1200 también ofrece medios para crear páginas HTML específicas para la aplicación que incorporan datos del PLC.

⚠ ADVERTENCIA

Acceso no autorizado a la CPU a través de páginas web definidas por el usuario

El acceso no autorizado a la CPU a través de páginas web definidas por el usuario puede afectar al funcionamiento del proceso y puede producir la muerte, lesiones corporales graves o daños materiales.

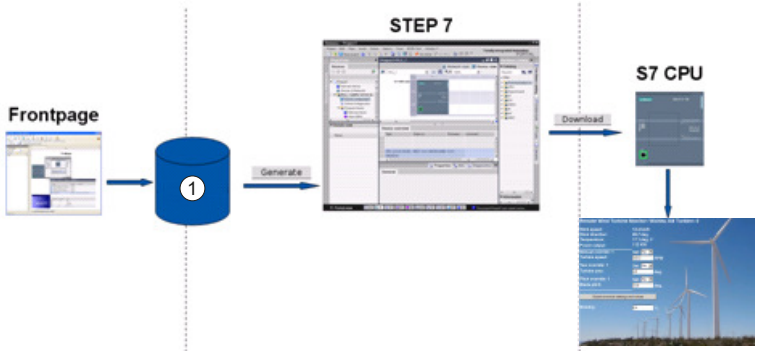
La codificación no segura de las páginas web definidas por el usuario incorpora vulnerabilidades de seguridad, como "cross-site scripting" (XSS), inyección de código y otras.

Proteja la CPU S7-1200 de los accesos no autorizados instalándola de forma segura, tal y como se indica en las Guías operacionales que encontrará en la página web sobre seguridad industrial (http://www.siemens.com/industrialsecurity).

Las páginas web definidas por el usuario se crean utilizando el editor de HTML deseado y descargándolas en la CPU en la que serán accesibles desde el menú de la página web estándar. Este proceso incluye varias tareas:

- Crear páginas HTML con un editor de HTML como Microsoft Frontpage. (Página 892)
- Incluir comandos AWP en comentarios HTML con el código HTML (Página 893): los comandos AWP son un conjunto fijo de comandos que suministra Siemens para acceder a la información de la CPU.
- Configurar STEP 7 para leer y procesar las páginas HTML. (Página 907)
- Generar bloques desde las páginas HTML (Página 907)
- Programar STEP 7 para controlar el uso de las páginas HTML (Página 909)
- Compilar y cargar los bloques en la CPU (Página 910)
- Acceder a las páginas web definidas por el usuario desde el PC (Página 911)

Este proceso se describe a continuación:



① Archivos HTML con comandos AWP incrustados

Consulte también

Servidor web (Página 843)

12.8.1 Creación de páginas HTML

Puede usar el paquete de software que desee para crear sus propias páginas HTML y utilizarlas con el servidor web. Asegúrese de que el código HTML que utilice cumple los estándares de HTML de W3C (World Wide Web Consortium). STEP 7 no realiza verificación alguna de la sintaxis HTML.

Se puede usar un paquete de software que permita diseñar en WYSIWYG o en modo de esquema de trazado, pero tiene que ser posible editar el código HTML en HTML puro. La mayor parte de las herramientas de diseño web ofrece este tipo de edición; si no es así, siempre se puede usar un simple editor de textos para editar el código HTML. Incluya la línea siguiente en su página HTML para configurar la fuente para la página a UTF-8:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Asegúrese también de guardar el archivo desde el editor con codificación de caracteres UTF-8.

Utilice STEP 7 para compilar todo en las páginas HTML en bloques de datos de STEP 7. Estos bloques de datos constan de un bloque de datos de control que controla la visualización de las páginas web, así como de uno o varios bloques de datos de fragmentos que contienen las páginas web compiladas. Hay que tener en cuenta que los conjuntos extendidos de páginas HTML, en especial los que contienen una gran cantidad de imágenes, requieren mucho espacio de memoria de carga (Página 911) para los DB de fragmentos. Si la memoria de carga interna de la CPU no es suficiente para las páginas web definidas por el usuario, se puede utilizar una Memory Card (Página 119) para ofrecer memoria de carga externa.

Para programar el código HTML con el objetivo de usar datos del S7-1200, hay que incluir comandos AWP (Página 893) como comentarios HTML. Al acabar, guarde las páginas HTML en el PC y anote la ruta de carpetas en la que se han guardado.

Nota

El tamaño máximo para archivos HTML que contengan comandos AWP es de 64 KB. Para que STEP 7 pueda compilar sus páginas correctamente debe mantener el tamaño de los archivos por debajo de dicho límite.

Siemens recomienda que todos los archivos de recursos web (archivos .ccc, imágenes, archivos JavaScript y archivos html) se creen con un tamaño no superior a 512 KB, ya que de lo contrario pueden aparecer problemas al enviar el archivo del servidor web al navegador. El tamaño del archivo de recursos web en cuestión puede verse en el explorador de archivos del directorio.

Actualizar páginas web definidas por el usuario

Las páginas web definidas por el usuario no se actualizan automáticamente. Es decisión del usuario si desea programar o no el HTML para actualizar la página. En páginas que muestran datos del PLC resulta muy útil actualizar los datos periódicamente. En páginas HTML que sirven para introducir datos, la actualización puede interferir con los datos que introduce el usuario. Si desea que la página entera se actualice automáticamente, es posible agregar esta línea al encabezado HTML, siendo "10" el número de segundos entre actualizaciones:

```
<meta http-equiv="Refresh" content="10">
```

También se puede utilizar JavaScript u otras técnicas HTML para controlar la actualización de la página o de los datos. Para tal fin, consulte la documentación acerca de HTML y JavaScript.

12.8.2 Comandos AWP soportados por el servidor web del S7-1200

El servidor web del S7-1200 ofrece comandos AWP que se incrustan en las páginas web definidas por el usuario como comentarios HTML para los fines siguientes:

- Leer variables (Página 894)
- Escribir variables (Página 895)
- Leer variables especiales (Página 897)
- Escribir variables especiales (Página 899)
- Definir tipos de enumeración (Página 901)
- Asignar variables a tipos de enumeración (Página 901)
- Crear bloques de datos de fragmentos (Página 903)

Sintaxis general

A excepción del comando para leer una variable, los comandos AWP tienen la sintaxis siguiente:

```
<!-- AWP_ <command name and parameters> -->
```

Utilice los comandos AWP en combinación con comandos típicos de formas HTML para escribir en variables de la CPU.

Las descripciones de los comandos AWP en las páginas siguientes utilizan las convenciones siguientes:

- Los elementos encerrados en corchetes [ ] son opcionales.
- Los elementos entre corchetes angulares < > son valores de parámetros que deben especificarse.
- Las comillas son parte literal del comando. Deben estar presentes tal y como se indica.
- Los caracteres especiales en nombres de variables o de bloques de datos, dependiendo de su uso, deben escaparse o encerrarse entre comillas (Página 905).

Utilice un editor de textos o un modo de edición HTML para insertar los comandos AWP en las páginas.

Nota

Sintaxis esperada de los comandos AWP

El espacio después de "<!--" y el espacio antes de "-->" en la formulación de un comando AWP son esenciales para compilar correctamente el comando. La omisión de los caracteres de espacio puede provocar que el compilador no pueda generar el código adecuado. El compilador no muestra un error en este caso.

Resumen de comandos AWP

Los detalles para utilizar cada comando AWP están reunidos en los apartados siguientes, no obstante aquí se muestra un resumen corto de los comandos:

Leer variables
```
:=<Varname>:
```

Escribir variables
```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Este comando AWP sólo declara que la variable en la cláusula de nombre es escribible. El código HTML permite operaciones de escritura en la variable a partir del nombre mediante <input>, <select> u otras instrucciones HTML dentro de una forma HTML.

Leer variables especiales
```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Escribir variables especiales
```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

Definir tipos de enumeración
```
<!--
 AWP_Enum_Def Name='<Enum type name>' Values='<Value>, <Value>,... '
 -->
```

Referenciar tipos de enumeración
```
<!-- AWP_In_Variable Name='<Varname>' Enum="<Enum type name>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<Enum type name>" -->
```

Crear fragmentos
```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>][ID=<id>] -->
```

Importar fragmentos
```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

12.8.2.1 Leer variables

Las páginas web definidas por el usuario pueden leer variables (variables PLC) y variables de bloques de datos de la CPU, siempre que se haya configurado un acceso desde HMI para las variables.

Sintaxis

```
:=<Varname>:
```

Parámetros

<Varname>	La variable que debe leerse; puede ser un nombre de variable PLC del programa STEP 7, una variable de bloque de datos, periferia o memoria direccionable. En las direcciones de memoria o de E/S o en los alias (Página 905) no utilice comillas para el nombre de la variable. Para variables PLC utilice comillas dobles para el nombre de la variable. Para variables de bloque de datos ponga el nombre del bloque solo entre comillas dobles. El nombre de la variable debe estar fuera de las comillas. Observe que debe utilizarse el nombre y no el número del bloque de datos. Elementos de matriz de referencia que utilizan sintaxis de elemento de matriz.

Ejemplos

```
:="Velocidad_cinta":
:="Mi_bloque_de_datos".flag1:
:=I0.0:
:=MW100:
:="Mi_bloque_de_datos".Array_Dim1[0]:
:="Mi_bloque_de_datos".Matriz_Dim2[0,0]:
```

Ejemplo de lectura de una variable con alias

```
<!-- AWP_Out_Variable Name='flag1' Use='"My_Data_Block".flag1' -->
:=flag1:
```

Nota

Encontrará información sobre cómo definir alias para variables PLC y variables de bloque de datos en el apartado Utilizar un alias para una referencia de variable (Página 900).

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado Procesamiento de nombres de variable que contienen caracteres especiales (Página 905).

12.8.2.2 Escribir variables

Las páginas definidas por el usuario pueden escribir datos en la CPU. Esto se realiza utilizando un comando AWP para identificar una variable en la CPU que sea escribible desde la página HTML. La variable debe especificarse por el nombre de variable PLC o el nombre de variable de bloque de datos. Es posible declarar múltiples nombres de variable en una instrucción. Para escribir los datos en la CPU, utilice el comando estándar HTTP POST.

Una aplicación típica es diseñar una forma en la página HTML con campos de entrada de texto u opciones en una lista de selección que se correspondan con variables de CPU escribibles. Como ocurre con todas las páginas definidas por el usuario, a continuación deben generarse los bloques desde STEP 7, como los que están incluidos en el programa de STEP 7. Cuando un usuario con privilegios para modificar variables accede a esta página e introduce datos en los campos de entrada o selecciona una opción de una lista de selección, el servidor web convierte la entrada al tipo de datos correcto para la variable y escribe el valor en la variable de la CPU. Tenga en cuenta que la cláusula de nombre para campos de entrada HTML y listas de selección HTML utiliza una sintaxis típica para la cláusula de nombre del comando AWP_In_Variable . Por

lo general debe ponerse el nombre entre comillas simples y, si se referencia un bloque de datos, el nombre del mismo entre comillas dobles.

Para más detalles sobre la gestión de formas, consulte la documentación de HTML.

Sintaxis

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Parámetros

<Varname1>	Si no está disponible ninguna cláusula de uso, entonces Varname1 es la variable que debe escribirse. Puede tratarse de un nombre de variable PLC del programa STEP 7, de una variable de un bloque de datos específico o de un nombre de bloque de datos. Si está disponible una cláusula de uso, Varname1 es un nombre alternativo para la variable referenciada en <Varname2> (Página 900). Se trata de un nombre local dentro de la página HTML.
<Varname2>	Si está disponible una cláusula de uso, entonces Varname2 es la variable que debe escribirse. Puede tratarse de un nombre de variable PLC del programa STEP 7 o de una variable de un bloque de datos específico.

Tanto para las cláusulas de nombre como para las cláusulas de uso, el nombre entero debe ponerse entre comillas simples. Dentro de las comillas simples, utilice comillas dobles para una variable PLC y para un nombre de bloque de datos. El nombre del bloque de datos está dentro de las comillas dobles pero no el nombre de la variable de bloque de datos. Observe que en el caso de variables de bloque de datos debe utilizarse el nombre y no el número del bloque de datos. Elementos de matriz de referencia que utilizan sintaxis de elemento de matriz.

Si se utiliza un comando AWP_In_Variable para que se pueda escribir en un bloque de datos, será posible escribir en todas las variables del bloque de datos.

Ejemplos utilizando un campo de entrada HTML

```
<!-- AWP_In_Variable Name='"Nivel_destino"' -->
<form method="post">
<p>Nivel de destino de entrada: <input name='"Nivel_destino"'
type="text" />
</p>
</form>

<!-- AWP_In_Variable Name='"Bloque_de_datos_1".Frenado' -->
<form method="post">
<p>Frenado: <input name='"Bloque_de_datos_1".Frenado' type="text" />
%</p>
</form>

<!-- AWP_In_Variable Name='"Bloque_de_datos_1".Matriz_Dim2' -->
<form method="post">
<p>Valor de matriz bidimensional [2,1]: <input
name='"Bloque_de_datos_1".Matriz_Dim2[2,1]' type="text" /> %</p>
</form>
```

Ejemplo de uso de una cláusula de uso

```
<!-- AWP_In_Variable Name='"Frenado"'
Use='"Bloque_de_datos_1".Frenado' -->
<form method="post">
<p>Frenado: <input name='"Frenado"' type="text" /> %</p>
</form>
```

Ejemplo de uso de un bloque de datos escribible

```
<!-- AWP_In_Variable Name='"Bloque_de_datos_1"' -->
<form method="post">
<p>Frenado: <input name='"Bloque_de_datos_1".Frenado' type="text" />
%
</p>
<p>Velocidad de la turbina: <input
name='"Bloque_de_datos_1".VelocidadTurbina' size="10"
value='"Bloque_de_datos_1".VelocidadTurbina' type="text" />
</p>
</form>
```

Ejemplo utilizando una lista de selección HTML

```
<!-- AWP_In_Variable
Name='"Bloque_de_datos_1".HabilitarCorrecciónManual'-->
<form method="post">
<select name='"Bloque_de_datos_1".HabilitarCorrecciónManual'>
<option value=:"Bloque_de_datos_1".HabilitarCorrecciónManual:> </
option>
<option value=1>Sí</option>
<option value=0>No</option>
</select><input type="submit" value="Enviar ajuste" /></form>
```

Nota

Solo un usuario con el privilegio (Página 848) "Escribir en páginas definidas por el usuario" pueden escribir datos en la CPU. El servidor web ignora los comandos si el usuario no tiene privilegios de modificación.

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado "Procesamiento de nombres de variable que contienen caracteres especiales (Página 905)".

12.8.2.3 Leer variables especiales

El servidor web ofrece la posibilidad de leer valores desde el PC para almacenarlos en variables especiales en el encabezado de respuesta HTTP. Es posible, por ejemplo, que se quiera leer un nombre de ruta desde una variable PLC para redireccionar la URL a otra ubicación utilizando la variable especial HEADER:Location.

Sintaxis

```
<!-- AWP_Out_Variable Name='<Type>:<Name>' [Use='<Varname>'] -->
```

Parámetros

<Type>	Se refiere al tipo de variable especial, que puede ser uno de los siguientes: HEADER COOKIE_VALUE COOKIE_EXPIRES
<Name>	En la documentación HTTP encontrará una lista con todos los nombres de variables HEADER. A continuación aparece una lista con algunos ejemplos: Status: código de respuesta Location: ruta para redireccionar Retry-After: tiempo previsto en que el servicio no está disponible para el cliente solicitante Para los tipos COOKIE_VALUE y COOKIE_EXPIRES, <Name> es el nombre de una cookie específica. COOKIE_VALUE:name: valor de la cookie indicada COOKIE_EXPIRES:name: tiempo de vencimiento en segundos de la cookie indicada La cláusula de nombre debe ponerse entre comillas simples o dobles. Si no se ha especificado ninguna cláusula de uso, el nombre de la variable especial corresponde a un nombre de variable PLC. Ponga toda la cláusula de nombre entre comillas simples y la variable PLC entre comillas dobles. El nombre de la variable especial y el de la variable PLC deben coincidir.
<Varname>	Nombre de la variable PLC o variable de bloque de datos en la que debe leerse la variable El Varname debe ponerse entre comillas simples. Dentro de las comillas simples, utilice comillas dobles para una variable PLC o un nombre de bloque de datos. El nombre del bloque de datos está dentro de las comillas dobles pero no el nombre de la variable de bloque de datos. Observe que en el caso de variables de bloque de datos debe utilizarse el nombre y no el número del bloque de datos.

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado Procesamiento de nombres de variable que contienen caracteres especiales (Página 905).

Ejemplo: lectura de una variable especial sin cláusula de uso

```
<!-- AWP_Out_Variable Name='"HEADER:Status"' -->
```

En este ejemplo, la variable especial HTTP "HEADER:Status" recibe el valor de la variable PLC "HEADER:Status". El nombre en la tabla de variables PLC debe coincidir exactamente con el nombre de la variable especial si no está especificada ninguna cláusula de uso.

Ejemplo: lectura de una variable especial con una cláusula de uso

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Estado"' -->
```

En este ejemplo, la variable especial "HEADER:Status" recibe el valor de de la variable PLC "Status".

12.8.2.4 Escribir variables especiales

El servidor web ofrece la posibilidad de escribir valores en la CPU de variables especiales en el encabezado de peticiones HTTP. Así, por ejemplo, se puede almacenar en STEP 7 información acerca de la cookie asociada a una página web definida por el usuario, acerca del usuario que acceda a la página o a la información del encabezado. El servidor web ofrece acceso a variables especiales específicas que se pueden escribir en la CPU, siempre y cuando se haya iniciado sesión como usuario con privilegios para modificar variables.

Sintaxis

```
<!-- AWP_In_Variable Name='<Type>:<Name>' [Use='<Varname>']-->
```

Parámetros

<Type>	Se refiere al tipo de variable especial y puede ser uno de los siguientes: HEADER SERVER COOKIE_VALUE
<Name>	Variable específica de uno de los tipos definidos previamente, como se muestra en estos ejemplos: HEADER:Accept: tipos de contenido que son aceptables HEADER:User-Agent: información acerca del agente usuario que origina la petición. SERVER:current_user_id: id del usuario actual; 0 si ningún usuario ha iniciado sesión SERVER:current_user_name: nombre del usuario actual COOKIE_VALUE:<name>: valor de la cookie indicada Ponga la cláusula de nombre entre comillas simples. Si no se ha especificado ninguna cláusula de uso, el nombre de la variable especial corresponde a un nombre de variable PLC. Ponga toda la cláusula de nombre entre comillas simples y la variable PLC entre comillas dobles. El nombre de la variable especial y el de la variable PLC deben coincidir. En la documentación HTTP encontrará una lista con todos los nombres de variables HEADER.
<Varname>	El nombre de variable en el programa STEP 7 en que desea escribir la variable especial, la cual puede ser un nombre de variable PLC o una variable de bloque de datos. El Varname debe ponerse entre comillas simples. Dentro de las comillas simples, utilice comillas dobles para una variable PLC o un nombre de bloque de datos. El nombre del bloque de datos está dentro de las comillas dobles pero no el nombre de la variable de bloque de datos. Observe que en el caso de variables de bloque de datos debe utilizarse el nombre y no el número del bloque de datos.

Ejemplos

```
<!-- AWP_In_Variable Name='"SERVER:current_user_id"' -->
```

En este ejemplo, la página web escribe el valor de la variable especial HTTP "SERVER:current_user_id" en la variable PLC con el nombre "SERVER:current_user_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id' Use='"my_userid"'
-->
```

En este ejemplo, la página web escribe el valor de la variable especial HTTP "SERVER:current_user_id" en la variable PLC con el nombre "my_userid".

Nota

Solo un usuario con privilegios para modificar variables puede escribir datos en la CPU. El servidor web ignora los comandos si el usuario no tiene privilegios de modificación.

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado "Procesamiento de nombres de variable que contienen caracteres especiales (Página 905)".

12.8.2.5 Utilizar un alias para una referencia de variable

Se puede utilizar un alias en la página web definida por el usuario para una In_Variable o una Out_Variable. Así, por ejemplo, es posible utilizar un nombre simbólico diferente en su página HTML del utilizado en la CPU o bien equiparar una variable en la CPU con una variable especial. La cláusula de uso AWP ofrece esta posibilidad.

Sintaxis

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

Parámetros

<Varname1>	El alias o el nombre de la variable especial Varname1 debe ponerse entre comillas simples o dobles.
<Varname2>	Nombre de la variable PLC a la que desea asignar un alias. La variable puede ser una variable PLC, una variable de bloque de datos o una variable especial. El Varname2 debe ponerse entre comillas simples. Dentro de las comillas simples, utilice comillas dobles para una variable PLC, una variable especial o un nombre de bloque de datos. El nombre del bloque de datos está dentro de las comillas dobles pero no el nombre de la variable de bloque de datos. Observe que en el caso de variables de bloque de datos debe utilizarse el nombre y no el número del bloque de datos.

Ejemplos

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use='"Data_Block_10".server_user' -->
```

En este ejemplo, la variable especial SERVER:current_user_id se escribe en la variable "server_user" del bloque de datos "Data_Block_10".

```
<-- AWP_Out_Variable Name='Weight'
Use='"Data_Block_10".Tank_data.Weight'  -->
```

En este ejemplo, el valor del elemento de estructura del bloque de datos Data_Block_10.Tank_data.Weight puede referenciarse simplemente con "Weight" en el resto de la página web definida por el usuario.

```
<-- AWP_Out_Variable Name='Weight' Use='"Raw_Milk_Tank_Weight"'  -->
```

En este ejemplo, el valor de la variable PLC "Raw_Milk_Tank_Weight" puede referenciarse simplemente con "Weight" en el resto de la página web definida por el usuario.

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado Procesamiento de nombres de variable que contienen caracteres especiales (Página 905).

12.8.2.6 Definir tipos de enumeración

Es posible definir tipos de enumeración en las páginas web definidas por el usuario y asignar los elementos en un comando AWP.

Sintaxis

```
<!-- AWP_Enum_Def Name='<Enum type name>' Values='<Value>,
<Value>,... ' -->
```

Parámetros

<Enum type name>	Nombre del tipo de enumeración, entre comillas simples o dobles.
<Value>	<constant>:<name> La constante indica el valor numérico para la asignación del tipo de enumeración. El número total es ilimitado. El nombre es el valor asignado al elemento de enumeración.

Tenga en cuenta que la cadena completa de valores de enumeración asignados está encerrada entre comillas sencillas y cada elemento individual del tipo de enumeración asignado está encerrado entre comillas dobles. El alcance de una definición de tipo de enumeración es global en las páginas web definidas por el usuario. Si ha configurado las páginas web definidas por el usuario en carpetas de lenguaje (Página 924), la definición del tipo de enumeración es global para todas las páginas de la carpeta de lenguaje.

Ejemplo

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"No alarms", 1:"Tank is
full", 2:"Tank is empty"' -->
```

12.8.2.7 Referenciar variables de CPU con un tipo de enumeración

Es posible asignar una variable en la CPU a un tipo de enumeración. Esta variable se puede utilizar en otro lugar de la página web definida por el usuario en una operación de lectura (Página 894) o una operación de escritura (Página 895). En una operación de lectura, el servidor web reemplaza el valor numérico leído desde la CPU por el valor de texto de enumeración correspondiente. En una operación de escritura, el servidor web reemplaza el valor de texto por el valor entero de la enumeración correspondiente al texto antes de escribir el valor en la CPU.

Sintaxis

```
<!-- AWP_In_Variable Name='<Varname>' Enum="<EnumType>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<EnumType>" -->
```

Parámetros

<Varname>	Nombre de la variable de PLC o de la variable de bloque de datos que se desea asociar con el tipo de enumeración o el alias para una variable de PLC (Página 900), de estar declarado. El Varname debe ponerse entre comillas simples. Dentro de las comillas simples, utilice comillas dobles para una variable PLC o un nombre de bloque de datos. Observe que en el caso de variables de bloque de datos debe utilizarse el nombre y no el número del bloque de datos. El nombre del bloque de datos está dentro de las comillas dobles pero no el nombre de la variable de bloque de datos.
<EnumType>	Nombre del tipo de enumeración que debe ponerse entre comillas simples o dobles

El alcance de una referencia de tipo de enumeración es el fragmento actual.

Ejemplo de utilización en una lectura de variable

```
<!-- AWP_Out_Variable Name='"Alarm"' Enum="AlarmEnum" -->...
<p>The current value of "Alarm" is :="Alarm":</p>
```

Si el valor de "Alarm" en la CPU es 2, la página HTML visualiza 'The current value of "Alarm" is Tank is empty' debido a que la definición del tipo de enumeración (Página 901) asigna la cadena de texto "Tank is empty" al valor numérico 2.

Ejemplo de utilización en una escritura de variable

```
<!-- 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>
```

Dado que la definición del tipo de enumeración (Página 901) asigna "Tank is full" al valor numérico 1, el valor 1 se escribe en la variable de PLC denominada "Alarm" en la CPU.

Tenga en cuenta que la cláusula de enumeración en la declaración AWP_In_Variable debe corresponderse exactamente con la cláusula de nombre en la declaración AWP_Enum_Def .

Ejemplo de utilización en una escritura de variable aplicando un alias

```
<!-- 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>
```

Dado que la definición del tipo de enumeración (Página 901) asigna "Tank is full" al valor numérico 1, el valor 1 se escribe en el alias "Alarm" que se corresponde con la variable PLC denominada "Motor1.Alarm" en el bloque de datos "Data_Block_4" de la CPU.

Si el nombre de la variable o del bloque de datos incluye caracteres especiales hay que utilizar comillas adicionales o caracteres de escape, tal como se describe en el apartado Procesamiento de nombres de variable que contienen caracteres especiales (Página 905).

Nota

En versiones anteriores se necesitaba una declaración AWP_Enum_Ref aparte para asociar una variable con un tipo de enumeración definido. STEP 7 y S7-1200 admiten el código existente con declaraciones with AWP_Enum_Ref; no obstante, este comando ya no es necesario.

12.8.2.8 Crear fragmentos

STEP 7 convierte y almacena páginas web definidas por el usuario como un DB de control y DBs de fragmentos al hacer clic en "Generar bloques" en las Propiedades de la CPU para el servidor web. Se pueden configurar fragmentos específicos para páginas específicas o para secciones de páginas específicas. Estos fragmentos se pueden identificar a través de un nombre y un número con el comando AWP "Start_Fragment". Todos los elementos de la página que siguen al comando AWP_Start_Fragment pertenecen a ese fragmento hasta que se lance otro AWP_Start_Command o hasta que se alcance el final del archivo.

Sintaxis

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Type>][ID=<id>]
[Mode=<Mode>] -->
```

Parámetros

<Name>	Cadena de texto: nombre del DB de fragmentos Los nombres de fragmentos deben empezar con una letra o un guión bajo y estar formados por letras, números y guiones bajos. El nombre del fragmento es una expresión regular con la forma: [a-zA-Z_][a-zA-Z_0-9]*
<Type>	"manual" o "automático" manual: El programa de STEP 7 debe solicitar este fragmento y puede responder en correspondencia. El funcionamiento del fragmento se debe controlar con STEP 7 y con las variables del DB de control. automático: El servidor web procesa el fragmento automáticamente. Si no se especifica el tipo de parámetro, el ajuste predeterminado es "automático".

<id>	Número de identificación entero. Si no se especifica el parámetro ID, el servidor web asigna un número predeterminado. Para fragmentos manuales, ajuste un número reducido para la ID. La ID es el medio mediante el cual el programa de STEP 7 controla un fragmento manual.
<Mode>	"visible" u "oculto" visible: Los contenidos del fragmento se mostrarán en la página web definida por el usuario. oculto: Los contenidos del fragmento no se mostrarán en la página web definida por el usuario. Si no se especifica el tipo de parámetro, el ajuste predeterminado es "visible".

Fragmentos manuales

Si se crea un fragmento manual para una página web definida por el usuario o para una parte de una página, el programa de STEP 7 debe controlar el momento en que se envía el fragmento. El programa de STEP 7 debe configurar los parámetros adecuados en el DB de control para una página definida por el usuario mediante control manual y luego debe llamar la instrucción WWW con el DB de control según se modifique. Consulte el apartado Control avanzado de páginas web definidas por el usuario (Página 927) para comprender la estructura del DB de control y aprender a manejar las páginas individuales y los fragmentos.

12.8.2.9 Importar fragmentos

Es posible crear un fragmento determinado a partir de una parte del código HTML y luego importar este fragmento a otro lugar del grupo de páginas web definidas por el usuario. Por ejemplo, tome un grupo de páginas web definidas por el usuario que tenga una página de inicio y luego varias páginas HTML accesibles a través de los enlaces de la página de inicio. Supongamos que cada página individual debe visualizar el logotipo de la empresa en la página. Esto se puede realizar creando un fragmento (Página 903) que cargue la imagen del logotipo de la empresa. A continuación, cada página HTML puede importar este fragmento para visualizar el logotipo de la empresa. Para este fin, utilice el comando AWP Import_Fragment. El código HTML del fragmento solo existe en un fragmento, no obstante este DB de fragmentos se puede importar tantas veces como se requiera en cualquier número de páginas web de su elección.

Sintaxis

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

Parámetros

<Name>	Cadena de texto: nombre del DB de fragmentos que se debe importar

Ejemplo

Extracto del código HTML que crea un fragmento para visualizar una imagen:

```
<!-- AWP_Start_Fragment Name='My_company_logo' --><p><img
src="company_logo.jpg"></p>
```

Extracto del código HTML en otro archivo .html que importa el fragmento que visualiza la imagen del logotipo:

```
<!-- AWP_Import_Fragment Name='My_company_logo' -->
```

Ambos archivos .html (el que crea el fragmento y el que lo importa) se encuentran en la estructura de carpetas que se define al configurar las páginas definidas por el usuario en STEP 7 (Página 907).

12.8.2.10 Combinar definiciones

Al declarar variables para usarlas en las páginas web definidas por el usuario se puede combinar una declaración de variable y un alias para la variable (Página 900). Asimismo, se pueden declarar varias In_Variables en una instrucción y varias Out_Variables en una instrucción.

Ejemplos

```
<!-- AWP_In_Variable Name='"Level"' Name='"Weight"' 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 Procesamiento de nombres de variable que contienen caracteres especiales

Al especificar nombres de variables en páginas web definidas por el usuario, hay que observar cuidadosamente si los nombres de variable contienen caracteres con significados especiales.

Leer variables

Para leer una variable (Página 894) se utiliza la sintaxis siguiente:

```
:=<Varname>:
```

Las reglas siguientes rigen para leer variables:

- Para nombres de variables de la tabla de variables PLC, ponga el nombre de la variable entre comillas dobles.
- Para nombres de variables de bloque de datos,ponga el nombre del bloque de datos entre comillas dobles. La variable debe quedar por fuera de las comillas.
- Para nombres de variables que son direcciones E/S directas, direcciones de memoria o alias no utilice comillas para la variable de lectura.
- Para nombres de variables o variables de bloque de datos que contienen una barra inversa, ponga otra barra inversa delante de la primera.
- Si un nombre de variable o variable de bloque de datos contiene dos puntos, el signo menos, el signo más o un signo &, defina un alias que no contenga caracteres especiales para la variable de lectura y lea la variable utilizando el alias. Ponga una barra inversa delante de dos puntos en nombres de variables dentro de una cláusula de uso.

Tabla 12-1 Ejemplos de lectura de variables

Nombre del bloque de datos	Nombre de la variable	Comando de lectura
n/a	ABC:DEF	`<!--AWP_Out_Variable Name='special_tag' Use ='"ABC:DEF"' --> :=special_tag:`
n/a	T\	`:="T\\":`
n/a	A \B 'C :D	`<!--AWP_Out_Variable Name='another_special_tag' Use='"A \\B \'C :D"' --> :=another_special_tag:`
n/a	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:`

Cláusulas de nombre y uso

Los comandos AWP_In_Variable, AWP_Out_Variable, AWP_Enum_Def, AWP_Enum_Ref, AWP_Start_Fragment y AWP_Import_Fragment tienen cláusulas de nombre. Los comandos de formas HTML como <input> y <select> también tienen cláusulas de nombre. AWP_In_Variable y AWP_Out_Variable pueden tener adicionalmente cláusulas de uso. Indistintamente del comando, la sintaxis para las cláusulas de nombre y uso es igual en lo referente al procesamiento de caracteres especiales.

- El texto que se suministra para una cláusula de nombre o uso debe ponerse entre comillas simples. Si el nombre entre comillas es un nombre de variable PLC o de bloque de datos, utilice comillas simples para toda la cláusula.
- Dentro de una cláusula de nombre o uso, los nombres de bloque de datos y de variables PLC deben ponerse entre comillas dobles.
- Si un nombre de variable o de bloque de datos incluye un carácter de comilla simple o una barra inversa, omita dicho carácter con una barra inversa. La barra inversa es el carácter "Escape" en el compilador de comandos AWP.

Tabla 12-2 Ejemplos de cláusulas de nombre

Nombre del bloque de datos	Nombre de la variable	Opciones de la cláusula de nombre
n/a	ABC'DEF	`Name='"ABC\'DEF"'`
n/a	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'`

Nombre del bloque de datos	Nombre de la variable	Opciones de la cláusula de nombre
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'`

Las cláusulas de uso se rigen por las mismas convenciones que las cláusulas de nombre.

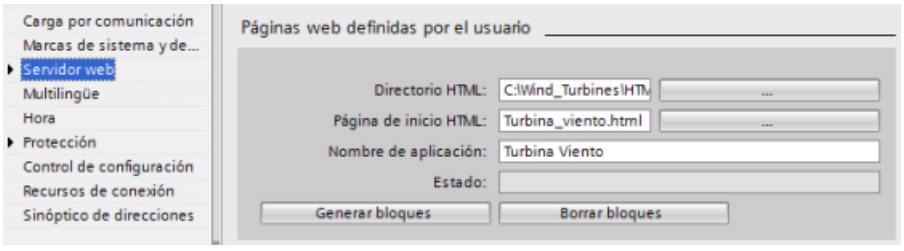
Nota

Indistintamente de qué caracteres se utilicen en la página HTML, configure la fuente de la página HTML a UTF-8 y guárdela desde el editor con codificación de caracteres UTF-8.

12.8.3 Configurar el uso de las páginas web definidas por el usuario

Para configurar las páginas web definidas por el usuario desde STEP 7, proceda del siguiente modo:

1. Seleccione la CPU en la vista de la configuración de dispositivos.
2. Visualice las propiedades del "Web server" en la ventana de inspección de la CPU.
3. Seleccione la casilla de verificación para "Activar servidor web en el módulo" si no está seleccionada.
4. Seleccione "Permitir acceso solo vía HTTPS" para asegurarse de que el servidor web utiliza comunicación cifrada y para aumentar la seguridad de la CPU accesible mediante web.
5. Introduzca o navegue hasta el nombre de la carpeta en el PC donde guardó la página HTML predeterminada (página de arranque).
6. Introduzca el nombre de la página predeterminada.
7. Asigne un nombre a la aplicación (opcional). El servidor web utiliza el nombre de la aplicación para subcategorizar o agrupar con más detalle las páginas web. Cuando se proporciona un nombre de aplicación, el servidor web crea una URL para la página definida por el usuario con el formato siguiente: http[s]://ww.xx.yy.zz/awp/<nombre aplicación>/<nombre página>.html. Si no introduce un nombre de aplicación, la URL es http[s]://ww.xx.yy.zz/awp/<nombre de la página>.html.
   Evite los caracteres especiales en el nombre de la aplicación. Algunos caracteres pueden hacer que el servidor web no pueda mostrar las páginas definidas por el usuario.



8. En el área Avanzado, introduzca extensiones del nombre para archivos que incluyan comandos AWP. De forma predeterminada, STEP 7 analiza archivos con extensiones .htm, .html o .js. Si dispone de otras extensiones de archivos, añádalas. Para ahorrar recursos de procesamiento, no introduzca extensiones si ningún archivo de ese tipo incluye comandos AWP.

9. Conserve el número de DB predeterminado de la web o introduzca un número de su elección. Este es el número del DB de control que controla la visualización de las páginas web.
10. Conserve el número inicial predeterminado del DB de fragmentos o introduzca un número de su elección. Este es el primer DB de fragmentos que contiene las páginas web.

Generar bloques de programa

Cuando se hace clic en el botón "Generar bloques", STEP 7 genera bloques de datos a partir de las páginas HTML del directorio fuente HTML especificado y un bloque de datos de control para operar las páginas web. Es posible configurar estos atributos para su aplicación según los requerimientos específicos (Página 909). STEP 7 genera también un conjunto de bloques de datos de fragmentos para garantizar la representación de todas las páginas HTML. Al generar los bloques de datos, STEP 7 actualiza las propiedades para visualizar el número del bloque de datos de control y el número del primer bloque de datos de fragmentos. Tras generar los bloques de datos, las páginas web definidas por el usuario forman parte del programa de STEP 7. Los bloques que pertenecen a estas páginas aparecen en una carpeta del servidor web, que se encuentra en los bloques de programa del árbol de navegación del proyecto, en la carpeta de bloques de programa.

Borrar bloques de programa

Para borrar bloques de datos generados previamente, haga clic en el botón "Borrar bloques de datos". STEP 7 borra el bloque de datos de control y todos los bloques de datos de fragmentos del proyecto asociados a páginas web definidas por el usuario.

12.8.4 Configurar la página de entrada

En la configuración de dispositivos de la CPU existe la posibilidad de establecer una página web definida por el usuario como página de entrada para acceder al servidor web desde un PC o un dispositivo móvil. En caso de no hacerse, la página de entrada es la página web estándar Introducción (Página 862).

Para establecer que una de las páginas web definidas por el usuario sea la página de entrada, proceda del siguiente modo:

1. Seleccione la CPU en la vista de configuración de dispositivos.
2. En la ventana de inspección, seleccione "Servidor web" desde las propiedades de la CPU y habilite (Página 846) el servidor web.
3. Seleccione "Página de entrada" en las propiedades del servidor web.
4. Seleccione "UP1" en la lista desplegable para configurar el servidor web de modo que muestre una página definida por el usuario al acceder. (La otra selección, "Página de introducción", ajusta el servidor web para que muestre la página web estándar de introducción al acceder.)

También hay que configurar el usuario Todos para que tenga el derecho (Página 848) "Abrir páginas web definidas por el usuario" e incluir una llamada de la instrucción WWW (Página 909) en el programa.

Una vez haya finalizado la configuración y se haya descargado el proyecto en la CPU, el servidor web puede utilizar como página de entrada la "página HTML predeterminada" que se seleccionó al configurar las páginas web definidas por el usuario (Página 907).

Nota

La CPU debe estar en RUN para visualizar una página de entrada definida por el usuario.

12.8.5 Programar la instrucción WWW para páginas web definidas por el usuario

El programa de usuario de STEP 7 debe incluir y ejecutar instrucciones WWW para que las páginas web definidas por el usuario sean accesibles desde las páginas web estándar. El bloque de datos de control es el parámetro de entrada de la instrucción WWW y especifica el contenido de las páginas, según se representan en los bloques de datos de fragmentos, así como la información de estado y control. STEP 7 crea el bloque de datos de control al hacer clic en el botón "Crear bloques" en la configuración de las páginas web definidas por el usuario (Página 907).

Programar la instrucción WWW

El programa STEP 7 debe ejecutar la instrucción WWW para que las páginas web definidas por el usuario sean accesibles desde las páginas web estándar. También puede resultar conveniente que las páginas web definidas por el usuario solo estén disponibles en determinadas circunstancias que están sujetas a las preferencias y a los requerimientos de la aplicación. En este caso, la lógica del programa puede controlar cuándo se invocará la instrucción WWW.

Tabla 12-3 Instrucción WWW

KOP / FUP	SCL	Descripción
	`ret_val := WWW(` `  ctrl_db:=_uint_in_);`	Permite acceder a las páginas web definidas por el usuario desde las páginas web estándar

Hay que especificar el parámetro de entrada del bloque de datos de control (CTRL_DB) que se corresponda con el número de DB entero del DB de control. Este número de DB de control (denominado número de DB de la web) aparece en las propiedades del servidor web de la CPU tras crear los bloques para las páginas web definidas por el usuario. Introduzca el número entero de DB como parámetro CTRL_DB para la instrucción WWW. El valor de retorno (RET_VAL) contiene el resultado de la función. Observe que la instrucción WWW se ejecuta de modo asíncrono y que la salida RET_VAL puede tener un valor inicial de 0 a pesar de que un error puede ocurrir más tarde. El programa puede comprobar el estado del DB de control para garantizar que

la aplicación se ha iniciado correctamente o bien puede comprobar RET_VAL llamando posteriormente la instrucción WWW.

Tabla 12-4 Valor de retorno

RET_VAL	Descripción
0	No hay error
16#00yx	x: La petición representada por el bit respectivo está en estado de espera: x=1: petición 0 x=2: petición 1 x=4: petición 2 x=8: petición 3 Los valores x pueden ser una operación lógica O para representar los estados de espera de varias peticiones. Si x = 6, por ejemplo, entonces las peticiones 1 y 2 están en espera. y: 0: sin error; 1: existe un error y "last_error" ha sido depositado en el DB de control (Página 927)
16#803a	El DB de control no está cargado.
16#8081	El DB de control es de un tipo, formato o versión incorrecta.
16#80C1	No hay recursos disponibles para inicializar la aplicación web.

Utilización del DB de control

STEP 7 crea el bloque de datos de control al hacer clic en "Generar bloques" y visualiza el número del DB de control en las propiedades de las páginas web definidas por el usuario. El DB de control también aparece en la carpeta de bloques de programa del árbol de navegación del proyecto.

Generalmente, el programa de STEP 7 emplea el DB de control directamente tal y como se creó en el proceso de generación de bloques, sin ninguna manipulación adicional. No obstante, el programa de usuario de STEP 7 puede establecer comandos globales en el DB de control para desactivar el servidor web o para volver a activarlo posteriormente. Asimismo, para las páginas definidas por el usuario creadas como DB de fragmentos manuales (Página 907), el programa de usuario de STEP 7 debe controlar el comportamiento de dichas páginas mediante una tabla de petición en el DB de control. Encontrará más información acerca de estas tareas avanzadas en el apartado Control avanzado de páginas web definidas por el usuario (Página 927).

12.8.6 Cargar los bloques de programa en la CPU

Tras generar los bloques para páginas web definidas por el usuario, éstos pasan a ser parte del programa de STEP 7 como es el caso de otros bloques de programa. Siga el proceso normal para cargar los bloques de programa en la CPU. Tenga en cuenta que solo es posible descargar bloques de programa para páginas web personalizadas cuando la CPU está en STOP.

12.8.7 Acceso a las páginas web definidas por el usuario

A las páginas web definidas por el usuario se accede desde las páginas web estándar (Página 851). Las páginas web estándar disponen de un enlace a las "Páginas definidas por el usuario" en el menú de navegación de la izquierda. La navegación de las páginas básicas también incluye un enlace a las "Páginas definidas por el usuario". Al hacer clic en el enlace "Páginas definidas por el usuario", el navegador web pasa a la página que dispone de un enlace a la página predeterminada. En las páginas definidas por el usuario, la navegación depende de cómo han sido diseñadas las páginas específicas.



Nota

También es posible definir una página web definida por el usuario como página de entrada (Página 908) al servidor web.

12.8.8 Limitaciones específicas de las páginas web definidas por el usuario

Las limitaciones para páginas web estándar (Página 932) también se aplican a las páginas web definidas por el usuario. Además, las páginas web definidas por el usuario presentan algunas características específicas.

Espacio de memoria de carga

Las páginas web definidas por el usuario se convierten en bloques de datos al hacer clic en "Generar bloques". Esta acción requiere mucho espacio en la memoria de carga. Si se ha instalado una Memory Card, se puede contar con la capacidad de dicha Memory Card como espacio de memoria de carga externa para las páginas web definidas por el usuario.

Si no se ha instalado ninguna Memory Card, estos bloques emplean el espacio de la memoria de carga interna, que está limitado según el modelo de CPU.

Es posible comprobar la cantidad de memoria de carga empleada y la cantidad disponible en las herramientas online y de diagnóstico de STEP 7. También se pueden consultar las propiedades de los bloques individuales que genera STEP 7 desde las páginas web definidas por el usuario y comprobar el consumo de memoria de carga.

Nota

Si hay que reducir el espacio necesario para las páginas web definidas por el usuario, se deberá reducir el uso de imágenes, si procede.

Comillas en cadenas de texto

Evite el empleo de cadenas de texto que contengan comillas simples o dobles en las variables de los bloques de datos que se utilizan para cualquier fin en páginas web definidas por el usuario. Puesto que la sintaxis de HTML suele usar comillas simples o dobles como delimitadores, las comillas en las cadenas de texto pueden crear saltos de línea en las páginas web definidas por el usuario.

Para las variables del bloque de datos del tipo String que se usan en páginas web definidas por el usuario, observe las normas siguientes:

- No introduzca comillas simples o dobles en el valor de cadena de la variable del bloque de datos en STEP 7.
- No deje que el programa de usuario haga asignaciones de cadenas que contienen comillas a estas variables de bloque de datos.

12.8.9 Ejemplo de una página web definida por el usuario

12.8.9.1 Página web para vigilar y controlar una turbina de viento

Una página web definida por el usuario podría ser, por ejemplo, una página web utilizada para vigilar y controlar a distancia una turbina de viento:



Descripción

En esta aplicación, cada turbina de un parque de turbinas de viento está equipada con un S7-1200 para ejecutar las funciones de control. En el programa de STEP 7, cada turbina de viento tiene un bloque de datos con información específica de la turbina de viento.

La página web definida por el usuario permite un acceso remoto a la turbina desde un PC. Un usuario puede conectarse con páginas web estándar de la CPU de una turbina de viento en particular y acceder a la página web definida por el usuario "Control remoto de la turbina de viento". Un usuario con privilegios para modificar variables también puede conmutar la turbina a modo manual y controlar las variables para la velocidad, el viraje y el cabeceo de la turbina desde la página web. Un usuario con privilegios para modificar variables también puede ajustar un valor de frenado indistintamente de si la turbina está bajo control manual o automático.

El programa de STEP 7 comprobaría los valores booleanos para corregir el control automático y, de estar ajustada la función correspondiente, aplicaría los valores especificados por el usuario para la velocidad, el viraje y el cabeceo de la turbina. De lo contrario, el programa ignoraría estos valores.

Archivos utilizados

Este ejemplo de página web definida por el usuario consta de tres archivos:

- **Wind_turbine.html**: Esta es la página HTML que implementa la visualización que aparece arriba, utilizando comandos AWP para acceder a los datos del controlador.
- **Wind_turbine.css**: Esta es la hoja de estilo en cascada que contiene los estilos de formato de la página HTML. El uso de una hoja de estilo en cascada es opcional pero puede simplificar el diseño de la página HTML.
- **Wind_turbine.jpg**: Esta es la imagen de fondo utilizada por la página HTML. El uso de imágenes en páginas web definidas por el usuario es, por supuesto, opcional y requiere espacio adicional en la memoria de la CPU.

Estos archivos no se suministran con la instalación, pero se describen a modo de ejemplo.

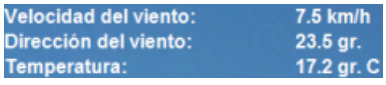
Implementación

La página HTML utiliza comandos AWP para leer valores del PLC (Página 894) para los campos de visualización y escribir valores en el PLC (Página 895) para datos provenientes de las entradas del usuario. Esta página también utiliza comandos AWP para definir tipos de enumeración (Página 901) y referenciar (Página 901) el manejo de ajustes ON/OFF.

Esta primera parte de la página visualiza una línea de encabezado que incluye el número de la turbina de viento.



La siguiente parte de la página muestra las condiciones atmosféricas a las que está sometida la turbina de viento. Las E/S ubicadas en el sitio donde está instalada la turbina suministran la velocidad y la dirección del viento, así como la temperatura actual.



A continuación, la página muestra la potencia de salida de la turbina con base en la lectura del S7-1200.

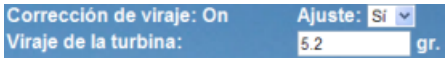


Las siguientes secciones permiten controlar la turbina manualmente, corrigiendo el control automático normal del S7-1200. Están disponibles las siguientes modalidades:

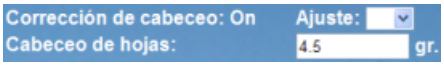
- Corrección manual: activa la corrección manual de la turbina. El programa de usuario de STEP 7 exige que el ajuste de corrección manual sea TRUE (verdadero) antes de habilitar el uso de cualquier ajuste manual para la velocidad, el viraje o el cabeceo de la turbina.



- Corrección de viraje: activa la corrección manual del viraje ajustado y un ajuste manual del mismo. El programa de usuario de STEP 7 exige que tanto la corrección manual como de viraje sean TRUE (verdadero) para aplicar el ajuste de viraje.



- Corrección de cabeceo: activa la corrección manual del cabeceo de las hojas. El programa de usuario de STEP 7 exige que tanto la corrección manual como de cabeceo sean TRUE (verdadero) para aplicar el ajuste de cabeceo de las hojas.



La página HTML dispone de un botón para enviar los ajustes de corrección al controlador.



El campo de entrada de usuario para frenado permite ajustar manualmente un porcentaje de frenado. El programa de usuario de STEP 7 no requiere una corrección manual para aceptar el valor de frenado.



Además, la página HTML utiliza un comando AWP para escribir la variable especial (Página 899) que contiene la ID del usuario que está accediendo a una variable de la lista de variables PLC a través de la página.

12.8.9.2 Leer y visualizar datos del controlador

La página HTML para la vigilancia remota de la turbina de viento utiliza numerosos comandos AWP para leer datos del controlador (Página 894) y visualizarlos en la página. Por ejemplo, tenga en cuenta el código HTML para visualizar la potencia de salida como se indica en esta parte de la página web a modo de ejemplo:



Ejemplo de código HTML

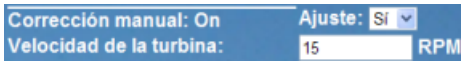
El siguiente fragmento de la página HTML "Control remoto de la turbina de viento" muestra el texto "Potencia de salida:" en la celda izquierda de una fila de la tabla, lee la variable de la potencia de salida y la visualiza en la celda derecha de la fila junto con la abreviatura de kilovatios, kW.

El comando AWP :="Data_block_1".PowerOutput: ejecuta la operación de lectura. Observe que los bloques de datos están referenciados por el nombre y no por el número del bloque de datos (es decir, "Bloque_de_datos_1" y no "DB1").

```
<tr style="height:2%;">
<td>
<p>Potencia de salida:</p>
</td>
<td>
<p style="margin-bottom:5px;"> :="Data_block_1".PotenciaSalida:
kW</p>
</td>
</tr>
```

12.8.9.3 Utilizar un tipo de enumeración

La página HTML "Control remoto de la turbina de viento" utiliza tipos de enumeración para las tres instancias donde la página HTML visualiza "ON" u "OFF" para un valor booleano y donde el usuario especifica un valor booleano. El tipo de enumeración para "ON" resulta en un valor de 1 y el tipo de enumeración para "OFF" resulta en un valor de 0. Por ejemplo, tenga en cuenta el código HTML para leer y escribir el ajuste de habilitación de corrección manual en el valor "Bloque_de_datos_1".HabilitarCorrecciónManual utilizando un tipo de enumeración:



Ejemplo de código HTML

Los fragmentos siguientes de la página HTML "Control remoto de la turbina de viento" muestran cómo declarar un tipo de enumeración denominado "EstadoCorrección" con valores de "Off" y "On" para 0 y 1 respectivamente y, a continuación, el ajuste de una referencia de tipo de enumeración a EstadoCorrección para la variable booleana HabilitarCorrecciónManual en el bloque de datos denominado "Data_block_1".

```
<!-- AWP_In_Variable
Name='"Bloque_de_datos_1".HabilitarCorrecciónManual'
Enum="EstadoCorrección" -->

<!-- AWP_Enum_Def Name="OverrideStatus" Values='0:"Off",1:"On"' -->
```

Donde la página HTML incluye un campo de visualización en una celda de la tabla para el estado actual de HabilitarCorrecciónManual, se utiliza solamente un comando normal de lectura de variables, pero utilizando el tipo de enumeración referenciado y declarado anteriormente, la página muestra "Off" o "On" en lugar de 0 o 1.

```
<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manual
override: :="Bloque_de_datos_1".HabilitarCorrecciónManual:</p>
</td>
```

La página HTML incluye una lista de selección desplegable para que el usuario modifique el valor de HabilitarCorrecciónManual. La lista de selección utiliza el texto "Sí" y "No" para visualizarlo en las listas de selección. Al utilizar el tipo de enumeración, "Sí" se correlaciona con el valor "On" del tipo de enumeración y "No" se correlaciona con el valor "Off". Una selección vacía deja el valor de HabilitarCorrecciónManual inalterado.

```
<select name='"Bloque_de_datos_1".HabilitarCorrecciónManual'>
<option value=':"Bloque_de_datos_1".HabilitarCorrecciónManual:'> </
option>
<option value="On">Sí</option>
<option selected value="Off">No</option>
</select>
```

La lista de selección está incluida dentro de un diálogo en la página HTML. Si el usuario hace clic en el botón de envío, la página envía el diálogo que escribe un valor de "1" en el booleano de HabilitarCorrecciónManual en Bloque_de_datos_1 en caso de que el usuario haya seleccionado "Si", o bien "0" si el usuario ha seleccionado "No".

12.8.9.4 Escribir una entrada del usuario en el controlador

La página HTML de vigilancia remota de la turbina de viento dispone de varios comandos AWP para escribir datos en el controlador (Página 895). La página HTML declara AWP_In_Variables para variables booleanas, de modo que un usuario con privilegios para modificar variables pueda conmutar la turbina de viento a control manual y activar la corrección manual para la velocidad de la turbina, la corrección de viraje o la corrección de cabeceo de hojas. Esta página también utiliza AWP_In_Variables para permitir a un usuario con privilegios para modificar variables especificar posteriormente valores en coma flotante para la velocidad, el viraje y el cabeceo de la turbina, así como para el porcentaje de frenado. La página utiliza un comando de envío de diálogos HTTP para escribir las AWP_In_Variables en el controlador.

Por ejemplo, tenga en cuenta el código HTML para ajustar manualmente el valor de frenado:



Ejemplo de código HTML

El extracto siguiente de la página HTML para la vigilancia remota de la turbina de viento declara primero una AWP_In_Variable para "Bloque_de_datos_1" que permite a la página HTML escribir en cualquier variable del bloque de datos "Bloque_de_datos_1". La página visualiza el texto "Frenado:" en la celda izquierda de una fila de la tabla. En la celda derecha de la fila de la tabla se encuentra el campo que acepta entradas del usuario para la variable "Frenado" de "Bloque_de_datos_1". Este valor de entrada de usuario está dentro de una forma HTML que utiliza el método HTTP "POST" para enviar los datos de texto introducidos a la CPU. A continuación, la página lee el valor de frenado actual del controlador y lo visualiza en el campo de entrada de datos.

Un usuario con privilegios para modificar variables puede utilizar posteriormente esta página para escribir un valor de frenado en el bloque de datos de la CPU que controla el frenado.

```
<!-- AWP_In_Variable Name='"Bloque_de_datos_1"' -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Frenado:</p></td>
<td>
<form method="POST">
<p><input name='"Bloque_de_datos_1".Frenado' size="10" type="text">
%</p>
</form>
</td>
</tr>
```

Nota

Tenga en cuenta que si una página definida por el usuario tiene un campo de entrada de datos para una variable de escritura del bloque de datos del tipo String, el usuario debe poner el string entre comillas simples cuando introduzca el valor del string en el campo.

Nota

Observe que si se declara un bloque de datos entero en una declaración de AWP_In_Variable, por ejemplo <!-- AWP_In_Variable Name="Data_block_1" -->, entonces toda variable dentro de ese bloque de datos se puede escribir desde la página web definida por el usuario. Utilice esta opción si pretende que todas las variables de un bloque de datos sean de escritura. De lo contrario, si desea que solo las variables de bloque de datos específicas se puedan escribir desde la página web definida por el usuario, declárelas específicamente con una declaración, por ejemplo <!-- AWP_In_Variable Name="Data_block_1".Braking' -->

12.8.9.5 Escribir una variable especial

La página web "Control remoto de la turbina de viento" escribe la variable especial SERVER:current_user_id en una variable PLC de la CPU, siempre y cuando ese usuario tenga derechos para modificar variables. En este caso, el valor de la variable PLC contiene la ID del usuario que esté accediendo a la página web para la vigilancia remota de la turbina de viento.

La página web escribe la variable especial en el PLC y no necesita interfaz de usuario.

Ejemplo de código HTML

```
<!-- AWP_In_Variable Name="SERVER:current_user_id"
Use="ID_Usuario"-->
```

12.8.9.6 Referencia: Lista HTML de la página web para la vigilancia remota de la turbina de viento

Turbina_Viento.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--
Este programa de prueba simula una página web para monitorizar y
controlar a distancia una turbina de viento.
Variables PLC y variables de bloque de datos requeridas en STEP 7:

Variable PLC:
ID_usuario: Int
Bloques de datos:
Bloque_de_datos_1
Variables en Bloque_de_datos_1:

NúmeroTurbina: Int
VelocidadViento: Real
DirecciónViento: Real
Temperatura: Real
PotenciaSalida: Real
HabilitarCorrecciónManual: Bool
VelocidadTurbina: Real
CorrecciónViraje: Bool
```

```
Viraje: Real
CorrecciónCabeceo: Bool
Cabeceo: Real
Frenado: Real
La página web definida por el usuario muestra los valores actuales de
los datos PLC y proporciona una lista de selección para establecer
los tres valores booleanos utilizando una asignación de tipo de
enumeración. El botón "Enviar" envía los valores booleanos
seleccionados y los campos de entrada de datos para la velocidad de
la turbina, el viraje y el cabeceo. El valor de frenado se puede
establecer sin utilizar el botón "Enviar".

Para utilizar esta página, no se requiere un programa STEP 7.
Teóricamente, el programa STEP 7 solo actuaría sobre los valores de
velocidad de la turbina, viraje y cabeceo si se hubiesen establecido
los valores booleanos asociados. El único requisito de STEP 7
consiste en llamar a la instrucción WWW con el número de DB de los
bloques de datos generados para esta página.
-->
<!-- AWP_In_Variable Name='"Bloque_de_datos_1"' -->
<!-- AWP_In_Variable
Name='"Bloque_de_datos_1".HabilitarCorrecciónManual'
Enum="EstadoCorrección" -->
<!-- AWP_In_Variable Name='"Bloque_de_datos_1".CorrecciónCabeceo'
Enum="EstadoCorrección" -->
<!-- AWP_In_Variable Name='"Bloque_de_datos_1".CorrecciónViraje'
Enum="EstadoCorrección" -->
<!-- AWP_In_Variable Name="SERVER:current_user_id"
Use="ID_Usuario"-->
<!-- AWP_Enum_Def Name="EstadoCorrección" Values='0:"Off",1:"On"' -->

<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8"><link rel="stylesheet" href="Turbina_Viento.css">
<title>Control remoto de la turbina de viento</title>
</head>
<body>
<table cellpadding="0" cellspacing="2">
<tr style="height: 2%;">
<td colspan="2">
<h2>Control remoto de la turbina de viento: Turbina
#:="Bloque_de_datos_1".NúmeroTurbina:</h2>
</td>

<tr style="height: 2%;"><td style="width: 25%;"><p>Velocidad del
viento:</p></td>
<td><p> :="Bloque_de_datos_1".VelocidadViento: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Dirección del viento:</p></td>
```

```
<td><p> :="Bloque_de_datos_1".DirecciónViento: gr.</p></td>
</tr>

<tr style="height: 2%;"><td style="width: 25%;"><p>Temperatura:</p></
td>
<td><p> :="Bloque_de_datos_1".Temperatura: gr. C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Potencia de salida:</p></td>
<td><p style="margin-
bottom:5px;"> :="Bloque_de_datos_1".PotenciaSalida: 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>Corrección
manual: :="Bloque_de_datos_1".HabilitarCorrecciónManual:</p>
</td>
<td class="Text">Ajuste:

<select name='"Bloque_de_datos_1".HabilitarCorrecciónManual'>
<option value=':="Bloque_de_datos_1".HabilitarCorrecciónManual:'> </
option>
<option value="On">Sí</option>
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
25%;"><p>Velocidad de la turbina:</p></td>
<td>
<p style="margin-bottom:5px;"><input
name='"Bloque_de_datos_1".TurbineSpeed' size="10"
value=':="Bloque_de_datos_1".VelocidadTurbina:' type="text"> RPM</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Corrección de viraje: :="Bloque_de_datos_1".CorrecciónViraje: </p>
</td>
<td class="Text">Ajuste:

<select name='"Bloque_de_datos_1".CorrecciónViraje'>
<option value=':="Bloque_de_datos_1".CorrecciónViraje:'> </option>
<option value="On">Sí</option>
```

```
<option value="Off">No</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Viraje de la turbina:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input
name='"Bloque_de_datos_1".Viraje' size="10"
value=':="Bloque_de_datos_1".Viraje:' type="text"> gr.</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Corrección de cabeceo: :="Bloque_de_datos_1".CorrecciónCabeceo: </
p>
</td>
<td class="Text">Ajuste:

<select name='"Bloque_de_datos_1".CorrecciónCabeceo'>
<option value=':="Bloque_de_datos_1".CorrecciónCabeceo:'> </option>
<option value="On">Sí</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>Cabeceo de hojas:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input
name='"Bloque_de_datos_1".Cabeceo' size="10"
value=':="Bloque_de_datos_1".Cabeceo:' type="text"> gr.</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2">
<input type="submit" value="Enviar ajustes y valores de corrección">
</td>
</tr>
</form>
```

```
<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;"><p>Frenado:</p></td>
<td>
<form method="POST" action="">
<p> <input name='"Bloque_de_datos_1".Frenado' size="10"
value=':="Bloque_de_datos_1".Frenado:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>
```

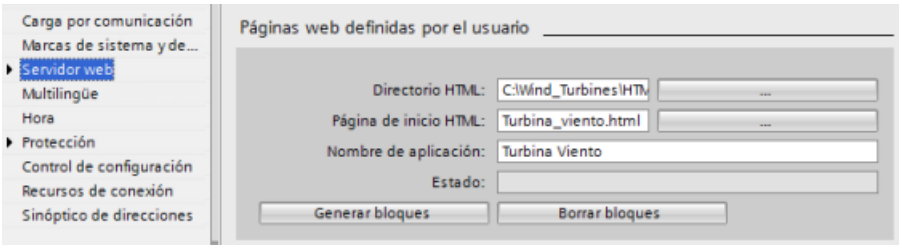
Turbina_Viento.css

```
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 Configuración en STEP 7 de la página web de ejemplo

Para incluir la página HTML "Control remoto de la turbina de viento" como página web definida por el usuario para el S7-1200, hay que configurar los datos acerca de la página HTML en STEP 7 y crear bloques de datos desde la página HTML.

Entre en las propiedades de CPU del S7-1200 que controla la turbina de viento e introduzca la información de configuración en las propiedades de las páginas definidas por el usuario para el servidor web:



Campos de configuración

- Directorio HTML: Este campo especifica el nombre de ruta plenamente habilitado para la carpeta donde está ubicada dentro del PC la página predeterminada (página principal o página de inicio). El botón "..." permite navegar hasta la carpeta requerida.
- Página HTML predeterminada: este campo especifica el nombre de archivo, la página predeterminada o la página de inicio de la aplicación HTML. El botón "..." permite seleccionar el archivo requerido. Para este ejemplo, WindTurbine.html es la página HTML predeterminada. El ejemplo de la vigilancia remota de la turbina de viento consiste solo de una página, pero en otras aplicaciones definidas por el usuario la página predeterminada puede llamar otras páginas a través de enlaces. En el código HTML, la página predeterminada debe referenciar otras páginas en relación con la carpeta de origen HTML.
- Nombre de la aplicación: Este campo opcional contiene el nombre que el navegador web incluye en el campo de dirección cuando visualiza la página. Para este ejemplo, el nombre es "Control remoto de la turbina de viento", pero puede ser cualquier otro.

Ningún otro campo requiere configuración.

Pasos finales

Para utilizar la vigilancia remota de la turbina de viento según se ha configurado, genere los bloques, programe la instrucción WWW (Página 909) con el número del DB generado como parámetro de entrada, cargue los bloques de programa y conmute la CPU a RUN.

Posteriormente, cuando un operador entra en las páginas web estándar del S7-1200 que controla la turbina de viento, puede acceder a la página web "Control remoto de la turbina de viento" desde el enlace "Páginas definidas por el usuario" en la barra de navegación. Esta página ofrece ahora los medios para monitorizar y controlar la turbina de viento.

12.8.10 Configurar páginas web definidas por el usuario en varios idiomas

El servidor web ofrece los medios para diseñar páginas web definidas por el usuario en los idiomas siguientes:

- Alemán (de)
- Inglés (en)
- Español (es)
- Francés (fr)
- Italiano (it)
- Chino simplificado (zh)

Esto se realiza configurando las páginas HTML en una estructura de carpetas (Página 924) que se corresponda con los idiomas y configurando una cookie específica denominada "siemens_automation_language" de las páginas (Página 924). El servidor web responde a esta cookie y cambia a la página predeterminada en la carpeta de idiomas correspondiente.

12.8.10.1 Crear la estructura de carpetas

Para ofrecer páginas web definidas por el usuario en varios idiomas, debe crearse una estructura de carpetas en el directorio HTML. Los nombres de carpeta de dos letras son específicos y deben nombrarse como se indica a continuación:



de: Alemán
en: Inglés
es: Español
fr: Francés
it: Italiano
zh: Chino simplificado

En el mismo nivel, también se pueden incluir otras carpetas que necesite la página, por ejemplo, carpetas para imágenes o scripts.

Es posible incluir cualquier subconjunto de carpetas de idiomas. No es necesario incluir los seis idiomas. Dentro de las carpetas de idiomas, cree y programe las páginas HTML en el idioma correspondiente.

12.8.10.2 Programar el cambio de idioma

El servidor web permite un cambio de idioma mediante el uso de una cookie denominada "siemens_automation_language". Esta es una cookie que está definida e integrada en las páginas HTML y que es interpretada por el servidor web para visualizar el idioma correspondiente a partir de la carpeta de idioma del mismo nombre. La página HTML debe incluir una JavaScript para configurar esta cookie a uno de los identificadores de idioma predefinidos: "de", "en", "es", "fr", "it' o "zh".

Si, por ejemplo, la página HTML configura la cookie en "de", el servidor web cambia a la carpeta "de" y visualiza la página con el nombre predeterminado de la página HTML según se haya definido en la configuración de STEP 7 (Página 927).

Ejemplo

El ejemplo siguiente utiliza una página HTML predeterminada con el nombre "langswitch.html" en cada una de las carpetas de idiomas. En el directorio HTML también hay una carpeta con el nombre "script". La carpeta script incluye un archivo de JavaScript llamado "lang.js". Cada página langswitch.html utiliza esta JavaScript para configurar la cookie de idioma, "siemens_automation_language".

HTML para "langswitch.html" en la carpeta "en"

El encabezado de la página HTML ajusta el idioma a inglés, configura el juego de caracteres a UTF-8 y define la ruta al archivo de JavaScript 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>
```

El cuerpo del archivo utiliza una lista de selección para que el usuario elija entre alemán e inglés. Inglés ("en") está preseleccionado como idioma Cuando el usuario cambia el idioma, la página llama la función de JavaScript DoLocalLanguageChange() con el valor o la opción seleccionada.

```
<!-- 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 para "langswitch.html" en la carpeta "de"

El ancabezado de la página langswitch.html en alemán es el mismo que en inglés, excepto que el idioma ajustado es alemán.

```
<!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>
```

El HTML de la página en alemán es idéntico al de la página en inglés, excepto que el valor predeterminado del idioma seleccionado es alemán ("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" en la carpeta "script"

La función "DoLocalLanguageChange()" se encuentra en el archivo lang.js. Esta función llama la función "SetLangCookie()" y luego vuelve a cargar la ventana que está visualizando la página HTML.

La función "SetLangCookie()" define una asignación que asigna el valor de la lista de selección a la cookie "siemens_automation_language" del documento. Además, se define la ruta para la aplicación, de modo que la página conmutada y no la solicitada recibe el valor de la cookie.

De manera opcional, la página puede ajustar un valor de vencimiento para la cookie en la sección de comentarios.

```
function DoLocalLanguageChange(oSelect) {
        SetLangCookie(oSelect.value);
        top.window.location.reload();
    }
function SetLangCookie(value) {
        var strval = "siemens_automation_language=";
        // Esta es la cookie con la que el servidor web
        // detecta la lengua deseada
        // Este nombre es requerido por el servidor web.
        strval = strval + value;
        strval = strval + "; path=/ ;";
        // Ajustar la ruta de la aplicación, pues en otro caso
        // la ruta se ajustará a la página solicitante
        // y esta página no obtendrá la cookie.
        /* OPTIONAL
           Utilice vencimiento si esta cookie debe existir más tiempo
           que la sesión actual del navegador:
           var now      = new Date();
           var endttime = new Date(now.getTime() + expiration);
           strval = strval + "; expires=" +
                    endttime.toGMTString() + ";";
        */
        document.cookie = strval;
    }
```

Nota

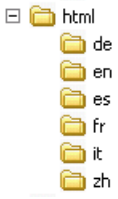
Si la implementación de la página web definida por el usuario incluye archivos HTML que están dentro de carpetas específicas de idioma (en, de, por ejemplo) y también archivos HTML que no están en las carpetas específicas de idioma, tenga en cuenta que no será posible definir tipos de enumeración con el comando AWP_Enum_Def en archivos de ambas ubicaciones. Si se utilizan enumeraciones será necesario definirlas dentro de archivos que están en carpetas específicas de idioma o bien dentro de archivos que están fuera de carpetas específicas de idioma. No es posible hacer declaraciones de enumeraciones en archivos de ambas ubicaciones.

12.8.10.3 Configurar STEP 7 para utilizar una estructura de página multilingüe

El procedimiento para configurar páginas web definidas por el usuario en varios idiomas es similar al proceso general para configurar páginas web definidas por el usuario (Página 907). Si las carpetas se han configurado para idiomas, no obstante se debe configurar el directorio HTML para la carpeta que contiene las carpetas de idiomas individuales. El directorio HTML no se debe configurar de modo que sea una de las carpetas de idiomas.

Al seleccionar la página HTML predeterminada, se navega hasta la carpeta de idiomas y se selecciona la página HTML que va a ser la página de arranque. Si posteriormente se generan bloques y éstos se cargan en la CPU, el servidor web visualiza la página de arranque en la carpeta de idiomas que ha sido configurada.

Por ejemplo, si la estructura de carpetas indicada aquí estuviese en C:\, la configuración del directorio HTML sería C:\html, y si inglés estuviese definido para visualizar la página de arranque, sería necesario navegar hasta en\langswitch.html para acceder a la configuración de la página HTML predeterminada.



12.8.11 Control avanzado de páginas web definidas por el usuario

Al generar bloques de datos para las páginas web definidas por el usuario, STEP 7 crea un DB de control que se utiliza para controlar la visualización de las páginas definidas por el usuario, así como la interacción con estas. STEP 7 crea también un juego de DBs de fragmentos que representa las páginas individuales. En condiciones normales, no es indispensable conocer la estructura del DB de control ni saber cómo manejarlo.

Si desea activar y desactivar una aplicación web o bien manipular fragmentos manuales individuales, utilice para tal fin las variables de DB de control y la instrucción WWW.

Estructura del DB de control

El DB de control es una estructura da datos extensiva que es accesible al programar el programa de usuario de STEP 7. Aquí se describen solo algunas de las variables de DB de control.

Estructura Commandstate

"Commandstate" es una estructura que contiene comandos y estados globales del servidor web.

Comandos globales en la estructura "Commandstate"

Los comandos globales rigen en general para el servidor web. El servidor web se puede activar o bien reiniciar desde los parámetros del DB de control.

Variable de bloque	Tipo de datos	Descripción
init	BOOL	Evalúa el DB de control e inicializa la aplicación web
deactivate	BOOL	Desactiva la aplicación web

Estados globales en la estructura Commandstate

Los estados globales rigen en general para el servidor web y contienen información de estado acerca de la aplicación web.

Variable de bloque	Tipo de datos	Descripción
initializing	BOOL	La aplicación web está leyendo un DB de control
error	BOOL	No ha sido posible inicializar la aplicación web
deactivating	BOOL	La aplicación web está finalizando
deactivated	BOOL	La aplicación ha finalizado
initialized	BOOL	La aplicación web ha sido inicializada
last_error	INT	Último error devuelto por una llamada de la instrucción WWW (Página 909) cuando del código de retorno de WWW es 16#0010: 16#0001: La estructura de DB de fragmento no es coherente. 16#0002: El nombre de la aplicación ya existe. 16#0003: Sin recursos (memoria) 16#0004: La estructura de DB de control no es coherente. 16#0005: El DB de fragmento no está disponible. 16#0006: El DB de fragmento no es para AWP. 16#0007: Los datos de enumeración no son coherentes. 16#000D: El tamaño del DB de control es conflictivo.

Tabla de órdenes

La tabla de órdenes es una matriz de estructuras que contiene comandos y estados que se aplican a determinados DB de fragmentos. Si ha creado fragmentos con el comando AWP_Start_Fragment (Página 903) de tipo "manual", el programa de usuario de STEP 7 debe controlar estas páginas con el DB de control. Los estados de órdenes solo se pueden leer y

ofrecen información sobre el fragmento actual. Es posible utilizar los comandos de órdenes para controlar el fragmento actual.

Variable de bloque	Tipo de datos	Descripción
requesttab	ARRAY [1 .. 4] OF STRUCT	Matriz de estructuras para controlar individualmente un DB de fragmentos. El servidor web puede procesar hasta cuatro fragmentos a la vez. El índice de matriz para un fragmento particular es arbitrario cuando el servidor web está procesando varios fragmentos o fragmentos de varias sesiones del navegador.

Elementos de la variable STRUCT requesttab

Variable de bloque	Tipo de datos	Descripción
page_index	UINT	Número de la página web actual
fragment_index	UINT	Número del fragmento actual - puede ajustarse para un fragmento diferente
// Comandos de órdenes		
continue	BOOL	Activa la transmisión para la página/fragmento actual y continúa con el siguiente fragmento
repeat	BOOL	Activa la retransmisión para la página/fragmento actual y continúa con el mismo fragmento
abort	BOOL	Cierra la conexión http sin transmitir
finish	BOOL	Transmite este fragmento; la página está completa - no procesa fragmentos adicionales
// Estados de órdenes		Los estados de órdenes solo se pueden leer
idle	BOOL	Ninguna orden pero activo
waiting	BOOL	El fragmento está en espera de ser activado
sending	BOOL	El fragmento está transmitiendo
aborting	BOOL	El usuario ha interrumpido la orden actual

Funcionamiento

Cada vez que el programa modifica el DB de control debe llamar la instrucción WWW aplicando el número del DB de control modificado como parámetro. Los comandos globales y de solicitud tienen efecto cuando el programa de usuario de STEP 7 ejecuta la instrucción WWW (Página 909).

El programa de usuario de STEP 7 puede definir el fragment_index explícitamente, lo cual hace que el servidor web procese el fragmento especificado con un comando de solicitud. De lo contrario, el servidor web procesa el fragmento actual para la página actual cuando se ejecuta la instrucción WWW.

Los métodos disponibles para utilizar el fragment_index incluyen:

- Procesar el fragmento actual: no modifique el fragment_index y active el comando "continue".
- Omitir el fragmento actual: ajuste el fragment_index a 0 y active el comando "continue".
- Reemplazar el fragmento actual por otro fragmento: ajuste en el fragment_index el nuevo ID de fragmento y active el comando "continue".

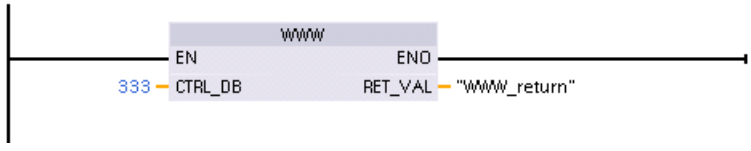
Para comprobar los estados globales o de solicitud que estén sujetos a cambio, el programa de usuario de STEP 7 debe llamar la instrucción WWW para evaluar los valores actuales de estos estados. Un caso típico puede ser llamar la instrucción WWW periódicamente hasta que ocurra un estado específico.

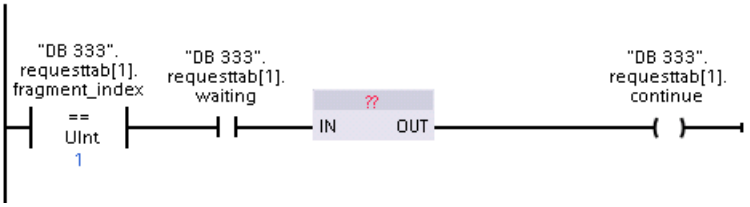
Nota

Si el programa de usuario de STEP 7 activa más de un comando de orden, la instrucción WWW procesa solo uno de acuerdo con el siguiente orden de prioridad: interrumpir, finalizar, repetir, continuar. La instrucción WWW borra todos los comandos de órdenes después del procesamiento.

Ejemplos

El ejemplo siguiente muestra un programa de usuario de STEP 7 comprobando que un fragmento con un ID de 1 esté en estado de espera, después de una llamada previa de la instrucción WWW. También puede esperar a que ocurran otras condiciones específicas de la aplicación. A continuación ejecuta el procesamiento que se requiera para el fragmento, por ejemplo configurar variables de bloque de datos, realizar cálculos u otras órdenes específicas de la aplicación. Después, activa la marca "continue" para que el servidor web ejecute este fragmento.





Cuando el programa llama la instrucción WWW con este DB de control modificado, la página web definida por el usuario a la que pertenece este fragmento puede visualizarse desde el navegador web.



Tenga en cuenta que se trata de un ejemplo simplificado. Realmente, el fragmento podría estar en cualquiera de los cuatro STRUCTS requesttab de la matriz. El programa debe procesar las cuatro estructuras requesttab.

12.8.12 Web API

12.8.12.1 Web API

La CPU S7-1200 proporciona una Web API, que es una interfaz que permite leer y escribir datos de proceso. La Web API del S7-1200 implementa la funcionalidad Web API de S7-1500, que se documenta en el capítulo **Páginas web > API (Application Programming Interface)** de este documento (https://support.industry.siemens.com/cs/us/en/view/59193560/es).

El documento correspondiente al S7-1500 describe la función Web API para una serie específica de CPU S7-1500. Esta funcionalidad está disponible para todas las CPU S7-1200 con firmware V4.5 o posterior. En STEP 7 hay que configurar la configuración del dispositivo como V4.5 o posterior.

Recuerde que la CPU S7-1200 limita el número de sesiones API simultáneas a 50.

La Web API de S7-1200 soporta los tipos de datos soportados por la Web API de S7-1500, exceptuando los siguientes:

Tipo de datos
ARRAY
BCD16
BCD32
VARIANT
REMOTE
HW_DEVICE
TP_TIME
TON_TIME
TOF_TIME
TONR_TIME
CTU_SINT
CTU_INT
CTU_DINT
CTU_USINT
CTU_UINT
CTU_UDINT
CTD_SINT
CTD_INT

CTD_DINT
CTD_USINT
CTD_UINT
CTD_UDINT
CTUD_SINT
CTUD_INT
CTUD_DINT
CTUD_USINT
CTUD_UINT
CTUD_UDINT

Encontrará los tipos de datos soportados por la Web API de S7-1500 aquí: **Páginas web > API (Application Programming Interface) > Lectura y escritura de datos de proceso > Tipos de datos soportados** en este documento (https://support.industry.siemens.com/cs/us/en/view/59193560/es).

12.8.12.2 Métodos Web API soportados

A partir de V4.5, el S7-1200 soporta los siguientes métodos Web API:

Métodos Web API soportados
Api.Ping
Api.Version
Api.GetCertificateUrl
Api.Browse
Api.Login
Api.Logout
Api.GetPermissions
PicProgram.Browse
PicProgram.Write
PicProgram.Read

12.9 Limitaciones

Los factores de TI siguientes pueden afectar al uso del servidor web:

- Generalmente, debe usar la dirección IP de la CPU para acceder a las páginas web estándar o las páginas web definidas por el usuario, o la dirección IP de un router inalámbrico con un número de puerto. Si el navegador web no permite la conexión directa a una dirección IP, consulte al administrador de TI. Si las políticas locales soportan DNS, el usuario se puede conectar a la dirección IP a través de una entrada DNS de dicha dirección.
- Los cortafuegos, la configuración del proxy y otras restricciones específicas del sitio también pueden restringir el acceso a la CPU. Consulte al administrador de TI para resolver estos problemas.

- Las páginas web estándar emplean JavaScript y cookies. Si la configuración del navegador web deshabilita el JavaScript o las cookies, habilítelos. Si no se pueden habilitar, algunas de las funciones están restringidas (Página 934). El uso de JavaScript y cookies en las páginas web definidas por el usuario es opcional. En caso de usarlos, es necesario habilitarlos en el navegador.
- El servidor web admite la capa de sockets seguros (SSL). Se puede acceder a las páginas web estándar y a las páginas web definidas por el usuario con una URL http://ww.xx.yy.zz o https://ww.xx.yy.zz, donde "ww.xx.yy.zz" representa la dirección IP de la CPU.
- Siemens ofrece un certificado de seguridad para el acceso seguro al servidor web. En la página web estándar de introducción (Página 862) es posible cargar e importar el certificado en las opciones de Internet del navegador web (Página 854). Si se elige no importar el certificado, aparece una solicitud de verificación de seguridad cada vez que se accede al servidor web https://.

Número de conexiones

El servidor web soporta un máximo de 30 conexiones activas. Varias acciones consumen las 30 conexiones, según el navegador web utilizado y el número de objetos diferentes por página (archivos .css, imágenes, archivos JavaScript, archivos .html adicionales). Algunas conexiones se mantienen mientras el servidor web está mostrando una página; otras conexiones no se mantienen después de la conexión inicial.

Si, por ejemplo, se utilizan determinadas versiones de Mozilla Firefox que soportan como máximo seis conexiones permanentes, se podrían emplear cinco instancias de navegador o de ficha de navegador antes de que el servidor web empiece a cerrar conexiones. Si una página no usa las seis conexiones, se pueden tener más instancias de pestañas de navegador o navegadores.

Tenga en cuenta que el número de conexiones activas puede afectar al rendimiento de presentación de páginas. Eso puede hacer que las páginas web no se carguen completamente.

Nota

Cierre de sesión antes de cerrar el servidor web

Si ha iniciado sesión en el servidor web, asegúrese de cerrarla antes de cerrar el navegador web. El servidor web soporta un máximo de siete sesiones activas al mismo tiempo.

Si no se cierra la sesión, según sea el navegador es posible que varias conexiones queden abiertas. En caso de abrir y cerrar muchas veces las páginas del navegador del servidor web sin cerrar sesión podrían agotarse las 30 conexiones disponibles. Si se gastan todas las conexiones se recibirá un mensaje de "Inicio de sesión no válido" al intentar iniciar sesión. En este caso, habrá que esperar hasta 30 minutos a que el servidor web libere suficientes conexiones para poder volver a iniciar sesión. Para evitar este problema, cierre siempre la sesión antes de cerrar el servidor web en caso de haber iniciado sesión previamente.

12.9.1 Uso de JavaScript

Las páginas web estándar emplean HTML, JavaScript y cookies. Si el sitio restringe el uso de JavaScript y cookies, hay que habilitarlos para que las páginas funcionen de manera adecuada. Si no se puede habilitar JavaScript para el navegador web no será posible ejecutar las páginas web estándar. Considere la posibilidad de utilizar páginas básicas, que no utilizan JavaScript.

Consulte también

Representación de las páginas web estándar (Página 857)

12.9.2 Restricciones de funciones cuando las opciones de Internet no permiten cookies

Si deshabilita las cookies en el navegador web, se aplican las restricciones siguientes:

- No puede iniciar sesión.
- No puede cambiar el ajuste de idioma.
- No puede cambiar de hora UTC a hora del PLC. Sin cookies, todas las horas están en hora UTC.

12.9.3 Reglas para introducir nombres de variables y valores

Al utilizar las páginas estándar Estado de variable (Página 875) y Tablas de observación (Página 877), tenga en cuenta las siguientes convenciones:

- Si se fuerza todo el valor de una variable DTL, por ejemplo "Data_block_1".DTL_tag, utilice la siguiente sintaxis DTL para el valor de forzado: DTL#YYYY-MM-DD-HH-MM-SS[.ssssssss]
- En caso de utilizar notación exponencial para introducir un valor del tipo de datos Real o LReal:
  - Para introducir un valor de número real (Real o LReal) con un exponente positivo (como +3,402823e+25), introduzca el valor en uno de estos formatos:
    +3,402823e25
    +3,402823e+25
  - Para introducir un valor de número real (Real o LReal) con un exponente negativo (como +3,402823e-25), introduzca el valor como se indica a continuación:
    +3,402823e-25
  - Asegúrese de que la mantisa del valor real en notación exponencial incluye un punto decimal. Si no se incluye un punto decimal, el valor se convertirá en un entero inesperado. Así, por ejemplo, introduzca -1.0e8 en lugar de -1e8.
- Los valores LReal pueden tener únicamente 15 dígitos (independientemente de la posición de la coma). Si se introducen más de 15 dígitos, se genera un error de redondeo.

Limitaciones de las páginas Estado de variable y Tabla de observación:

- El número máximo de caracteres de la URL es de 2083. La URL que representa la página actual se puede ver en la barra de direcciones del navegador.
- Para el formato de visualización de caracteres, si los valores reales de la CPU son caracteres ASCII no válidos según la interpretación del navegador, la página muestra el carácter precedido por el signo del dólar: $.

12.9.4 Importar registros de datos en formato CSV a versiones de Microsoft Excel que no son estadounidenses ni británicas

Los archivos de registro tienen el formato separado por comas (CSV). Estos archivos se pueden abrir directamente con Excel desde la página de Data Logs si su sistema utiliza una versión de Excel estadounidense o británica. Sin embargo, en otros países este formato no está muy difundido porque las comas están presentes frecuentemente en la notación numérica.

Para abrir un archivo de registro guardado previamente, proceda del siguiente modo en caso de utilizar otras versiones de Excel que no sean estadounidenses o británicas.

1. Abra Excel y cree un libro vacío.
2. Seleccione el comando "Importar datos" del menú "Datos > Obtener datos externos".
3. Navegue hasta el archivo de registros que desea abrir y selecciónelo. Se inicia el asistente de importación de texto.
4. En el asistente de importación de texto, cambie la opción predeterminada "Tipo de datos original" de "Longitud fija" a "Delimitado".
5. Haga clic en el botón "Siguiente".
6. En el diálogo del paso 2, seleccione la casilla de verificación "Coma" para cambiar el tipo de delimitador de "Tabulador" a "Coma".
7. Haga clic en el botón "Siguiente".
8. En el diálogo del paso 3 se puede cambiar el formato de fecha de MDY (month/day/year) a otro formato.
9. Lleve a cabo los pasos restantes del asistente de importación de texto para importar el archivo.

Procesador de comunicaciones y Modbus TCP 13

13.1 Utilizar las interfaces de comunicación serie

Dos módulos de comunicaciones (CM) y una Communication Board (CB) ofrecen la interfaz para la comunicación PtP:

- CM 1241 RS232 (Página 1421)
- CM 1241 RS422/485 (Página 1422)
- CB 1241 RS485 (Página 1418)

Se pueden conectar hasta tres CMs (de cualquier tipo) y una CB para un total de cuatro interfaces de comunicación. Instale el CM a la izquierda de la CPU o de otro CM. Instale la CB en la parte frontal de la CPU. Consulte las instrucciones de montaje (Página 55) para más información sobre el montaje y la retirada del módulo.

Las interfaces de comunicación serie tienen las siguientes características:

- Cuentan con un puerto aislado
- Soportan protocolos punto a punto
- Se configuran y programan mediante las instrucciones de procesador de comunicaciones punto a punto
- Muestran la actividad de transmisión y recepción mediante LED
- Muestran un LED de diagnóstico (sólo CM)
- Reciben alimentación de la CPU: No necesita conexión a una fuente de alimentación externa.

Consulte los datos técnicos de las interfaces de comunicación (Página 1408).

Indicadores LED

Los módulos de comunicación tienen tres indicadores LED:

- LED de diagnóstico (DIAG): este LED parpadea en color rojo hasta ser direccionado por la CPU. Tras el arranque de la CPU, ésta detecta si hay CMs y los direcciona. El LED de diagnóstico comienza a parpadear en color verde. Esto indica que la CPU ha direccionado el CM, pero aún no ha suministrado la configuración correspondiente. La CPU carga la configuración en los CMs configurados cuando el programa se carga en la CPU. Una vez cargado el programa en la CPU, el LED de diagnóstico del módulo de comunicación debería encenderse en color verde.
- LED de transmisión (Tx): el LED de transmisión se enciende cuando el puerto de comunicación envía datos.
- LED de recepción (Rx): este LED se enciende cuando el puerto de comunicación recibe datos.

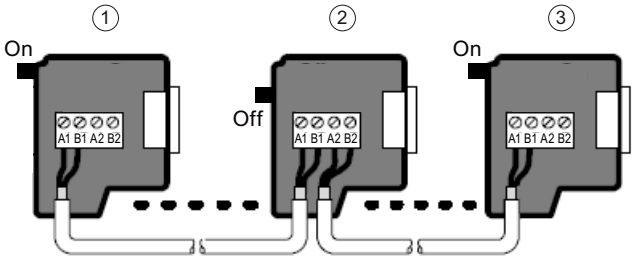
La Communication Board cuenta con un LED de transmisión (TxD) y uno de recepción (RxD). No tiene ningún LED de diagnóstico.

13.2 Polarizar y terminar un conector de red RS485

Siemens ofrece un conector de red RS485 (Página 1437) que permite conectar fácilmente varios aparatos a una red RS485. El conector posee dos juegos de terminales para fijar los cables de entrada y salida. También incluye interruptores para polarizar y terminar selectivamente la red.

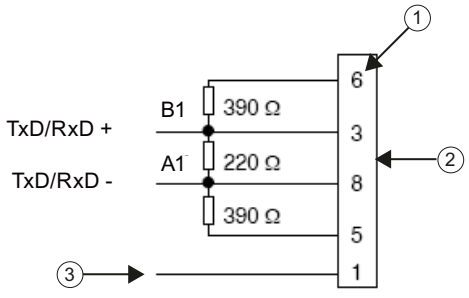
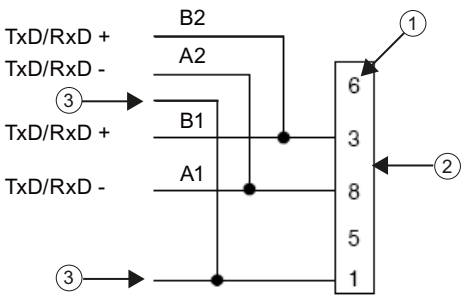
Nota

Solo se terminan y polarizan los dos extremos de la red RS485. Los dispositivos que están entre los dos dispositivos terminadores no se terminan ni polarizan. Pantalla de protección pelada: aprox. 12 mm (1/2 pulg.) deben hacer contacto con la guía de metal en todos los puntos.



① Selector en posición ON: terminado y polarizado
② Selector en posición OFF: no terminado ni polarizado
③ Selector en posición ON: terminado y polarizado

Tabla 13-1 Terminación y polarización para el conector RS485

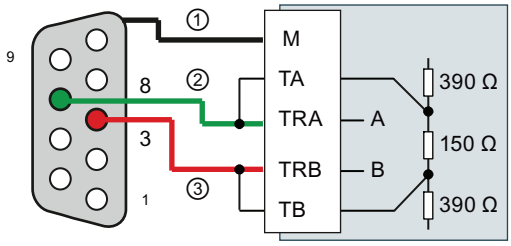
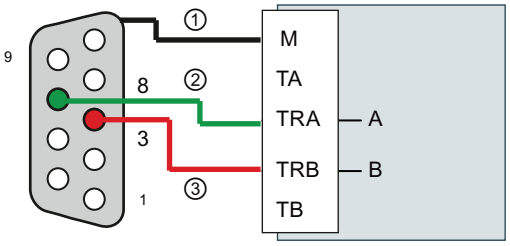
Dispositivo terminador (polarización ON)	Dispositivo no terminador (polarización OFF)
	

① Número de pin
② Conector de bus
③ Pantalla de cable

El CB 1241 ofrece resistores internos para terminar y polarizar la red. Para terminar y polarizar la conexión, conecte TRA con TA y TRB con TB con el fin de incluir los resistores internos en el

circuito. El CB 1241 no tiene ningún conector de 9 pines. La tabla siguiente muestra las conexiones con un conector de 9 pines del interlocutor.

Tabla 13-2 Terminación y polarización para el CB 1241

Dispositivo terminador (polarización ON)	Dispositivo no terminador (polarización OFF)
	

① Conecte M a la pantalla de protección
② A = TxD/RxD - (cable verde / pin 8)
③ B = TxD/RxD + (cable rojo / pin 3)

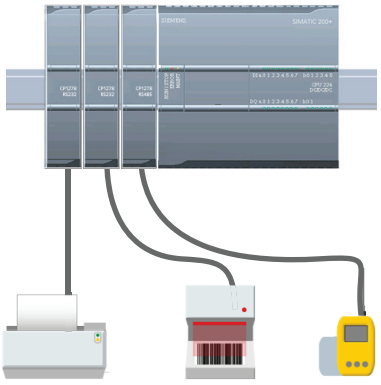
13.3 Comunicación punto a punto (PtP)

La CPU admite la siguiente comunicación punto a punto (PtP) para protocolos serie basados en caracteres:

- PtP, Freeport (Página 940)
- PtP, 3964(R) (Página 942)
- USS (Página 994)
- Modbus (Página 1013)

13.3.1 Comunicación PtP, Freeport

PtP con Freeport, o protocolo de construcción libre, ofrece una libertad y flexibilidad máximas, pero requiere una implementación exhaustiva en el programa de usuario.



PtP ofrece numerosas posibilidades, a saber:

- Posibilidad de enviar directamente información a un dispositivo externo, p. ej. una impresora
- Posibilidad de recibir información de otros dispositivos, p. ej. lectores de código de barras, lectores RFID, cámaras o sistemas de visión de terceros y muchos dispositivos más
- Posibilidad de intercambiar información (enviar y recibir datos) con otros dispositivos tales como equipos GPS, cámaras o sistemas de visión de terceros, módems radio y muchos más

Este tipo de comunicación PtP es una comunicación serie que utiliza UART estándar para admitir distintas velocidades de transferencia y opciones de paridad. Los módulos de comunicaciones RS232 y RS422/485 (CM 1241) y las Communication Board RS485 (CB 1241) ofrecen las interfaces eléctricas para realizar la comunicación PtP.

PtP Freeport vía PROFIBUS o PROFINET

PtP permite usar un rack de periferia descentralizada PROFINET o PROFIBUS para comunicarse con varios dispositivos (lectores RFID, dispositivos GPS y otros):

- PROFINET (Página 594): la interfaz Ethernet de la CPU S7-1200 se conecta a un módulo de interfaz PROFINET. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.
- PROFIBUS (Página 777): un módulo de comunicación PROFIBUS se inserta en el lado izquierdo del rack con la CPU S7-1200. El módulo de comunicación PROFIBUS se conecta al rack que contiene un módulo de interfaz PROFIBUS. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.

Por este motivo, S7-1200 soporta dos juegos de instrucciones PtP:

- Instrucciones Legacy punto a punto (Página 1112): estas instrucciones son anteriores a la versión V4.0 de S7-1200 y solo funcionan con la comunicaciones serie si se utiliza un módulo de comunicaciones CM 1241 o una Communication Board CB 1241.
- Instrucciones punto a punto (Página 959): Estas instrucciones ofrecen toda la funcionalidad de las instrucciones antiguas más la capacidad de soportar módulos de comunicación PtP mediante la periferia descentralizada de PROFINET y PROFIBUS. Las instrucciones punto a punto permiten acceder a los módulos de comunicación a través del rack de periferia descentralizada.
  Los módulos CM 1241 del S7-1200 deben tener como mínimo la versión de firmware V2.1 para poder utilizar las instrucciones punto a punto. Estos módulos están limitados al rack local situado a la izquierda de la CPU S7-1200. También se puede utilizar la instrucción punto a punto con un CB 1241.
  La comunicación a través de la periferia descentralizada utiliza los módulos siguientes:

Estación	Módulo	Referencia	Interfaz
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 y RS422/RS485

Nota

Las instrucciones punto a punto pueden utilizarse para acceder a una tarjeta de comunicación, módulos serie locales (o del lado izquierdo), módulos serie vía PROFINET y módulos serie vía PROFIBUS. STEP 7 ofrece las instrucciones antiguas punto a punto solo por su compatibilidad con programas existentes. Las instrucciones antiguas siguen funcionando con las CPU S7-1200 actuales. No es necesario convertir programas anteriores de un juego de instrucciones a otro.

Nota

Versión de firmware requerida del módulo CM para la sincronización horaria y la comunicación PtP

Si se ha activado "La CPU sincroniza los módulos del dispositivo" en las propiedades Sincronización horaria (Página 172) de la configuración de dispositivos de la interfaz PROFINET, actualice las versiones de firmware de los módulos de comunicación conectados a las últimas versiones disponibles. Si se activa la sincronización horaria de módulos de comunicación con versiones de firmware antiguas pueden producirse problemas y errores de comunicación.

13.3.2 Comunicación 3964(R)

La CPU S7-1200 soporta el protocolo 3964(R) para habilitar la comunicación entre un módulo CM 1241 RS232 o un módulo CM 1241 (RS422/485) y un interlocutor de comunicación que utilice el protocolo 3964(R). A diferencia de la comunicación PtP descrita anteriormente, en la que deben definirse características específicas de transmisión y recepción para los mensajes, el protocolo 3964(R) prescribe un protocolo estricto con los siguientes caracteres de control:

- STX Start of text
  Inicio de la cadena de caracteres que debe transmitirse
- DLE Data Link Escape
  Conmutación de la transmisión de datos
- ETX End of Text
  Fin de la cadena de caracteres que debe transmitirse
- BCC Block check character (carácter de comprobación de bloque)
- NAK Negative Acknowledge (acuse negativo)

Consulte el capítulo que describe los principios de la transmisión de datos en el manual Manual S7-300 Acoplamiento punto a punto CP 341, configuración y parametrización. (https://support.industry.siemens.com/cs/es/es/view/1117397) para obtener una descripción completa del protocolo.

Configurar el módulo de comunicación

Para comunicarse con un interlocutor mediante el protocolo 3964(R) hay que incluir uno de los siguientes módulos de comunicación en la configuración de dispositivos de STEP 7:

- CM 1241 (RS232)
- CM 1241 (RS422/485)

La versión de firmware del módulo CM debe ser V2.2.0 o superior.

Seguidamente, para el módulo de comunicación se configuran los puertos de comunicación (Página 943), la prioridad y los parámetros del protocolo (Página 957).

Comunicación con un interlocutor con el protocolo 3964(R)

Cuando se configura un CM para el protocolo 3964(R) se utilizan las instrucciones estándar de transmisión y recepción punto a punto para transferir datos entre la CPU y sus interlocutores.

El CM integra los datos del parámetro BUFFER de la instrucción de transmisión en el protocolo 3964(R) y envía los datos al interlocutor de la comunicación.

El CM recibe los datos del interlocutor de la comunicación a través del protocolo 3964(R), elimina la información del protocolo y devuelve los datos al parámetro BUFFER de la instrucción de recepción.

Consulte las siguientes instrucciones punto a punto:

- Send_P2P (transmitir datos del búfer de transmisión) (Página 973)
- Receive_P2P (habilitar recepción de mensajes) (Página 976)

También es posible utilizar las instrucciones antiguas de transmisión y recepción punto a punto:

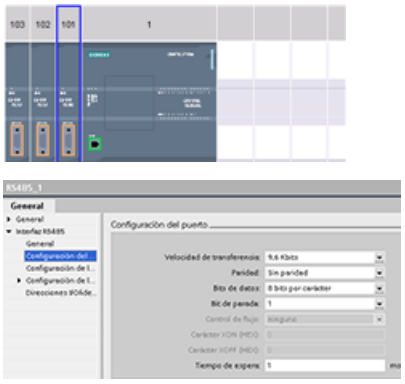
- SEND_PTP (transmitir datos del búfer de transmisión) (Página 1120)
- RCV_PTP (habilitar recepción de mensajes) (Página 1122)

13.3.3 Configurar la comunicación PtP Freeport

Puede usar uno de los métodos siguientes para configurar las interfaces de comunicación para la comunicación PtP Freeport:

- Utilizando la configuración de dispositivos en STEP 7 para configurar los parámetros de puerto (velocidad de transferencia y paridad), así como los de transmisión y recepción. La CPU almacena los ajustes de configuración de dispositivos y aplica los ajustes tras desconectar y conectar la alimentación y una transición de RUN a STOP.
- Use las instrucciones Port_Config (Página 961), Send_Config (Página 964) y Receive_Config (Página 966) para ajustar los parámetros. La configuración del puerto ajustada mediante las instrucciones es válida mientras la CPU esté en modo RUN. La configuración del puerto se inicializa con los valores predeterminados de la configuración del dispositivo tras cambiar a STOP o desconectar y volver a conectar la alimentación.

Tras configurar los dispositivos de hardware (Página 137) es preciso ajustar los parámetros de las interfaces de comunicación seleccionando uno de los CMs del rack o bien la CB, de estar configurada.



La ficha "Propiedades" de la ventana de inspección muestra los parámetros del CM o de la CB seleccionada. Seleccione "Configuración del puerto" para editar los parámetros siguientes:

- Velocidad de transferencia
- Paridad
- Bits de datos por carácter
- Número de bits de parada
- Control de flujo (solo RS232)
- Tiempo de espera

Para el CM 1241 RS232 y la CB RS485 (excepto el control de flujo (Página 945), que solo admite el CM 1241 RS232), los parámetros de configuración del puerto son iguales, indistintamente de si se desea configurar un módulo de comunicación RS232 o RS485 o bien la placa de comunicación RS485. Los valores de los parámetros pueden diferir.

En el caso del CM 1241 RS422/485, dispone de opciones adicionales para la configuración del puerto como se muestra a continuación. El modo 422 del módulo CM 1241 RS422/485 también admite el control de flujo por software.



Seleccione "Configuración del puerto" para editar los parámetros de RS422/485 siguientes:

- "Estado operativo":
  - Modo dúplex (RS422) de cuatro hilos (conexión punto a punto)
  - Modo dúplex (RS422) de cuatro hilos (maestro multipunto)
  - Modo dúplex (RS422) de cuatro hilos (esclavo multipunto)
  - Modo semidúplex (RS485) de dos hilos
- "Inicialización de la línea de recepción":
  - Ninguno
  - Polarización directa (señal R(A) 0V, señal R(B) 5V)

El programa de usuario de STEP 7 también puede configurar el puerto o modificar la configuración existente con la instrucción Port_Config (Página 961). El apartado correspondiente a la instrucción ofrece más detalles sobre el modo de operación y el estado inicial de la línea así como otros parámetros.

Parámetro	Definición
Velocidad de transferencia	El valor predeterminado para la velocidad de transferencia es 9,6 Kbits/s. Los ajustes posibles son: 300 baudios, 600 baudios, 1,2 Kbits, 2,4 Kbits, 4,8 Kbits, 9,6 Kbits, 19,2 Kbits, 38,4 Kbits, 57,6 Kbits, 76,8 Kbits y 115,2 Kbits.
Paridad	El valor predeterminado para la paridad es "sin paridad". Los ajustes posibles son: Sin paridad, par, impar, marca (bit de paridad siempre establecido a 1), espacio (bit de paridad siempre establecido a 0).
Bits de datos por carácter	Número de bits de datos de un carácter. Los valores válidos son 7 u 8.
Número de bits de parada	Es posible ajustar uno o dos bits de parada. El ajuste predeterminado es uno.
Control de flujo	En el caso del módulo de comunicación RS232, puede seleccionar el control de flujo (Página 945) del hardware o del software. Si se selecciona el control de flujo por hardware, es posible indicar si la señal RTS debe estar siempre ON o si debe conmutarse. Si se selecciona el control de flujo por software, es posible definir los caracteres XON y XOFF. Las interfaces de comunicación RS485 no soportan el control de flujo. El modo 422 del módulo CM 1241 RS422/485 soporta el control de flujo por software.
Tiempo de espera	El tiempo de espera especifica el periodo que el CM o la CB espera hasta recibir CTS tras confirmar RTS, o bien hasta recibir un XON tras recibir un XOFF, dependiendo del tipo de control de flujo. Si el tiempo de espera transcurre antes de que la interfaz de comunicación reciba un CTS o XON esperado, el CM o la CB cancelará la operación de transmisión y devolverá un error al programa de usuario. El tiempo de espera se indica en milisegundos. El rango válido está comprendido entre 0 y 65535 milisegundos.
Estado operativo	Selecciona el modo operativo de RS422 o RS485 y las configuraciones de red.
Inicialización de la línea de recepción	Selecciona las opciones de polarización. Los valores válidos son Ninguno, Polarización directa y Polarización inversa. La polarización inversa sirve para permitir la detección de rotura de cable.

13.3.3.1 Gestionar el control de flujo

El control de flujo es un mecanismo que permite regular el intercambio de datos entre un emisor y un receptor para evitar pérdidas de datos. El control de flujo garantiza que un emisor no envíe más información de la que el receptor es capaz de procesar. El control de flujo puede realizarse por hardware o software. El CM RS232 soporta el control de flujo tanto por hardware como por software. El CM y la CB RS485 no soportan el control de flujo. El modo 422 del módulo CM 1241 RS422/485 soporta el control de flujo por software. El tipo de control de flujo se especifica al configurar el puerto (Página 943) o con la instrucción PORT_CFG (Página 1112).

El control de flujo por hardware funciona a través de las señales de comunicación RTS (Request To Send o petición de transmitir) y CTS (Clear To Send o listo para transmitir). En el caso del CM RS232, la señal RTS se emite desde el pin 7 y la señal CTS se recibe por el pin 8. El CM RS232 es un DTE (Data Terminal Equipment o equipo terminal de datos) que confirma RTS como salida y monitoriza CTS como entrada.

Control de flujo por hardware: RTS conmutado

Si se habilita el control de flujo por hardware con RTS conmutado en un CM RS232, el módulo activa la señal RTS para enviar datos. El módulo vigila la señal CTS para determinar si el receptor puede aceptar datos. Estando activa la señal CTS, el módulo puede transmitir datos mientras que la señal CTS permanezca activa. Si se desactiva la señal CTS, la transmisión debe detenerse.

La transmisión se reanuda cuando se reactiva la señal CTS. Si la señal CTS no se vuelve a activar dentro del tiempo de espera configurado, el módulo cancelará la transmisión y devolverá un error al programa de usuario. El tiempo de espera se especifica en la configuración del puerto (Página 943).

El control de flujo con RTS conmutado es útil para los dispositivos que requieren una señal de que la transmisión está activa. Un ejemplo sería un módem radio que utiliza RTS como señal "Key" para energizar el transmisor de radio. El control de flujo con RTS conmutado no funciona con módems telefónicos estándar. Seleccione la opción "RTS siempre on" para los módems telefónicos.

Control de flujo por hardware: RTS siempre ON

Si se utiliza la opción "RTS siempre on", el CM 1241 activará RTS de forma predeterminada. Un dispositivo (p. ej. un módem telefónico) vigila la señal RTS del CM y la utiliza como CTS (Clear To Send o listo para transmitir). El módem transmitirá al CM sólo si RTS está activo, es decir, cuando el módem telefónico detecte un CTS activo. Si RTS no está activo, el módem telefónico no transmitirá al CM.

Para permitir que el módem envíe datos al CM en cualquier momento, configure el control de flujo por hardware con "RTS siempre ON". De esta manera, el CM activa la señal RTS permanentemente. El CM no desactivará RTS incluso si el módulo no puede aceptar caracteres. El emisor debe garantizar que no se desborde el búfer de recepción del CM.

Uso de las señales DTR (Data Terminal Ready) y DSR (Data Set Ready)

El CM activa DTR para cualquier tipo de control de flujo por hardware. El módulo transmite datos sólo cuando se activa la señal DSR. El estado de DSR se evalúa únicamente al comienzo de la transmisión. Si DSR se desactiva tras haberse iniciado la transmisión, ésta no se detendrá.

Control de flujo por software

El control de flujo por software utiliza caracteres especiales en los mensajes para proporcionar el control de flujo. Se configuran caracteres hexadecimales que representan XON y XOFF.

XOFF indica que una transmisión debe detenerse. XON indica que una transmisión puede reanudarse. XOFF y XON no deben ser el mismo carácter.

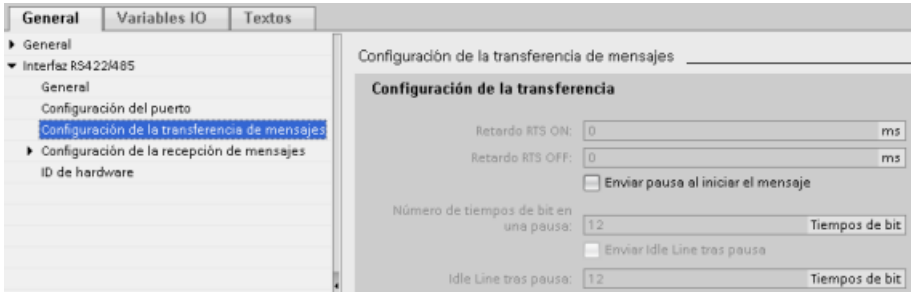
Cuando el emisor recibe un carácter XOFF del receptor, deja de transmitir datos. La transmisión se reanuda cuando el emisor recibe un carácter XON. Si no recibe un carácter XON dentro del tiempo de espera indicado en la configuración del puerto (Página 943), el CM cancelará la transmisión y devolverá un error al programa de usuario.

Para el control de flujo por software se requiere la comunicación dúplex, puesto que el receptor debe poder enviar XOFF al emisor durante una transmisión. El control de flujo por software sólo es posible en los mensajes que contengan únicamente caracteres ASCII. Los protocolos binarios no pueden utilizar el control de flujo por software.

Antes de que la CPU pueda iniciar una comunicación PtP Freeport hay que configurar parámetros para transmitir y recibir mensajes. Estos parámetros determinan cómo deben funcionar las comunicaciones al transmitir o recibir mensajes a/de un dispositivo de destino.

13.3.3.2 Configurar los parámetros de transmisión

Desde la configuración de dispositivos de la CPU se ajusta cómo una interfaz de comunicación debe transmitir datos configurando las propiedades "Configuración de la transferencia de mensajes" para la interfaz seleccionada.



Los parámetros de transmisión de mensajes también se pueden configurar o modificar dinámicamente desde el programa de usuario utilizando la instrucción Send_Config (Página 964).

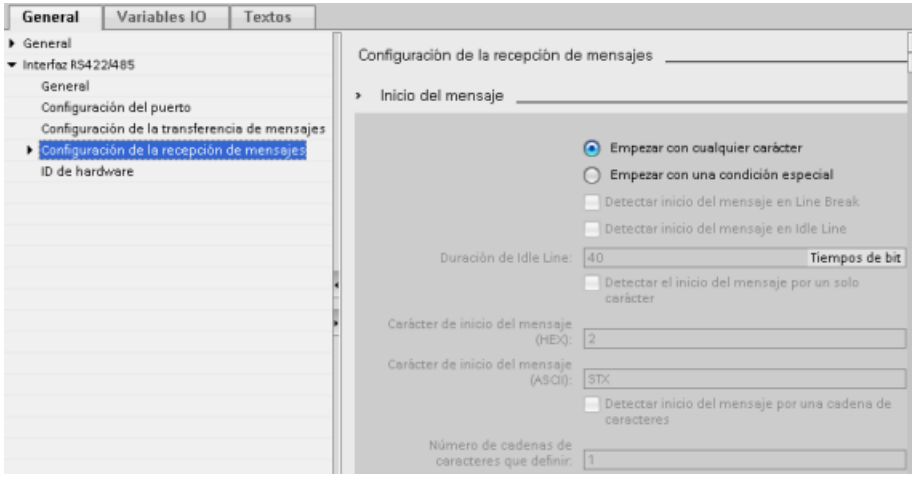
Nota

Los valores de parámetros ajustados mediante la instrucción Send_Config en el programa de usuario prevalecen sobre las propiedades de "Configuración de la transferencia de mensajes". La CPU no conserva los parámetros ajustados mediante la instrucción Send_Config si se desconecta la alimentación.

Parámetro	Definición
Retardo RTS ON	Determina el tiempo que debe esperarse tras activar RTS antes de iniciar la transmisión. El rango está comprendido entre 0 y 65535 ms (el valor predeterminado es 0). Este parámetro solo es válido si en la configuración del puerto (Página 943) se ha definido el control de flujo por hardware. CTS se evalúa una vez transcurrido el retardo RTS ON. Este parámetro es aplicable únicamente a los módulos RS232.
Retardo RTS OFF	Determina el tiempo que debe esperarse antes de desactivar RTS tras finalizar la transmisión. El rango está comprendido entre 0 y 65535 ms (el valor predeterminado es 0). Este parámetro solo es válido si en la configuración del puerto (Página 943) se ha definido el control de flujo por hardware. Este parámetro es aplicable únicamente a los módulos RS232.
Enviar pausa al inicio del mensaje Número de bit times en una pausa	Determina que, al inicio de cada mensaje, se enviará una pausa una vez transcurrido el tiempo de retardo RTS ON (si se ha configurado) y si CTS está activo. Es preciso indicar cuántos tiempos de bit ("bit times") constituyen una pausa cuando la línea se mantiene en una condición "Space". El ajuste predeterminado es 12 y el valor máximo es 65535, hasta un límite de ocho segundos.
Enviar Idle Line tras la pausa Idle Line tras pausa	Determina que se enviará una "idle line" antes del inicio del mensaje. Esta se enviará tras una pausa, siempre y cuando esté configurada una pausa. El parámetro "Idle Line tras pausa" indica cuántos tiempos de bit constituyen una "idle line" cuando la línea se mantiene en una condición "Mark". El ajuste predeterminado es 12 y el valor máximo es 65535, hasta un límite de ocho segundos.

13.3.3.3 Configurar los parámetros de recepción

En la configuración de dispositivos de la CPU se determina cómo debe recibir datos una interfaz de comunicación, así como detectar el comienzo y fin de un mensaje. Estos parámetros se configuran en las propiedades de "Configuración de la recepción de mensajes" para la interfaz seleccionada.



Los parámetros de recepción de mensajes también se pueden configurar o modificar dinámicamente desde el programa de usuario utilizando la instrucción Receive_Config (Página 966).

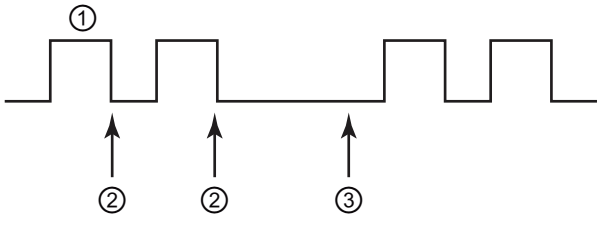
Nota

Los valores de parámetros ajustados mediante la instrucción Receive_Config en el programa de usuario prevalecen sobre las propiedades de "Configuración de la recepción de mensajes". La CPU no conserva los parámetros ajustados mediante la instrucción RCV_CFG si se desconecta la alimentación o hay una transición a STOP.

Condiciones de inicio del mensaje

Es posible determinar cómo la interfaz de comunicación debe detectar el inicio de un mensaje. Los caracteres iniciales y los caracteres que contiene el mensaje se depositan en el búfer de recepción hasta que se cumpla una condición de fin configurada.

Es posible definir varias condiciones de inicio. Si se especifica más de una condición de inicio, todas las condiciones de inicio deberán cumplirse antes de que el mensaje se considere como iniciado. Por ejemplo, si se configura un tiempo de "idle line" y un carácter de inicio específico, el CM o la CB determinará primero si se cumple el requisito de tiempo de "idle line" y buscará luego el carácter de inicio indicado. Si se recibe algún otro carácter (que no sea el carácter de inicio indicado), el CM o la CB reiniciará la búsqueda del inicio del mensaje buscando nuevamente el tiempo de "idle line".

Parámetro	Definición
Empezar con cualquier carácter	La condición "Carácter cualquiera" determina que cualquier carácter que se reciba correctamente indicará el inicio de un mensaje. Este carácter es el primer carácter del mensaje.
Line Break	La condición "Line Break" especifica que la recepción de un mensaje comienza cuando se reciba un carácter de salto de línea.
Idle Line	La condición "Idle Line" especifica que la recepción de un mensaje comienza una vez que la línea de recepción haya estado inactiva o en reposo durante el número de tiempos de bit indicado. Cuando se cumple esta condición, se inicia el mensaje.  ① Caracteres ② Reinicia el temporizador de "idle line" ③ Se ha detectado la "idle line" e iniciado la recepción de un mensaje

Parámetro	Definición
Condición especial: Detectar el inicio del mensaje por un solo carácter	Determina que un carácter en particular indique el principio de un mensaje. Este carácter es por consiguiente el primer carácter del mensaje. Cualquier carácter que se reciba antes de este carácter específico se descartará. El carácter predeterminado es STX.
Condición especial: Detectar inicio del mensaje por una cadena de caracteres	Determina que una secuencia de caracteres en particular de hasta cuatro secuencias configuradas indica el inicio de un mensaje. Es posible especificar cinco caracteres como máximo para una secuencia. Para cada posición de carácter se indica bien sea un carácter hexadecimal específico, o bien que el carácter se ignore en la comparación de secuencias (carácter comodín). El último carácter específico de una secuencia de caracteres finaliza esa secuencia de condiciones de inicio. Las secuencias entrantes se comparan con las condiciones de inicio configuradas hasta que se cumple una condición de inicio. Una vez que cumplida la secuencia de inicio, se iniciará la evaluación de las condiciones de fin. Es posible configurar hasta cuatro secuencias de caracteres específicos. Utilice una secuencia múltiple de condiciones de inicio cuando diferentes secuencias de caracteres pueden indicar el inicio de un mensaje. Si alguna de las secuencias de caracteres concuerda, se iniciará el mensaje.

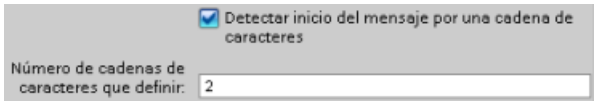
El orden de comprobación de las condiciones de inicio es el siguiente:

- Idle Line
- Line Break
- Caracteres o secuencias de caracteres

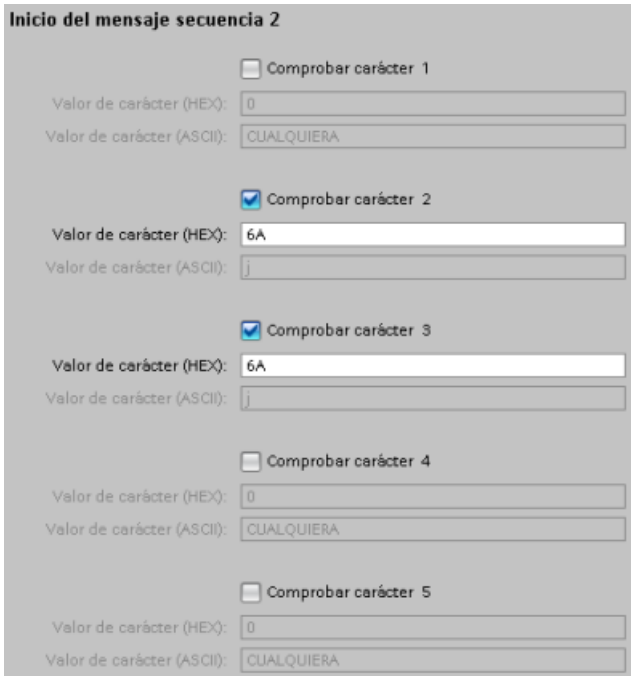
Si al comprobar varias condiciones de inicio no se cumple una de ellas, el CM o la CB reiniciará la comprobación con la primera condición requerida. Una vez el CM o la CB determina que se han cumplido las condiciones de inicio, procede a comparar las condiciones de fin.

Configuración de ejemplo: iniciar mensaje en una o dos secuencias de caracteres

Tenga en cuenta la siguiente configuración de las condiciones de inicio del mensaje:




Secuencia de 5 caracteres para iniciar el mensaje
Inicio del mensaje secuencia 1
Comprobar carácter 1
Valor de carácter (HEX): 6A
Valor de carácter (ASCII): j
Comprobar carácter 2
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA
Comprobar carácter 3
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA
Comprobar carácter 4
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA
Comprobar carácter 5
Valor de carácter (HEX): 1C
Valor de carácter (ASCII): FS


Inicio del mensaje secuencia 2
Comprobar carácter 1
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA
Comprobar carácter 2
Valor de carácter (HEX): 6A
Valor de carácter (ASCII): j
Comprobar carácter 3
Valor de carácter (HEX): 6A
Valor de carácter (ASCII): j
Comprobar carácter 4
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA
Comprobar carácter 5
Valor de carácter (HEX): 0
Valor de carácter (ASCII): CUALQUIERA

En esta configuración, la condición de inicio se cumple cuando se presenta uno de los patrones siguientes:

- Cuando se recibe una secuencia de cinco caracteres en la que el primer carácter es 0x6A y, el quinto, 0x1C. Los caracteres de las posiciones 2, 3 y 4 pueden ser un carácter cualquiera en esta configuración. Tras recibirse el quinto carácter comienza la evaluación de las condiciones de fin.
- Cuando se reciben dos caracteres 0x6A consecutivos precedidos de un carácter cualquiera. En este caso, la evaluación de las condiciones de fin comienza tras recibirse el segundo 0x6A (3 caracteres). El carácter que precede el primer 0x6A se incluye en la condición de inicio.

Secuencias de ejemplo que cumplirían esta condición de inicio:

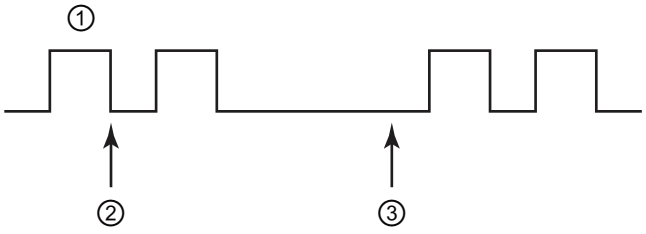
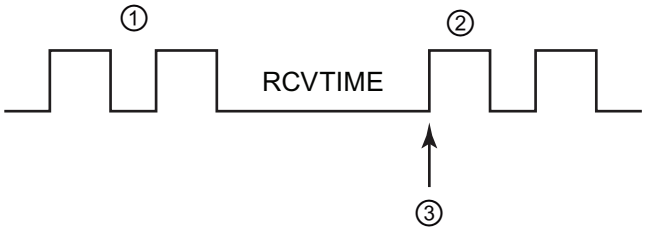
- <carácter cualquiera> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

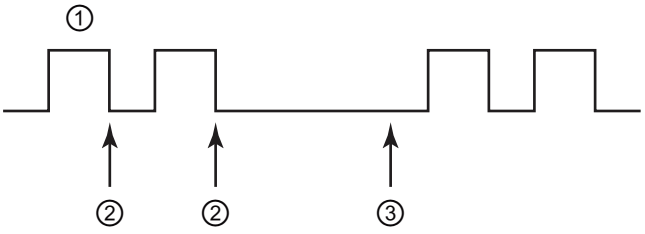
Condiciones de fin del mensaje

También es posible definir cómo la interfaz de comunicación debe detectar el fin de un mensaje. Es posible configurar varias condiciones de fin del mensaje. Si se cumple alguna de las condiciones configuradas, finalizará el mensaje.

Por ejemplo, es posible especificar una condición de fin con un timeout de fin de mensaje de 300 milisegundos, un timeout entre caracteres de 40 tiempos de bit y una longitud máxima de 50

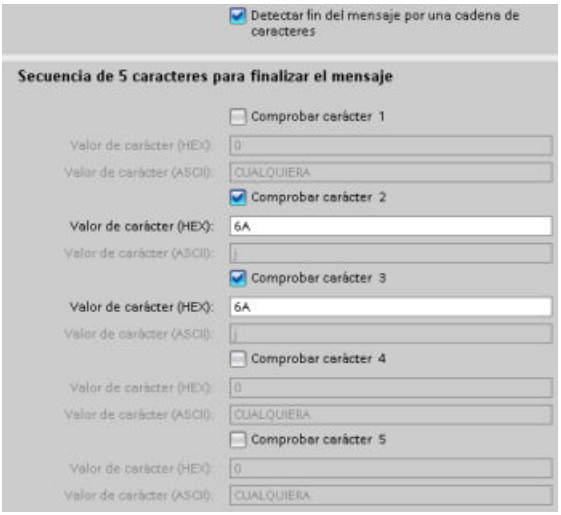
bytes. El mensaje terminará si la recepción tarda más de 300 milisegundos o si la distancia temporal entre dos caracteres es superior a 40 tiempos de bit o si se reciben 50 bytes.

Parámetro	Definición
Detectar fin del mensaje por tiempo de mensaje excedido	El mensaje finaliza cuando ha transcurrido el tiempo de espera configurado para el fin del mensaje. El periodo de timeout del mensaje comienza cuando se ha cumplido una condición de inicio. El valor predeterminado es 200 ms. El rango válido está comprendido entre 0 y 65535 ms.  ① Caracteres recibidos ② Condición de inicio del mensaje cumplida: el temporizador del mensaje arranca ③ El temporizador del mensaje se detiene y finaliza el mensaje
Detectar fin del mensaje por tiempo de respuesta excedido	El mensaje finaliza cuando ha transcurrido el tiempo de espera configurado para una respuesta antes de que se reciba una secuencia de inicio válida. El periodo de timeout de respuesta comienza cuando una transmisión finaliza y el CM la CB inicia la recepción. El timeout de respuesta predeterminado es de 200 ms. El rango válido está comprendido entre 0 y 65535 ms. Si un carácter no se recibe dentro del periodo de tiempo de respuesta RCVTIME, entonces se devuelve un error a la instrucción RCV_PTP correspondiente. El timeout de respuesta no define una condición de fin específica. Solo determina que un carácter tiene que recibirse correctamente dentro del tiempo indicado. Hay que configurar otra condición de fin para indicar el fin real de un mensaje.  ① Caracteres transmitidos ② Caracteres recibidos ③ En este momento el primer carácter debe haber sido recibido correctamente.

Parámetro	Definición
Detectar fin del mensaje por tiempo excedido entre caracteres	El mensaje finaliza cuando ha transcurrido el timeout máximo configurado entre cualquier par de caracteres consecutivos de un mensaje. El valor predeterminado del tiempo excedido entre caracteres es 12 tiempos de bit y el valor máximo es 65535 tiempos de bit, hasta un límite máximo de ocho segundos.  ① Caracteres recibidos ② Reinicia el temporizador entre caracteres ③ El temporizador entre caracteres se detiene y finaliza el mensaje.
Detectar fin del mensaje recibiendo un número fijo de caracteres	El mensaje finaliza cuando se ha recibido el número de caracteres especificado. El rango válido de la longitud fija está comprendido entre 1 y 4096. Tenga en cuenta que para el S7-1200, esta condición de fin solo es válida para CPU V4.0 o superiores.
Detectar fin del mensaje por longitud máxima	El mensaje finaliza cuando se ha recibido el número de caracteres máximo configurado. El rango válido de la longitud máxima está comprendido entre 1 y 1024. Esta condición sirve para impedir un error de desbordamiento del búfer de mensajes. Si esta condición de fin se combina con condiciones de fin por tiempo excedido (timeout) y ocurre una condición de timeout, los caracteres válidos recibidos hasta entonces estarán disponibles, aunque no se haya alcanzado la longitud máxima. Esto permite soportar protocolos de longitud variable si solo se conoce la longitud máxima.
Leer longitud del mensaje en el mensaje	El mensaje en sí especifica la longitud del mensaje. El mensaje finaliza cuando se recibe un mensaje con la longitud especificada. El método para especificar e interpretar la longitud del mensaje se describe más adelante.
Detectar fin del mensaje por un carácter	El mensaje finaliza cuando se recibe un carácter especificado.
Detectar fin del mensaje por una cadena de caracteres	El mensaje finaliza cuando se recibe una secuencia de caracteres especificada. Es posible especificar una secuencia de cinco caracteres como máximo. Para cada posición de carácter se indica bien sea un carácter hexadecimal específico, o bien que el carácter se ignore en la comparación de secuencias. Los caracteres iniciales ignorados no forman parte de la condición de fin. Los caracteres finales ignorados forman parte de la condición de fin.

Configuración de ejemplo: fin del mensaje con una cadena de caracteres

Tenga en cuenta la siguiente configuración de las condiciones de fin del mensaje:



En este caso, la condición de fin se cumple cuando se reciben dos caracteres 0x6A consecutivos, seguidos de dos caracteres cualesquiera. El carácter que precede el patrón 0x6A 0x6A no forma parte de la secuencia de caracteres de fin. Los dos caracteres que le siguen al patrón 0x6A 0x6A se requieren para terminar la secuencia de caracteres de fin. Aunque los valores recibidos de las posiciones de carácter 4 y 5 son irrelevantes, deben recibirse para que se cumpla la condición de fin.

Nota

Si desea que su secuencia de caracteres indique el final del mensaje, coloque la secuencia en las últimas posiciones de los caracteres. En el ejemplo anterior, si quiso que 0x6A 0x6A finalizara el mensaje sin caracteres finales, debió configurar 0x6A en las posiciones de los caracteres 4 y 5.

Indicación de la longitud del mensaje dentro del mensaje

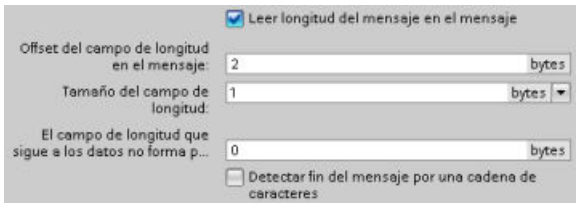
Si se selecciona la condición especial en la que la longitud del mensaje se incluye en el mensaje, es preciso indicar tres parámetros que definen la información acerca de la longitud del mensaje.

La estructura real del mensaje varía según el protocolo utilizado. Los tres parámetros son los siguientes:

- n: Posición de carácter (en base 1) dentro del mensaje que inicia el indicador de longitud
- Tamaño de longitud: Número de bytes (uno, dos o cuatro) del indicador de longitud
- Longitud m: Número de caracteres posteriores al indicador de longitud que no se incluyen en el contaje de longitud

Los caracteres de fin no tienen que ser contiguos. El valor "Longitud M" puede utilizarse para indicar la longitud de un campo de suma de verificación cuyo tamaño no se incluye en el campo de longitud.

Estos campos aparecen en la configuración de la recepción de mensajes de las propiedades del dispositivo:



Ejemplo 1: Considerar un mensaje estructurado según el protocolo siguiente:

STX	Len (n)	Caracteres 3 a 14 contados según la longitud											
		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

Configure los parámetros de longitud de recepción de este mensaje como se indica a continuación:

- n = 2 (la longitud del mensaje comienza con el byte 2.)
- Tamaño de longitud = 1 (la longitud del mensaje se define en un byte.)
- Longitud m = 0 (no hay caracteres adicionales posteriores al indicador de longitud no incluidos en el contaje de longitud. Doce caracteres le siguen al indicador de longitud.)

En este ejemplo, los caracteres 3 a 14 (inclusive) son los caracteres que cuenta Len (n).

Ejemplo 2: Considerar otro mensaje estructurado según el protocolo siguiente:

SD1	Len (n)	Len (n)	SD2	Caracteres 5 a 10 contados según la longitud						FCS	ED
				DA	SA	FA	Unidad de datos=3 bytes				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Configure los parámetros de longitud de recepción de este mensaje como se indica a continuación:

- n = 3 (la longitud del mensaje comienza con el byte 3.)
- Tamaño de longitud = 1 (la longitud del mensaje se define en un byte.)
- Longitud m = 3 (tres caracteres posteriores al indicador de longitud no se cuentan en la longitud. En el protocolo de este ejemplo, los caracteres SD2, FCS y ED no se incluyen en el contaje de longitud. Los seis caracteres restantes se incluyen en el contaje de longitud. Por tanto, el número total de caracteres posteriores al indicador de longitud es nueve.)

En este ejemplo, los caracteres 5 a 10 (inclusive) son los caracteres que cuenta Len (n).

13.3.4 Configurar la comunicación 3964(R)

13.3.4.1 Configurar los puertos de comunicación 3964(R)

Puede usar uno de los métodos siguientes para configurar las interfaces de comunicación para la comunicación 3964(R):

- Utilice la configuración de dispositivos en STEP 7 para configurar los parámetros de puerto. La CPU almacena los ajustes de configuración de dispositivos y los aplica tras desconectar y conectar la alimentación.
- Use la instrucción Port_Config (Página 961) para ajustar los parámetros de puerto. La configuración del puerto ajustada mediante las instrucciones es válida mientras la CPU esté en estado RUN. La configuración del puerto se inicializa con los valores predeterminados de la configuración del dispositivo tras desconectar y volver a conectar la alimentación.

Tras agregar las interfaces de comunicación a la configuración de dispositivos (Página 141) es preciso ajustar los parámetros de las interfaces de comunicación seleccionando uno de los CM del rack.





La ficha "Propiedades" de la ventana de inspección muestra los parámetros del CM seleccionado. Seleccione "Configuración del puerto" para editar los parámetros siguientes:

- Protocolo: 3964(R)
- Estado operativo (solo módulo CM 1241 (RS422/485))
- Estado inicial de la línea de recepción (solo módulo CM 1241 (RS422/485))
- Rotura de hilo (solo módulo CM 1241 (RS422/485))
- Velocidad de transferencia
- Paridad
- Bits de datos
- Bits de parada

Parámetro	Definición
Protocolo	3964R o Freeport. Seleccione 3964R para configurar el puerto para la comunicación 3964(R)
Estado operativo*	Dúplex (RS422) a cuatro hilos punto a punto. (Activado)
Estado inicial de la línea de recepción*	Active una de las opciones siguientes: • Ninguno • Polarización con R(A)>R(B)>=0V • Polarización con R(B)>R(A)>=0V

Parámetro	Definición
Rotura de hilo*	Active una de las opciones siguientes: • Sin comprobación de rotura de hilo • Activar comprobación de rotura de hilo
Velocidad de transferencia	El valor predeterminado para la velocidad de transferencia es de 9,6 Kbits/s. Los ajustes posibles son: 300 baudios, 600 baudios, 1,2 Kbits, 2,4 Kbits, 4,8 Kbits, 9,6 Kbits, 19,2 Kbits, 38,4 Kbits, 57,6 Kbits, 76,8 Kbits y 115,2 Kbits.
Paridad	El valor predeterminado para la paridad es "sin paridad". Los ajustes posibles son: Sin paridad, par, impar, marca (bit de paridad siempre puesto a 1), espacio (bit de paridad siempre puesto a 0) y cualquier paridad (poner a 0 el bit de paridad para transmisión, ignorar error de paridad a la recepción).
Bits de datos por carácter	Número de bits de datos de un carácter. Los valores válidos son 7 u 8.
Número de bits de parada	Es posible ajustar uno o dos bits de parada. El ajuste predeterminado es uno.

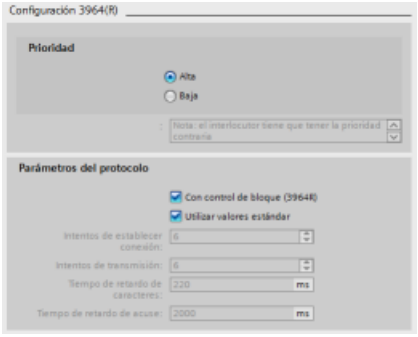
* Solo módulo CM 1241 (RS422/485)

13.3.4.2 Configurar la prioridad y los parámetros de protocolo para 3964(R)

Puede usar uno de los métodos siguientes para configurar las interfaces de comunicación para la comunicación 3964(R):

- En la configuración de dispositivos de la interfaz de comunicación, haga clic en "Configuración 3964(R)" para ajustar la prioridad y configurar los parámetros de protocolo. La CPU almacena los ajustes de configuración de dispositivos y los aplica tras desconectar y conectar la alimentación.
- Utilice la instrucción P3964_Config (Página 971) para ajustar la prioridad y los parámetros de configuración del protocolo. Los valores ajustados mediante las instrucciones son válidos mientras la CPU esté en estado RUN. Los valores se inicializan con los valores predeterminados de la configuración del dispositivo tras desconectar y volver a conectar la alimentación.





La ficha "Propiedades" de la ventana de inspección muestra los parámetros del CM seleccionado. Seleccione "Configuración 3964(R)" para editar los parámetros siguientes:

- Prioridad (alta o baja)
- Parámetros de protocolo
  - Con comprobación de bloque (3964R)
  - Utilizar valores predeterminados
    Intentos de conexión
    Intentos de transmisión
    Tiempo de retardo de carácter
    Retardo de acuse

Parámetro	Definición
Prioridad	Alta o baja: El CM tendrá la prioridad alta o baja y el interlocutor de la comunicación debe tener la opuesta.
Con comprobación de bloque (3964)	Si está seleccionado, la comunicación 3964(R) emplea la seguridad de la transmisión incluyendo un carácter de comprobación de bloque (BCC). Si no está seleccionado, la seguridad de la transmisión no incluye un carácter de comprobación de bloque.
Utilizar valores predeterminados	Si está seleccionado, 3964(R) utiliza valores predeterminados para los siguientes parámetros de protocolo: • Intentos de conexión • Intentos de transmisión • Tiempo de retardo de carácter • Retardo de acuse Si no está seleccionado es posible configurar valores para cada uno de estos parámetros.
Intentos de conexión	Número de intentos de conexión (valor predeterminado: 6 intentos de conexión) 1 a 255
Intentos de transmisión	Número de intentos de transmisión (valor predeterminado: 6 intentos de conexión) 1 a 255
Tiempo de retardo de carácter	Ajuste del tiempo de retardo de carácter (en función de la velocidad de transferencia de datos ajustada) (valor predeterminado: 220 ms) 1 ms a 65535 ms
Retardo de acuse	Ajuste del tiempo de retardo de acuse (en función de la velocidad de transferencia de datos ajustada) (valor predeterminado: 2000 ms cuando la comprobación de bloque está activada; 550 ms cuando la comprobación de bloque no está activada) 1 ms a 65535 ms

Nota

A excepción de la prioridad, los ajustes del protocolo deben ser los mismos para el módulo CM y para el interlocutor de la comunicación.

13.3.5 Instrucciones de comunicación punto a punto

13.3.5.1 Parámetros comunes de las instrucciones de comunicación punto a punto

Tabla 13-3 Parámetros de entrada comunes para las instrucciones PTP

Parámetro	Descripción
REQ	Muchas de las instrucciones PtP utilizan la entrada REQ para iniciar la operación en una transición de "low" a "high". El estado lógico de la entrada REQ debe ser "high" (TRUE) durante una ejecución de la instrucción. No obstante, la entrada REQ puede permanecer TRUE durante un tiempo cualquiera. La instrucción no inicia ninguna operación diferente hasta que no sea llamada con la entrada REQ puesta a FALSE, de manera que pueda inicializar el histórico de la entrada REQ. Esto es necesario para que la instrucción pueda detectar la transición de "low" a "high" para iniciar la siguiente operación. Al insertar una instrucción PtP en su programa, STEP 7 le solicitará identificar el DB de instancia. Utilice un DB unívoco para cada llamada de la instrucción PtP. Esto garantiza que cada instrucción procese correctamente la entradas, p. ej. REQ.
PORT	Se asigna una dirección de puerto durante la configuración del dispositivo de comunicación. Después de la configuración, es posible seleccionar un nombre simbólico predeterminado para el puerto en la lista desplegable auxiliar de parámetros. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes" de la tabla de variables PLC.
Resolución de tiempos de bit	Numerosos parámetros se definen en un número de tiempos de bit a la velocidad de transferencia configurada. Si el parámetro se define en tiempos de bit, puede ser independiente de la velocidad de transferencia. Todos los parámetros definidos en unidades de tiempos de bit pueden especificarse hasta un número máximo de 65535. No obstante, el tiempo máximo que puede medir el CM o la CB es ocho segundos.

Los parámetros de salida DONE, NDR, ERROR y STATUS de las instrucciones PtP ponen a disposición de las operaciones PtP el estado de ejecución finalizada.

Tabla 13-4 Parámetros de salida DONE, NDR, ERROR y STATUS

Parámetro	Tipo de datos	Valor predeterminado	Descripción
DONE	Bool	FALSE	Adopta el estado lógico TRUE durante una ejecución para indicar que la última petición se ha finalizado sin errores; de lo contrario, FALSE.
NDR	Bool	FALSE	Adopta el estado lógico TRUE durante una ejecución para indicar que la acción solicitada se ha finalizado sin errores y que se han recibido datos nuevos; de lo contrario, FALSE.
ERROR	Bool	FALSE	Adopta el estado lógico TRUE durante una ejecución para indicar que la última petición se ha finalizado con errores. El código de error aplicable aparece en STATUS; de lo contrario, FALSE.
STATUS	Word	0	Resultado del estado: • Si se activa el bit DONE o NDR, STATUS se pone a 0 o a un código de información. • Si se activa el bit ERROR, STATUS se pone a un código de error. • Si no se activa ninguno de estos bits, la instrucción devuelve resultados de estado que describen el estado actual de la función. STATUS conserva su valor durante la ejecución de la función.

Nota

Los parámetros DONE, NDR y ERROR solo están activados durante una ejecución. La lógica del programa debe guardar temporalmente los valores de estado de la salida en flip-flops de datos, de modo que sea posible detectar cambios de estado en posteriores ciclos del programa.

Tabla 13-5 Códigos de condición comunes

STATUS (W#16#....)	Descripción
0000	No hay error
7000	La función no está ocupada
7001	La función está ocupada con la primera llamada.
7002	La función está ocupada con las llamadas posteriores (sondeos tras la primera llamada).
8x3A	Puntero no permitido en el parámetro x
8070	Se está utilizando toda la memoria de instancia interna, hay demasiadas instrucciones simultáneas en curso
8080	Número de puerto no permitido.
8081	Timeout, error del módulo u otro error interno
8082	Ha fallado la parametrización porque se está parametrizando en segundo plano.
8083	Desbordamiento del búfer: El CM o la CB ha devuelto un mensaje recibido con una longitud superior a la que permite el parámetro de longitud.
8090	Error interno: Longitud de mensaje incorrecta, submódulo incorrecto o mensaje no permitido Póngase en contacto con el Customer Support.
8091	Error interno: Versión incorrecta en el mensaje de parametrización Póngase en contacto con el Customer Support.
8092	Error interno: Longitud de registro incorrecta en el mensaje de parametrización Póngase en contacto con el Customer Support.

Tabla 13-6 Clases de error comunes

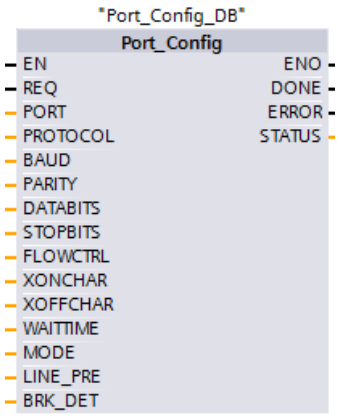
Descripción de la clase	Clases de error	Descripción
Configuración del puerto	16#81Ax	Permite definir errores comunes de configuración del puerto
Configuración de la transmisión	16#81Bx	Permite definir errores comunes de configuración de la transmisión
Configuración de la recepción	16#81Cx 16#82Cx	Permite definir errores comunes de configuración de la recepción
Tiempo de ejecución de la transmisión	16#81Dx	Permite definir errores comunes en tiempo de ejecución de la transmisión
Tiempo de ejecución de la recepción	16#81Ex	Permite definir errores comunes en tiempo de ejecución de la recepción
Procesamiento de señales	16#81Fx	Permite definir errores comunes en relación con el procesamiento de señales

Descripción de la clase	Clases de error	Descripción
Errores de puntero	16#8p01 a 16#8p51	Usado para errores del puntero ANY, siendo "p" el número de parámetro de la instrucción
Errores de protocolo integrados	16#848x 16#858x	Usado para errores de protocolo integrados

13.3.5.2 Port_Config (configurar los parámetros de comunicación dinámicamente)

Tabla 13-7 Instrucción Port_Config (configuración de puerto)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción Port_Config permite cambiar los parámetros de puerto (p. ej. la velocidad de transferencia) desde el programa.
		La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción Port_Config puede ejecutarse en el programa con el fin de modificar la configuración.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La CPU no guarda de forma permanente los valores ajustados con la instrucción Port_Config. La CPU restablece los parámetros ajustados en la configuración de dispositivos cuando cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Encontrará más información en Configurar los puertos de comunicación (Página 943) y Gestionar el control de flujo (Página 945).

Tabla 13-8 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de la configuración cuando se detecta un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
PROTOCOL	IN	UInt	0 - protocolo Freeport (valor predeterminado) 1. Protocolo 3964(R)

Parámetro y tipo		Tipo de datos	Descripción
BAUD	IN	UInt	Velocidad de transferencia del puerto (valor predeterminado: 6): 1 = 300 baudios, 2 = 600 baudios, 3 = 1200 baudios, 4 = 2400 baudios, 5 = 4800 baudios, 6 = 9600 baudios, 7 = 19200 baudios, 8 = 38400 baudios, 9 = 57600 baudios, 10 = 76800 baudios, 11 = 115200 baudios
PARITY	IN	UInt	Paridad del puerto (Valor predeterminado: 1): 1 = Sin paridad, 2 = Paridad par, 3 = Paridad impar, 4 = Paridad marca, 5 = Paridad espacio
DATABITS	IN	UInt	Bits por carácter (valor predeterminado: 1): 1 = 8 bits de datos, 2 = 7 bits de datos
STOPBITS	IN	UInt	Bits de parada (valor predeterminado: 1): 1 = 1 bit de parada, 2 = 2 bits de parada
FLOWCTRL*	IN	UInt	Control de flujo (valor predeterminado: 1): 1 = Sin control de flujo, 2 = XON/XOFF, 3 = RTS de hardware siempre ON, 4 = RTS de hardware conmutado
XONCHAR[1]	IN	Char	Determina el carácter que se utiliza como carácter XON. Generalmente, es un carácter DC1 (16#11). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 16#11)
XOFFCHAR[1]	IN	Char	Determina el carácter que se utiliza como carácter XOFF. Generalmente, es un carácter DC3 (16#13). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 16#13)
WAITTIME[1]	IN	UInt	Determina cuánto tiempo se debe esperar un carácter XON tras recibir un carácter XOFF o cuánto tiempo se debe esperar la señal CTS tras habilitar RTS (0 a 65535 ms). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 2000)
MODE[2]	IN	UInt	Especifica la selección del modo de operación del módulo. • 0 = dúplex (RS232) • 1 = dúplex (RS422) a cuatro hilos (punto a punto), transmisor siempre activado • 2 = dúplex (RS422) a cuatro hilos (maestro multipunto), transmisor siempre activado • 3 = dúplex (RS422) a cuatro hilos (esclavo multipunto), transmisor activado al enviar • 4 = semidúplex (RS485) a dos hilos
LINE_PRE	IN	UInt	Especifica la condición de "idle line" inactiva. Para los módulos RS422 y RS485 la condición de "idle line" se establece con la aplicación de un voltaje polarizado a las señales R(A) y R(B). Las selecciones siguientes son posibles: • 0 = Sin polarización (sin preajuste) (predeterminado) • 1 = Polarizada con R(A) > R(B) ≥ 0V; solo RS422 • 2 = Polarizada con R(B) > R(A) ≥ 0V; RS422 y RS485
BRK_DET	IN	UInt	Activa/desactiva la detección de rotura de hilo para la comunicación. Si se habilita la detección de rotura de hilo, el módulo indica un error en caso de que el cable de comunicación no esté conectado al módulo. En el modo punto a punto de RS422, la detección de rotura de hilo solo es posible si se usa el preajuste de la línea de recepción con polarización de forma que R(A) > R(B) ≥ 0V. • 0 = Sin detección de rotura de hilo (predeterminado) • 1 = Detección de rotura de hilo habilitada
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

[1] No aplicable si el protocolo = 1 (protocolo 3964(R))

[2] Solo son válidos los modos 0 y 1 cuando el protocolo = 1 (protocolo 3964(R)) en función de si el módulo CM es un RS232 o un RS422.

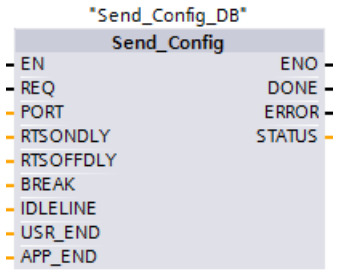
Tabla 13-9 Códigos de condición

STATUS (W#16#....)	Descripción
81A0	El protocolo indicado no existe.
81A1	La velocidad de transferencia indicada no existe.
81A2	La opción de paridad indicada no existe.
81A3	El número de bits de datos indicado no existe.
81A4	El número de bits de parada indicado no existe.
80A5	El tipo de control de flujo indicado no existe.
81A6	El tiempo de espera es 0 y el control de flujo está habilitado
81A7	XON y XOFF son valores no permitidos (p. ej. el mismo valor)
81A8	Error en el encabezado del bloque (por ejemplo, tipo de bloque erróneo o longitud errónea del bloque)
81A9	Reconfiguración denegada porque hay una configuración en curso
81AA	Modo de operación RS422/RS485 no válido
81AB	Preajuste no válido de la línea de recepción para la detección de la rotura
81AC	Manejo de rotura RS232 no válido
8280	Acuse negativo al leer del módulo
8281	Acuse negativo al escribir en el módulo
8282	Esclavo DP o módulo no disponible

13.3.5.3 Send_Config (configurar los parámetros de transmisión en serie dinámicamente)

Tabla 13-10 Instrucción Send_Config (configuración de transmisión)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción Send_Config permite configurar dinámicamente los parámetros de transmisión serie de un puerto de comunicación PtP. Todos los mensajes en cola de espera en un CM o una CB se rechazarán cuando se ejecute Send_Config.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción Send_Config puede ejecutarse en el programa con el fin de modificar la configuración.

La CPU no guarda de forma permanente los valores ajustados con la instrucción Send_Config. La CPU restablece los parámetros ajustados en la configuración de dispositivos cuando cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Véase Configurar los parámetros de transmisión (Página 946).

Tabla 13-11 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
RTSONDLY	IN	UInt	Número de milisegundos que debe esperarse tras la habilitación de RTS antes de cualquier transmisión de datos Tx. Este parámetro solo es válido si está habilitado el control de flujo por hardware. El rango válido es de 0 a 65535 ms. Un valor 0 desactiva la función. (Valor predeterminado: 0)
RTSOFFDLY	IN	UInt	Número de milisegundos que se debe esperar después de la transmisión de datos Tx antes de que se inhiba RTS: Este parámetro solo es válido si está habilitado el control de flujo por hardware. El rango válido es de 0 a 65535 ms. Un valor 0 desactiva la función. (Valor predeterminado: 0)
BREAK	IN	UInt	Este parámetro indica que se enviará una pausa tras el inicio de cada mensaje durante el número de tiempos de bit indicado. El máximo es 65535 tiempos de bit hasta ocho segundos como máximo. Un valor 0 desactiva la función. (Valor predeterminado: 12)

Parámetro y tipo		Tipo de datos	Descripción
IDLELINE	IN	UInt	Este parámetro indica que la línea permanecerá inactiva durante el número de tiempos de bit indicado hasta el inicio de cada mensaje. El máximo es 65535 tiempos de bit hasta ocho segundos como máximo. Un valor 0 desactiva la función. (Valor predeterminado: 0)
USR_END*	IN	STRING[2]	Especifica el número y los caracteres en el delimitador de fin. El delimitador de fin está integrado en el búfer de transmisión (solo caracteres) y marca el final del mensaje transmitido (los caracteres se transmiten hasta que se encuentra el delimitador de fin). El delimitador de fin se agrega al final del mensaje. • STRING[2,0,xx,yy] – El delimitador de fin no se usa (predeterminado) • STRING[2,1,xx,yy] – El delimitador de fin es un carácter individual • STRING[2,2,xx,yy] – El delimitador de fin está compuesto por dos caracteres USR_END o APP_END tienen que tener una longitud cero.
APP_END*	IN	STRING[5]	Especifica el número y los caracteres que deben agregarse al mensaje transmitido (solo se agregan los caracteres). STRING[5,0,aa,bb,cc,dd,ee] – El carácter de fin no se usa (predeterminado) • STRING[5,1,aa,bb,cc,dd,ee] – Transmite un carácter de fin • STRING[5,2,aa,bb,cc,dd,ee] – Transmite dos caracteres de fin • STRING[5,3,aa,bb,cc,dd,ee] – Transmite tres caracteres de fin • STRING[5,4,aa,bb,cc,dd,ee] – Transmite cuatro caracteres de fin • STRING[5,5,aa,bb,cc,dd,ee] – Transmite cinco caracteres de fin
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

* No apto para CM y CB 1241s; debe usarse una cadena vacía ("") para el parámetro.

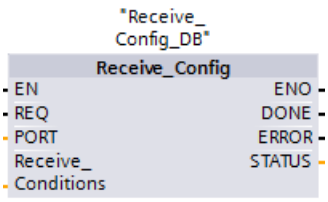
Tabla 13-12 Códigos de condición

STATUS (W#16#....)	Descripción
81B0	No se permite configurar una alarma de transmisión. Póngase en contacto con el Customer Support.
81B1	El tiempo de pausa excede el valor máximo permitido.
81B2	El tiempo de inactividad excede el valor máximo permitido.
81B3	Error en el encabezado del bloque, por ejemplo, tipo de bloque erróneo o longitud errónea del bloque
81B4	Reconfiguración denegada porque hay una configuración en curso
81B5	El número de delimitadores de fin especificado es mayor que dos o el número de caracteres de fin es mayor que cinco
81B6	Configuración de transmisión rechazada si se configura para protocolos de firmware integrados
8280	Acuse negativo al leer del módulo
8281	Acuse negativo al escribir en el módulo
8282	Esclavo DP o módulo no disponible

13.3.5.4 Receive_Config (configurar los parámetros de recepción en serie dinámicamente)

Tabla 13-13 Instrucción Receive_Config (configuración de recepción)

KOP / FUP	SCL	Descripción

	`"Receive_Config_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`Receive_Conditions:=_struct_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		La instrucción Receive_Config permite configurar dinámicamente los parámetros de recepción serie de un puerto de comunicación PtP. Esta instrucción configura las condiciones que indican el inicio y fin de un mensaje recibido. Todos los mensajes en cola de espera en un CM o una CB se rechazarán cuando se ejecute Receive_Config.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La configuración estática inicial del puerto de comunicación puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción Receive_Config puede ejecutarse en el programa con el fin de modificar la configuración.

La CPU no guarda de forma permanente los valores ajustados con la instrucción Receive_Config. La CPU restablece los parámetros ajustados en la configuración de dispositivos cuando cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Encontrará más información en el apartado "Configurar los parámetros de recepción (Página 947)".

Tabla 13-14 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
CONDITIONS	IN	CONDITIONS	La estructura de datos CONDITIONS indica las condiciones de inicio y fin del mensaje, como se describe a continuación.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

Condiciones de inicio de la instrucción Receive_P2P

La instrucción Receive_P2P utiliza la configuración indicada por la instrucción Receive_Config para determinar el inicio y fin de los mensajes de comunicación punto a punto. Las condiciones de inicio determinan el inicio de un mensaje. Una o más condiciones de inicio pueden determinar el inicio de un mensaje. Si se especifica más de una condición de inicio, todas las condiciones deberán cumplirse antes de iniciar el mensaje.

En el apartado "Configurar los parámetros de recepción (Página 947)" encontrará una descripción de las condiciones de inicio del mensaje.

Estructura de los tipos de datos del parámetro CONDITIONS, 1ª parte (condiciones de inicio)

Tabla 13-15 Estructura de CONDITIONS para condiciones de inicio

Parámetro y tipo		Tipo de datos	Descripción
STARTCOND	IN	UInt	Indica la condición de inicio (valor predeterminado: 1) • 01H - Carácter inicial • 02H - Cualquier carácter • 04H - Line Break • 08H - Idle Line • 10H - Secuencia 1 • 20H - Secuencia 2 • 40H - Secuencia 3 • 80H - Secuencia 4
IDLETIME	IN	UInt	Número de tiempos de bit necesario para el timeout por "idle line". (Valor predeterminado: 40). Se utiliza únicamente para una condición de "idle line". 0 a 65535
STARTCHAR	IN	Byte	Carácter de inicio utilizado con la condición de carácter de inicio. (Valor predeterminado: B#16#2)
STRSEQ1CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 1: (Valor predeterminado: B#16#0) Estos son los bits de habilitación de cada carácter de la secuencia de inicio. • 01H - Carácter 1 • 02H - Carácter 2 • 04H - Carácter 3 • 08H - Carácter 4 • 10H - Carácter 5 Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
STRSEQ1	IN	Char[5]	Caracteres de inicio de la secuencia 1 (5 caracteres). Valor predeterminado: 0
STRSEQ2CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 2. Valor predeterminado: B#16#0)
STRSEQ2	IN	Char[5]	Caracteres de inicio de la secuencia 2 (5 caracteres). Valor predeterminado: 0

Parámetro y tipo		Tipo de datos	Descripción
STRSEQ3CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 3. Valor predeterminado: B#16#0
STRSEQ3	IN	Char[5]	Caracteres de inicio de la secuencia 3 (5 caracteres). Valor predeterminado: 0
STRSEQ4CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 4. Valor predeterminado: B#16#0
STRSEQ4	IN	Char[5]	Caracteres de inicio de la secuencia 4 (5 caracteres), valor predeterminado: 0

Ejemplo

Considere el siguiente mensaje recibido en código hexadecimal: "**68** 10 aa **68** bb 10 aa 16" y las secuencias de inicio configuradas que muestra la tabla siguiente. Las secuencias de inicio se comienzan a evaluar al recibirse correctamente el primer carácter 68H. Tras recibirse correctamente el cuarto carácter (el segundo 68H), se cumple la condición de inicio 1. Una vez cumplidas las condiciones de inicio, se iniciará la evaluación de las condiciones de fin.

El procesamiento de la secuencia de inicio puede cancelarse debido a distintos errores de paridad, trama o tiempo excedido entre caracteres. Debido a estos errores, no se recibe el mensaje puesto que no se ha cumplido la condición de inicio.

Tabla 13-16 Condiciones de inicio

Condición de inicio	Primer carácter	Primer carácter +1	Primer carácter +2	Primer carácter +3	Primer carácter +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

Condiciones de fin de la instrucción Receive_P2P

Las condiciones de fin configuradas determinan el fin de un mensaje. La primera aparición de una o más condiciones de fin configuradas determina el fin de un mensaje. La sección "Condiciones de fin del mensaje" del apartado "Configurar los parámetros de recepción (Página 947)" describe las condiciones de fin que se pueden configurar en la instrucción Receive_Config.

Las condiciones de fin se pueden configurar tanto en las propiedades de la interfaz de comunicación en la configuración de dispositivos como desde la instrucción Receive_Config. Cada vez que la CPU realiza una transición de STOP a RUN, los parámetros de recepción (condiciones de inicio y fin) vuelven a los ajustes de la configuración de dispositivos. Si el programa de usuario de STEP 7 ejecuta la instrucción Receive_Config, los ajustes cambiarán a las condiciones de Receive_Config.

Estructura de los tipos de datos del parámetro CONDITIONS, 2ª parte (condiciones de fin)

Tabla 13-17 Estructura de CONDITIONS para condiciones de fin

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
ENDCOND	IN	UInt 0	Este parámetro define la condición de fin del mensaje: • 01H -Tiempo de respuesta • 02H -Tiempo del mensaje • 04H - Tiempo excedido entre caracteres • 08H - Longitud máxima • 10H - N + LEN + M • 20H - Secuencia
MAXLEN	IN	UInt 1	Longitud máxima del mensaje: Se utiliza solo si está seleccionada la condición de fin "Longitud máxima". 1 a 1024 bytes
N	IN	UInt 0	Posición de byte dentro del campo de longitud en el mensaje. Se utiliza solo con la condición de fin N + LEN + M. 1 a 1022 bytes
LENGTHSIZE	IN	UInt 0	Tamaño del campo de byte (1, 2 o 4 bytes). Se utiliza solo con la condición de fin N + LEN + M.
LENGTHM	IN	UInt 0	Indica el número de caracteres posteriores al campo de longitud que no se incluyen en el valor del campo de longitud. Se utiliza solo con la condición de fin N + LEN + M. 0 a 255 bytes
RCVTIME	IN	UInt 200	Indica cuánto tiempo debe esperarse hasta que se reciba el primer carácter. La recepción se terminará con un error si no se recibe correctamente ningún carácter dentro del tiempo indicado. Se utiliza solo con la condición de fin "Tiempo de respuesta". (de 0 a 65535 tiempos de bit con un máximo de 8 segundos) Este parámetro no es una condición de fin del mensaje dado que la evaluación finaliza cuando se recibe el primer carácter de una respuesta. Es una condición de fin solo en el sentido de que finaliza una operación de recepción debido a que no se recibe ninguna respuesta cuando se espera una. Se debe seleccionar una condición de fin distinta.
MSGTIME	IN	UInt 200	Indica cuánto tiempo debe esperarse hasta la recepción completa del mensaje una vez recibido el primer carácter. Este parámetro se utiliza solo si está seleccionada la condición de fin "Tiempo de mensaje excedido". (de 0 a 65535 milisegundos)
CHARGAP	IN	UInt 12	Indica el número de tiempos de bit entre caracteres. Si el número de tiempos de bit entre caracteres excede el valor especificado, se cumplirá la condición de fin. Se utiliza solo con la condición de fin "Tiempo excedido entre caracteres". (de 0 a 65535 tiempos de bit hasta 8 segundos máx.)

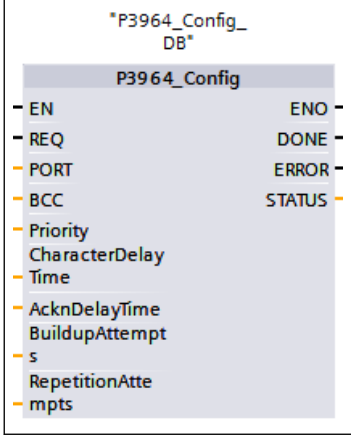
Parámetro	Tipo de parámetro	Tipo de datos	Descripción
ENDSEQ1CTL	IN	Byte B#16#0	Ignorar/comparar el control de cada carácter de la secuencia 1: Estos son los bits de habilitación de cada carácter de la secuencia de fin. El carácter 1 es el bit 0, el carácter 2 es el bit 1, ..., el carácter 5 es el bit 4. Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
ENDSEQ1	IN	Char[5] 0	Caracteres de inicio de la secuencia 1 (5 caracteres)

Tabla 13-18 Códigos de condición

STATUS (W#16#....)	Descripción
81C0	Se ha seleccionado una condición de inicio no permitida
81C1	Se ha seleccionado una condición de fin no permitida o no se ha seleccionado ninguna condición de fin
81C2	Hay una alarma de recepción habilitada, pero esto no es posible.
81C3	La condición de fin "Longitud máxima" está habilitada y la longitud máxima es 0 o > 1024.
81C4	La longitud calculada está habilitada y N es >= 1023.
81C5	La longitud calculada está habilitada y la longitud no es 1, 2 o 4.
81C6	La longitud calculada está habilitada y el valor M es > 255.
81C7	La longitud calculada está habilitada y es > 1024.
81C8	El timeout de respuesta está habilitado y es igual a cero.
81C9	El tiempo excedido entre caracteres está habilitado y es igual a cero.
81CA	El timeout por "idle line" está habilitado y es igual a cero.
81CB	La secuencia de fin está habilitada, pero todos los caracteres son irrelevantes.
81CC	La secuencia de inicio (una cualquiera de 4) está habilitada, pero todos los caracteres son irrelevantes.
81CD	Mensaje de recepción no válido, protección contra sobrescritura, error de selección
81CE	Mensaje de recepción no válido, manejo del búfer en la transición de STOP a RUN, error de selección
81CF	Error en el encabezado del bloque, por ejemplo, tipo de bloque erróneo o longitud errónea del bloque
8281	Acuse negativo al escribir en el módulo
8282	Esclavo DP o módulo no disponible
82C0	Reconfiguración denegada porque hay una configuración en curso
82C1	El valor especificado para el número de mensajes que el módulo puede respaldar es mayor que el valor máximo permitido.
82C2	Configuración de recepción rechazada si se configura para protocolos de firmware integrados
8351	Tipo de datos no permitido en este puntero Variant

13.3.5.5 P3964_Config (configurar el protocolo 3964(R))

Tabla 13-19 Instrucción P3964_Config (configurar el protocolo 3964(R))

KOP / FUP	SCL	Descripción
	"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_);	P3964_Config permite cambiar la prioridad y los parámetros de protocolo en tiempo de ejecución. La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción P3964_Config puede ejecutarse en el programa con el fin de modificar la configuración.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La CPU no guarda de forma permanente los valores ajustados con la instrucción P3964_Config. La CPU restablece los parámetros ajustados en la configuración de dispositivos tras desconectar y volver a conectar la alimentación. Encontrará más información en Configurar la prioridad de la comunicación y los parámetros de protocolo para 3964(R) (Página 957).

Tabla 13-20 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de la configuración cuando se detecta un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	UInt	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes de sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
BCC	IN	USInt	Activa/desactiva el uso de la comprobación de bloque • 0 = sin comprobación de bloque • 1 = con comprobación de bloque
Priority	IN	UInt	Selección de la prioridad • 0 = prioridad baja • 1 = prioridad alta La prioridad del CM debe ser la contraria que la prioridad del interlocutor de la comunicación.
CharacterDelay-Time	IN	UInt	Ajuste del tiempo de retardo de carácter (en función de la velocidad de transferencia de datos ajustada) (valor predeterminado: 220 ms) 1 ms a 65535 ms

Parámetro y tipo		Tipo de datos	Descripción
AcknDelayTime	IN	UInt	Ajuste del tiempo de retardo de acuse (en función de la velocidad de transferencia de datos ajustada) (valor predeterminado: 2000 ms) 1 ms a 65535 ms
BuildupAttempts	IN	UInt	Número de intentos de conexión (valor predeterminado: 6 intentos de conexión) 1 a 255
RepetitionAttempts	IN	UInt	Número de intentos de transmisión (valor predeterminado: 6 intentos de conexión) 1 a 255
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

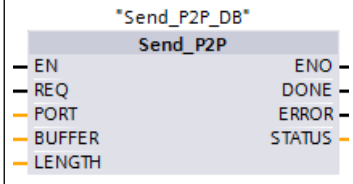
Tabla 13-21 Códigos de condición

STATUS (W#16#....)	Descripción
16#8380	Error de parametrización: valor no válido para "Tiempo de retardo de carácter".
16#8381	Error de parametrización: valor no válido para "Tiempo de respuesta excedido".
16#8382	Error de parametrización: valor no válido para "Prioridad".
16#8383	Error de parametrización: valor no válido para "Comprobación de bloque".
16#8384	Error de parametrización: valor no válido para "Intentos de conexión".
16#8385	Error de parametrización: valor no válido para "Intentos de transmisión".
16#8386	Error en tiempo de ejecución: número de intentos de conexión excedido
16#8387	Error en tiempo de ejecución: número de intentos de transmisión excedido
16#8388	Error en tiempo de ejecución: error en el "Carácter de comprobación de bloque" El valor calculado internamente para el carácter de comprobación de bloque no se corresponde con el carácter de comprobación de bloque recibido por el interlocutor al final de la conexión.
16#8389	Error en tiempo de ejecución: se ha recibido un carácter no válido mientras se esperaba que el búfer de recepción estuviera libre
16#838A	Error en tiempo de ejecución: error lógico durante la recepción. Después de recibir DLE se ha recibido un carácter adicional aleatorio (distinto de DLE o ETX).
16#838B	Error en tiempo de ejecución: tiempo de retardo de carácter excedido
16#838C	Error en tiempo de ejecución: se ha iniciado el tiempo de espera para que el búfer de recepción esté libre
16#838D	Error en tiempo de ejecución: la repetición del frame no se inicia en los 4 s posteriores a NAK
16#838E	Error en tiempo de ejecución: en modo de reposo se han recibido uno o varios caracteres (distintos de NAK o STX).
16#838F	Error en tiempo de ejecución: conflicto de inicialización; los dos interlocutores tienen ajustada una prioridad alta
16#8391	Error de parametrización: los datos de configuración 3964 se han rechazado porque está ajustado Freeport

13.3.5.6 Send_P2P (transmitir datos del búfer de transmisión)

Tabla 13-22 Instrucción Send_P2P (enviar datos punto a punto)

KOP / FUP	SCL	Descripción

	`"Send_P2P_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_word_in_,`
	`BUFFER:=_variant_in_,`
	`LENGTH:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		La instrucción Send_P2P inicia la transferencia de datos y transmite el búfer asignado a la interfaz de comunicación. El programa de la CPU se sigue ejecutando hasta que el CM o la CB envía los datos a la velocidad de transferencia asignada. Solo una operación de transmisión puede estar pendiente en un momento dado. El CM o la CB devuelve un error si una segunda instrucción Send_P2P se ejecuta mientras el CM o la CB está transmitiendo un mensaje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-23 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa la transmisión solicitada cuando se produce un flanco ascendente en esta entrada de habilitación de la transmisión. Esto inicia la transferencia del contenido del búfer a la interfaz de comunicación punto a punto. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de transmisión. (Valor predeterminado: 0) **Nota:** no se soportan los datos booleanos ni las matrices booleanas.
LENGTH	IN	UInt	Longitud de la trama transmitida en bytes (valor predeterminado: 0) Para transmitir una estructura compleja, utilice siempre la longitud 0. Si la longitud es 0, la instrucción transmite siempre toda la trama.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

El estado lógico de las salidas DONE y ERROR es FALSE durante la operación de transmisión. Una vez finalizada la operación de transmisión, la salida DONE o ERROR se pone a TRUE para mostrar el estado de la transmisión. Mientras que DONE o ERROR sea TRUE, la salida STATUS será válida.

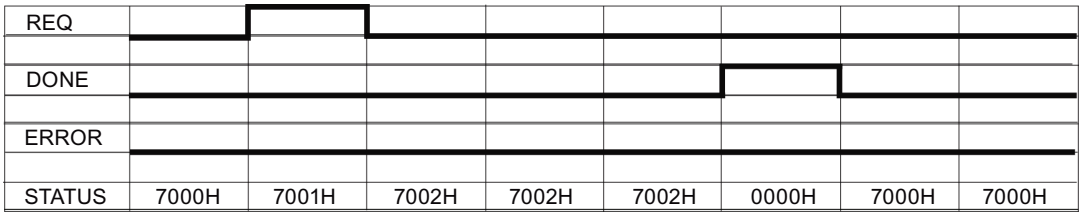
La instrucción devuelve el estado 16#7001 si la interfaz de comunicación acepta los datos transmitidos. Las ejecuciones posteriores de Send_P2P devuelven 16#7002 si el CM o la CB aún están ocupados transmitiendo datos. Una vez finalizada la operación de transmisión, el CM o la CB devuelve el estado de transmisión como 16#0000 (si no han ocurrido errores). Las

ejecuciones posteriores de Send_P2P con REQ puesto a "low" devuelven el estado 16#7000 (no ocupado).

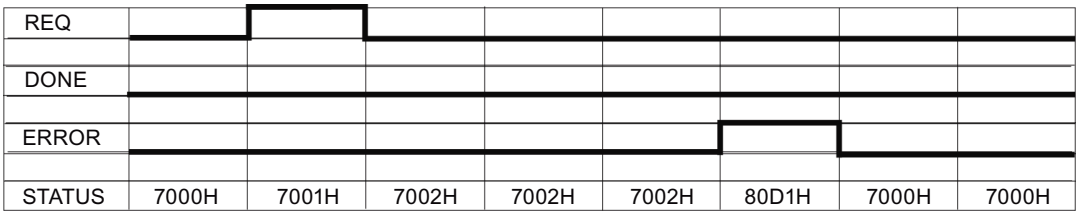
Los diagramas siguientes muestran la relación entre los valores de salida y REQ. Para esto se supone que la instrucción se llama periódicamente para comprobar el estado del proceso de transmisión. En el diagrama siguiente se supone que la instrucción se llama en cada ciclo (representado por los valores de STATUS).



El diagrama siguiente muestra cómo los parámetros DONE y STATUS son válidos solo durante un ciclo si la línea REQ se impulsa (durante un ciclo) para iniciar la operación de transmisión.



El diagrama siguiente muestra la relación de los parámetros DONE, ERROR y STATUS cuando se presenta un error.



Los valores DONE, ERROR y STATUS solo son válidos hasta que se ejecute nuevamente Send_P2P con el mismo DB de instancia.

Tabla 13-24 Códigos de condición

STATUS (W#16#....)	Descripción
81D0	Nueva petición estando activo el transmisor
81D1	Transmisión cancelada debido a que no se ha detectado ningún CTS durante el tiempo de espera
81D2	Transmisión cancelada debido a que no se ha recibido DSR del dispositivo DCE
81D3	Transmisión cancelada debido a desbordamiento de la cola de espera (transmisión de más de 1024 bytes)
81D5	Señal de polarización inversa (condición de rotura de hilo)
81D6	Solicitud de transmisión rechazada porque no se ha encontrado el delimitador de fin en el búfer de transmisión
81D7	Error interno / error de sincronización entre FB y CM

STATUS (W#16#....)	Descripción
81D8	Intento de transmisión rechazado porque el puerto no ha sido configurado
81DF	El CM ha reseteado la interfaz hacia el FB por una de las razones siguientes • El módulo se ha reiniciado (conexión y desconexión) • La CPU ha alcanzado un punto de parada • El módulo ha sido reparametrizado En cada caso, el módulo indica este código en el parámetro de estado. El módulo pone estado y error a cero después del primer registro recibido para SEND_P2P.
8281	Acuse negativo al escribir en el módulo
8282	Esclavo DP o módulo no disponible
8301	ID de sintaxis ilegal en el puntero ANY
8322	Error de longitud de rango cuando se lee un parámetro
8324	Error de rango cuando se lee un parámetro
8328	Error de alineación cuando se lee un parámetro
8332	El parámetro contiene un número de DB que es mayor que el número más alto permitido (error de número de DB).
833A	El DB para el parámetro BUFFER no existe.

Nota

Ajuste de la longitud máxima del registro para la comunicación Profibus

Al utilizar un módulo maestro Profibus CM1243-5 para controlar un dispositivo Profibus ET 200SP o ET 200MP que utilice un módulo punto a punto RS232, RS422 o RS485, es necesario ajustar explícitamente la variable del bloque de datos "max_record_len" a 240, como se describe a continuación:

Ajustar "max_record_len" en el DB de instancia (por ejemplo, "Send_P2P_DB".max_record_len) a 240 tras ejecutar cualquier instrucción de configuración como, por ejemplo, Port_Config, Send_Config o Receive_Config.

Solo es necesario asignar max_record_len si se utiliza la comunicación Profibus; la comunicación Profinet ya utiliza un valor max_record_len válido.

Interacción de los parámetros LENGTH y BUFFER

El tamaño mínimo de los datos que pueden transmitirse con la instrucción SEND_P2P es un byte. El parámetro BUFFER determina el tamaño de los datos por transmitir. No puede utilizarse el tipo de datos Bool ni matrices de Bool para el parámetro BUFFER.

El parámetro LENGTH se puede poner siempre a 0 para garantizar que SEND_P2P envíe la estructura completa de datos representada por el parámetro BUFFER. Si solo desea enviar una

parte de una estructura de datos del parámetro BUFFER , LENGTH se puede ajustar como se indica a continuación:

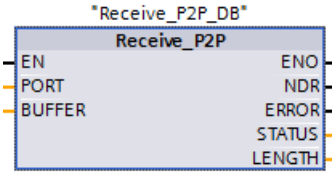
Tabla 13-25 Parámetros LENGTH y BUFFER

LENGTH	BUFFER	Descripción
= 0	No utilizado	Los datos completos se envían según se define en el parámetro BUFFER. No es necesario especificar el número de bytes transmitidos si LENGTH = 0.
> 0	Tipo de datos simple	El valor de LENGTH debe contener el contaje de bytes de este tipo de datos. Por ejemplo, para un valor Word, el valor LENGTH debe ser dos. Para Dword o Real, el valor LENGTH debe ser cuatro. De lo contrario, no se transferirá nada y se devolverá el error 8088H.
	Estructura	El valor LENGTH puede contener un contaje de bytes menor que la longitud total de bytes de la estructura, en cuyo caso la instrucción solo envía los primeros n bytes de la estructura desde BUFFER,, siendo n = LENGTH. Dado que la organización interna de bytes de una estructura no siempre se puede determinar, es posible que se obtengan resultados inesperados. En este caso, utilice un LENGTH de 0 para enviar la estructura completa.
	Matriz	El valor LENGTH debe contener un contaje de bytes que sea menor o igual a la longitud total de bytes de la matriz y que sea múltiplo del contaje de bytes del elemento de datos. Por ejemplo, el parámetro LENGTH para una matriz de Words debe ser un múltiplo de dos y para una matriz de Reals, un múltiplo de cuatro. Cuando se especifica LENGTH, la instrucción transfiere el número de elementos de matriz que corresponden al valor de LENGTH en bytes. Si su BUFFER, por ejemplo, contiene una matriz de 15 Dwords (60 bytes totales) y se especifica un valor de 20 para LENGTH, entonces las primeras cinco Dwords de la matriz se transfieren. El valor de LENGTH debe ser un múltiplo del contaje de bytes del elemento de datos. De lo contrario, STATUS = 8088H, ERROR = 1 y la transmisión no se efectuará.
	String	El parámetro LENGTH contiene el número de caracteres que debe transmitirse. Solo se transmiten los caracteres del String. Los bytes de longitud máxima y real del String no se transmiten.

13.3.5.7 Receive_P2P (habilitar recepción de mensajes)

Tabla 13-26 Instrucción Receive_P2P (recepción punto a punto)

KOP / FUP	SCL	Descripción

	`"Receive_P2P_DB"(`
	`PORT:=_word_in_,`
	`BUFFER:=_variant_in_,`
	`NDR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`LENGTH=>_uint_out_);`
		Receive_P2P comprueba si se han recibido mensajes en el CM o la CB. Si hay un mensaje disponible, se transfiere del CM o de la CB a la CPU. Un error devuelve el valor de STATUS apropiado.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-27 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de recepción. Este búfer debería ser lo suficientemente grande para recibir el mensaje de longitud máxima. No se soportan los datos booleanos ni las matrices booleanas. (Valor predeterminado: 0)
NDR	OUT	Bool	TRUE durante una ejecución cuando se dispone de datos nuevos y la operación ha finalizado sin errores.
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la operación con un error.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)
LENGTH	OUT	UInt	Longitud del mensaje devuelto (en bytes) (valor predeterminado: 0)

El valor de STATUS es válido si NDR o ERROR es TRUE. El valor STATUS proporciona el motivo para finalizar la recepción en el CM o la CB. Por lo general, este valor es positivo, lo cual indica que la recepción se realizó correctamente y que el proceso de recepción finalizó normalmente. Si el valor de STATUS es negativo (es decir, si está activado el bit más significativo del valor hexadecimal), esto indica que la recepción se ha finalizado debido a un error p. ej. de paridad, trama o desbordamiento.

Cada interfaz de comunicación PtP puede almacenar como máximo 1024 bytes en un búfer. Esto podría ser un mensaje grande o varios mensajes pequeños. Si más de un mensaje está disponible en el CM o en la CB, la instrucción Receive_P2P devolverá el mensaje más antiguo disponible. La siguiente ejecución de la instrucción Receive_P2P devolverá el siguiente mensaje más antiguo disponible.

Tabla 13-28 Códigos de condición

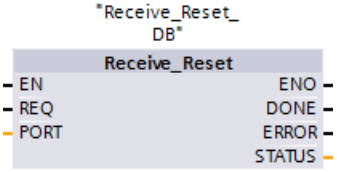
STATUS (W#16#...)	Descripción
0000	No hay ningún búfer disponible
0094	Mensaje terminado porque se ha recibido la longitud máxima de caracteres
0095	Mensaje terminado debido a timeout del mensaje
0096	Mensaje terminado porque se ha excedido el tiempo entre caracteres
0097	Mensaje terminado debido a timeout de respuesta
0098	Mensaje terminado porque se ha cumplido la condición de longitud "N+LEN+M"
0099	Mensaje terminado porque se ha cumplido la condición de secuencia de fin
8085	El parámetro LONGITUD tiene un valor de 0 o es superior a 1KB.
8088	El parámetro LENGTH o la longitud recibida es más larga que el área especificada en BUFFER o la longitud recibida es más larga que el área especificada en BUFFER.
8090	Mensaje de configuración incorrecto, longitud de mensaje errónea, submódulo erróneo, mensaje no permitido
81E0	Mensaje terminado debido a que el búfer de recepción está lleno
81E1	Mensaje terminado debido a un error de paridad
81E2	Mensaje terminado debido a un error de trama

STATUS (W#16#...)	Descripción
81E3	Mensaje terminado debido a un error de desbordamiento
81E4	Mensaje terminado porque la longitud calculada excede el tamaño del búfer
81E5	Señal de polarización inversa (condición de rotura de hilo)
81E6	La cola del mensaje está llena. Este error se notifica sin datos. Si esto ocurre, el módulo conmuta entre una transferencia de datos sin errores y este error.
81E7	Error interno, error de sincronización entre la instrucción y el CM: se activa cuando se detecta un error de secuencia
81E8	Mensaje terminado, el tiempo de espera entre caracteres ha excedido antes de que se hayan cumplido los criterios de final de mensaje.
81E9	Se ha detectado un error Modbus CRC (solo usado por módulos que soportan la generación/comprobación CRC para el protocolo Modbus)
81EA	El telegrama Modbus es demasiado corto (solo usado por módulos que soportan la generación/comprobación CRC para el protocolo Modbus)
81EB	Mensaje terminado porque se ha excedido el tamaño máximo del mensaje
8201	ID de sintaxis ilegal en el puntero ANY
8223	Error de longitud de rango cuando se escribe un parámetro. El parámetro está completa o parcialmente fuera del rango de una dirección o la longitud de un rango de bits no es un múltiplo de 8 con un puntero ANY.
8225	Error de rango cuando se escribe un parámetro. El parámetro está en un rango no permitido para la función de sistema.
8229	Error de alineación cuando se escribe un parámetro. El parámetro referenciado está en una dirección de bit que no es igual a 0.
8230	El parámetro está en un DB global de solo lectura.
8231	El parámetro está en un DB de instancia de solo lectura.
8232	El parámetro contiene un número de DB que es mayor que el número de bloque más alto permitido (error de número de DB).
823A	El DB para el parámetro BUFFER no existe.
8280	Acuse negativo al leer del módulo
8282	Esclavo DP o módulo no disponible

13.3.5.8 Receive_Reset (borrar el búfer de recepción)

Tabla 13-29 Instrucción Receive_Reset (inicializar receptor)

KOP / FUP	SCL	Descripción

	`"Receive_Reset_DB"(`
	`  REQ:=_bool_in_,`
	`  PORT:=_word_in_,`
	`  DONE=>_bool_out_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_);`
		La instrucción Receive_Reset borra los búferes de recepción en el CM o la CB.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

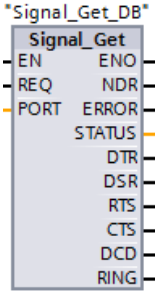
Tabla 13-30 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa la inicialización del receptor cuando se produce un flanco ascendente en esta entrada de habilitación (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
DONE	OUT	Bool	Si es TRUE durante un ciclo, indica que la última petición se ha finalizado sin errores.
ERROR	OUT	Bool	Si es TRUE, indica que la última petición se ha finalizado con errores. Además, si esta salida es TRUE, la salida STATUS contendrá los códigos de error respectivos.
STATUS	OUT	Word	Código de error (valor predeterminado: 0)

13.3.5.9 Signal_Get (consultar señales RS-232)

Tabla 13-31 Instrucción Signal_Get (consultar señales RS-232)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción Signal_Get lee los estados actuales de las señales de comunicación RS232.
		Esta función es válida solo para el CM RS232.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-32 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Los valores de las señales RS232 se consultan cuando se produce un flanco ascendente en esta entrada (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
NDR	OUT	Bool	TRUE durante un ciclo cuando se dispone de datos nuevos y la operación ha finalizado sin errores
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la operación con un error

Parámetro y tipo		Tipo de datos	Descripción
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)
DTR	OUT	Bool	Terminal de datos disponible, módulo listo (salida). Valor predeterminado: False
DSR	OUT	Bool	Equipo de datos listo, interlocutor listo (entrada). Valor predeterminado: False
RTS	OUT	Bool	Petición de transmitir, módulo listo para transmitir (salida). Valor predeterminado: False
CTS	OUT	Bool	Listo para transmitir; el interlocutor puede recibir datos (entrada). Valor predeterminado: False
DCD	OUT	Bool	Detección de portadora de datos, nivel de señal de recepción (siempre False, no se soporta)
RING	OUT	Bool	Indicación de timbre, señalización de una llamada entrante (siempre False, no se soporta)

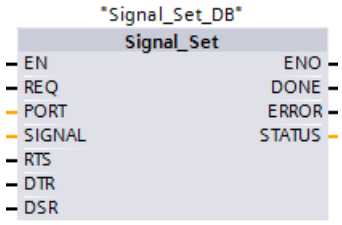
Tabla 13-33 Códigos de condición

STATUS (W#16#....)	Descripción
81F0	El CM o la CB es un RS485; no hay señales disponibles
81F4	Error en el encabezado del bloque, por ejemplo, tipo de bloque erróneo o longitud errónea del bloque
8280	Acuse negativo al leer del módulo
8282	Esclavo DP o módulo no disponible

13.3.5.10 Signal_Set (activar señales RS-232)

Tabla 13-34 Instrucción Signal_Set (activar señales RS-232)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción Signal_Set activa los estados de las señales de comunicación RS232.
		Esta función es válida solo para el CM RS232.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-35 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Inicia la activación de las señales RS232 cuando se produce un flanco ascendente en esta entrada (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
SIGNAL	IN	Byte	Selecciona las señales que deben activarse: (se permiten varias). Valor predeterminado: 0 • 01H = Activar RTS • 02H = Activar DTR • 04H = Activar DSR
RTS	IN	Bool	Petición de transmitir, módulo listo para transmitir el valor que debe ajustarse (True o False), valor predeterminado: False
DTR	IN	Bool	Terminal de datos disponible, módulo listo para transmitir el valor que debe ajustarse (True o False). Valor predeterminado: False
DSR	IN	Bool	Equipo de datos listo (aplicable solo a las interfaces DCE) (no se utiliza).
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

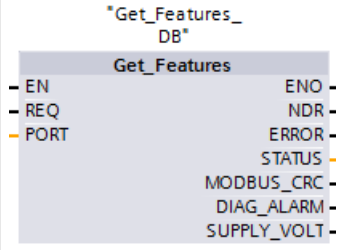
Tabla 13-36 Códigos de condición

STATUS (W#16#....)	Descripción
81F0	El CM o la CB es un RS485; no pueden activarse señales
81F1	Las señales no pueden activarse debido al control de flujo por hardware
81F2	Imposible activar DSR porque el módulo es DTE
81F3	Imposible activar DTR porque el módulo es DCE
81F4	Error en el encabezado del bloque, por ejemplo, tipo de bloque erróneo o longitud errónea del bloque
8280	Acuse negativo al leer del módulo
8281	Acuse negativo al escribir en el módulo
8282	Esclavo DP o módulo no disponible

13.3.5.11 Get_Features

Tabla 13-37 Instrucción Get_Features (consultar funciones avanzadas)

KOP / FUP	SCL	Descripción

	`"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);`
		Get_Features lee las funciones avanzadas de un módulo.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-38 Tipos de datos de los parámetros

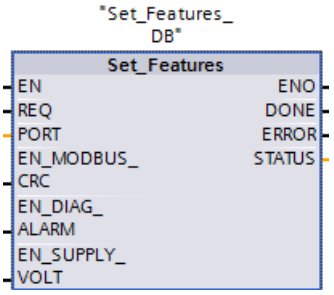
Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
NDR	OUT	Bool	Indica que los datos nuevos están preparados.
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)
MODBUS_CRC*	OUT	Bool	Generación y comprobación MODBUS CRC
DIAG_ALARM*	OUT	Bool	Generación de alarma de diagnóstico
SUPPLY_VOLT*	OUT	Bool	El diagnóstico de falta de tensión de alimentación L+ está disponible

*Get_Features devuelve TRUE (1) si la función está disponible, FALSE (0) si la función no está disponible

13.3.5.12 Set_Features

Tabla 13-39 Instrucción Set_Features (ajustar funciones avanzadas)

KOP / FUP	SCL	Descripción

	`"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 ajusta las funciones avanzadas que soporta un modulo.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-40 Tipos de datos de los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
EN_MODBUS_CRC	IN	Bool	Activar la generación y comprobación de MODBUS CRC: • 0: cálculo CRC desconectado (predeterminado) • 1: cálculo CRC conectado Nota: solo soportan este parámetros los CMs V2.1, las CPUs V4.1 con CBs y los módulos CM PtP para periferia descentralizada.
EN_DIAG_ALARM	IN	Bool	Activar la generación de alarma de diagnóstico: • 0: alarma de diagnóstico desconectada • 1: alarma de diagnóstico conectada (predeterminado)
EN_SUPPLY_VOLT	IN	Bool	Activar el diagnóstico de falta de tensión de alimentación L+: • 0: diagnóstico de tensión de alimentación desactivado (predeterminado) • 1: diagnóstico de tensión de alimentación activado
DONE	OUT	Bool	Indica que se ha realizado el ajuste de funciones
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

13.3.6 Programar la comunicación PtP

STEP 7 ofrece instrucciones avanzadas que permiten al programa de usuario establecer comunicaciones punto a punto utilizando un protocolo diseñado e implementado en el programa de usuario. Estas instrucciones corresponden a dos categorías:

- Instrucciones de configuración
- Instrucciones de comunicación

Instrucciones de configuración

Para que el programa de usuario pueda intervenir en la comunicación PtP, es preciso configurar el puerto de comunicación y los parámetros para enviar y recibir datos.

El puerto y los mensajes pueden configurarse para todo CM o toda CB en la configuración de dispositivos o mediante estas instrucciones en el programa de usuario:

- Port_Config (Página 961)
- Send_Config (Página 964)
- Receive_Config (Página 966)

Instrucciones de comunicación

Las instrucciones de comunicación PtP permiten al programa de usuario enviar y recibir mensajes a/de las interfaces de comunicación. Encontrará más información sobre cómo transferir datos con estas instrucciones en el apartado Coherencia de datos (Página 187).

Todas las funciones PtP funcionan de forma asíncrona. El programa de usuario puede utilizar una arquitectura de sondeo para determinar el estado de las transmisiones y recepciones. Send_P2P y Receive_P2P pueden ejecutarse simultáneamente. Según sea necesario, los módulos de comunicación y la placa de comunicación depositan los mensajes de transmisión y recepción en un búfer cuyo tamaño máximo es 1024 bytes.

Los CMs y la CB envían y reciben mensajes a/de los dispositivos punto a punto reales. El protocolo de mensajes se encuentra en un búfer que se recibe de o se envía a un puerto de comunicación específico. El búfer y el puerto son parámetros de las instrucciones de envío y recepción.

- Send_P2P (Página 973)
- Receive_P2P (Página 976)

Se dispone de instrucciones adicionales que permiten inicializar el búfer de recepción, así como consultar y activar determinadas señales RS232:

- Receive_Reset (Página 978)
- Signal_Get (Página 979)
- Signal_Set (Página 980)

13.3.6.1 Arquitectura de sondeo

El programa de usuario de STEP 7 debe llamar las instrucciones de comunicación punto a punto de S7-1200 de forma cíclica/periódica para comprobar los mensajes recibidos. Un sondeo de la transmisión notifica al programa de usuario el fin de la transmisión.

Arquitectura de sondeo: maestro

La secuencia típica de un maestro es la siguiente:

1. Una instrucción Send_P2P (Página 973) inicia una transmisión al CM o la CB.
2. La instrucción Send_P2P se ejecuta en los ciclos posteriores para determinar el progreso de la transmisión.
3. Cuando la instrucción Send_P2P indica que ha finalizado la transmisión, el programa de usuario se prepara para recibir la respuesta.
4. La instrucción Receive_P2P (Página 976) se ejecuta repetidamente para comprobar si hay una respuesta. Una vez que el CM o la CB haya recibido un mensaje de respuesta, la instrucción Receive_P2P copia la respuesta en la CPU e indicará que se han recibido datos nuevos.
5. El programa de usuario puede procesar la respuesta.
6. Vaya al paso 1 y repita el ciclo.

Arquitectura de sondeo: esclavo

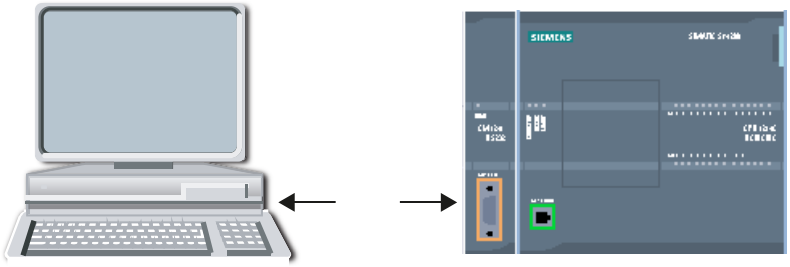
La secuencia típica de un esclavo es la siguiente:

1. El programa de usuario ejecuta la instrucción Receive_P2P en cada ciclo.
2. Una vez que el CM o la CB ha recibido una petición, la instrucción Receive_P2P indica que los nuevos datos están listos y la petición se copia en la CPU.
3. El programa de usuario procesa la petición y genera una respuesta.
4. Utilice una instrucción Send_P2P para enviar la respuesta al maestro.
5. Ejecute repetidamente Send_P2P para asegurarse de que se realiza la transmisión.
6. Vaya al paso 1 y repita el ciclo.

El esclavo debe llamar Receive_P2P con la suficiente frecuencia para recibir una transmisión del maestro antes de que se produzca un timeout del maestro a la espera de una respuesta. Para realizar esta tarea, el programa de usuario puede llamar RCV_PTP desde un OB de ciclo, previendo un tiempo de ciclo suficiente para recibir una transmisión del maestro antes de que transcurra el periodo de timeout. Si el tiempo de ciclo del OB se ajusta para que pueda ejecutarse dos veces dentro del periodo de timeout del maestro, el programa de usuario puede recibir todas las transmisiones sin pérdidas.

13.3.7 Ejemplo: Comunicación punto a punto

En este ejemplo, una CPU S7-1200 se comunica con un PC con emulador de terminal a través de un módulo CM 1241 RS232. La configuración punto a punto y el programa de STEP 7 muestran en este ejemplo cómo la CPU puede recibir un mensaje del PC y enviar el mensaje de vuelta al PC.



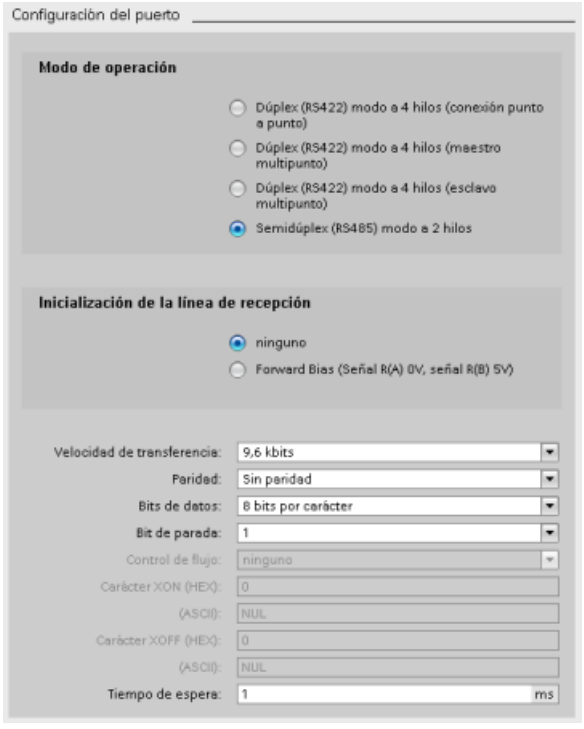
Hay que conectar la interfaz de comunicación del módulo CM 1241 RS232 a la interfaz RS232 del PC, que por lo general es COM1. Debido a que estos dos puertos son equipos terminales de datos (Data Terminal Equipment DTE), hay que intercambiar los pines de recepción y transmisión (2 y 3) al conectar ambos puertos. Esto se puede realizar aplicando uno de los métodos siguientes:

- Utilice un adaptador de módem NULO para intercambiar conjuntamente los pines 2 y 3 con un cable RS232 estándar.
- Utilice un cable de módem NULO que ya tenga intercambiados los pines 2 y 3. Generalmente, un cable de módem NULO se puede reconocer porque tiene en sus dos extremos un conector hembra sub-D de 9 pines.

13.3.7.1 Configurar los módulos de comunicación

El CM 1241 se puede configurar desde la configuración de dispositivos de STEP 7 o con las instrucciones del programa de usuario. Este ejemplo utiliza la variante de la configuración de dispositivos.

- Configuración del puerto: Haga clic en el puerto de comunicación del módulo CM desde la configuración de dispositivos y configure el puerto como se indica a continuación:

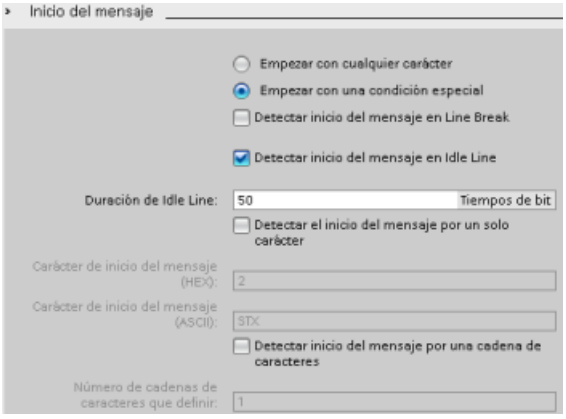


Nota

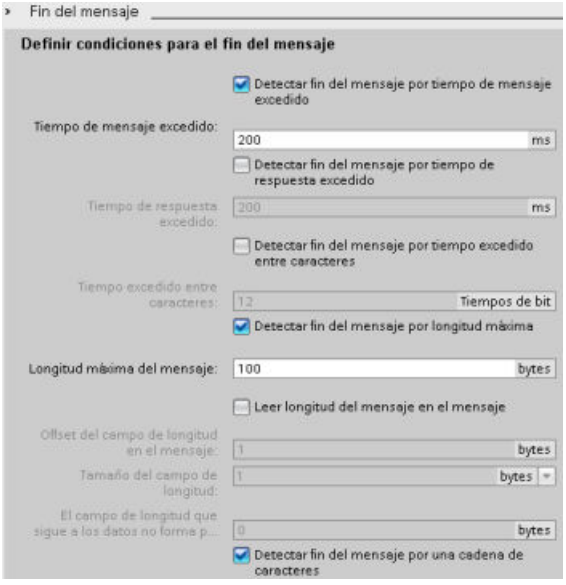
Los ajustes para la configuración del "Modo de operación" y "Inicialización de la línea de recepción" solo son aplicables al módulo CM 1241 (RS422/RS485). Los demás módulos CM 1241 no tienen estos ajustes para la configuración del puerto. Consulte Configurar RS422 y RS485 (Página 989).

- Configuración de la transferencia de mensajes: Acepte la configuración de la transferencia de mensajes predeterminada. No debe enviarse una pausa al inicio del programa.

- Configuración de inicio de recepción de mensajes: Configure el CM 1241 para que inicie la recepción de un mensaje cuando la línea de comunicación esté inactiva durante al menos 50 tiempos de bit (aprox. 5 milisegundos a 9600 baudios = 50 * 1/9600):



- Configuración de fin de recepción de mensajes: configure el CM 1241 para que finalice un mensaje cuando haya recibido como máximo 100 bytes o un carácter de salto de línea (10 decimales o un hexadecimal). La secuencia de fin permite hasta cinco caracteres de fin en secuencia. El quinto carácter en la secuencia es el carácter de salto de línea. Los cuatro caracteres de fin precedentes son caracteres "irrelevantes" o no seleccionados. El CM 1241 no compara los caracteres "irrelevantes" pero busca un carácter de salto de línea precedido de cero o más caracteres "irrelevantes" para indicar el fin del mensaje.





13.3.7.2 Estados operativos de RS422 y RS485

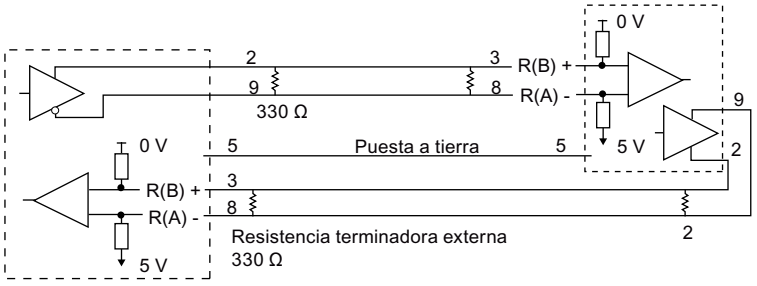
Configurar RS422

Para el modo RS422, hay tres modos de operación según la configuración de la red. Seleccione uno de estos modos en función de los dispositivos de la red. Según la selección que se realice en Inicialización de la línea de recepción, se podrían dar los casos que se indican a continuación.

- Modo dúplex (RS422) de cuatro hilos (conexión punto a punto): seleccione esta opción si hay dos dispositivos en la red. Inicialización de la línea de recepción:
  - Seleccione Ninguno si suministra la polarización y la terminación (caso 3).
  - Seleccione Polarización directa para utilizar la polarización y terminación internas (caso 2).
  - Seleccione Polarización inversa para utilizar la polarización y la terminación internas y habilitar la detección de rotura de cable para ambos dispositivos (caso 1).
- Modo dúplex (RS422) de cuatro hilos (maestro multipunto): Seleccione esta opción para el dispositivo maestro cuando tenga una red con un maestro y varios esclavos. Inicialización de la línea de recepción:
  - Seleccione Ninguno si suministra la polarización y la terminación (caso 3).
  - Seleccione Polarización directa para utilizar la polarización y terminación internas (caso 2).
  - La detección de rotura de cable no se puede realizar en este modo.
- Modo dúplex (RS422) de cuatro hilos (esclavo multipunto): Seleccione esta opción para todos los dispositivos esclavos cuando tenga una red con un maestro y varios esclavos. Inicialización de la línea de recepción:
  - Seleccione Ninguno si suministra la polarización y la terminación (caso 3).
  - Seleccione Polarización directa para utilizar la polarización y terminación internas (caso 2).
  - Seleccione Polarización inversa para utilizar la polarización y la terminación internas y habilitar la detección de rotura de cable para los esclavos (caso 1).

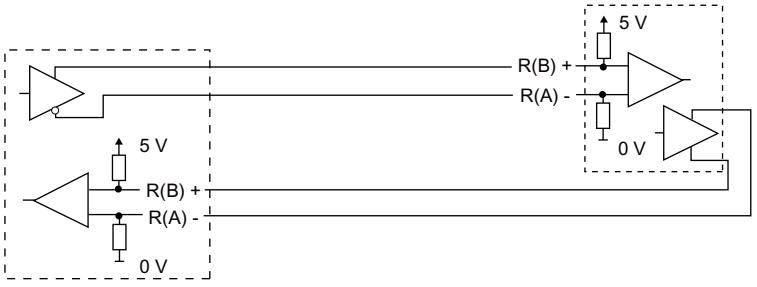
Caso 1: RS422 con detección de rotura de cable

- Modo de operación: RS422
- Inicialización de la línea de recepción: Polarización inversa (polarizada con R(A) > R(B) > 0V)
- Rotura de cable: Detección de rotura de cable habilitada (transmisor siempre activo)



Caso 2: RS422 sin detección de rotura de cable, polarización directa

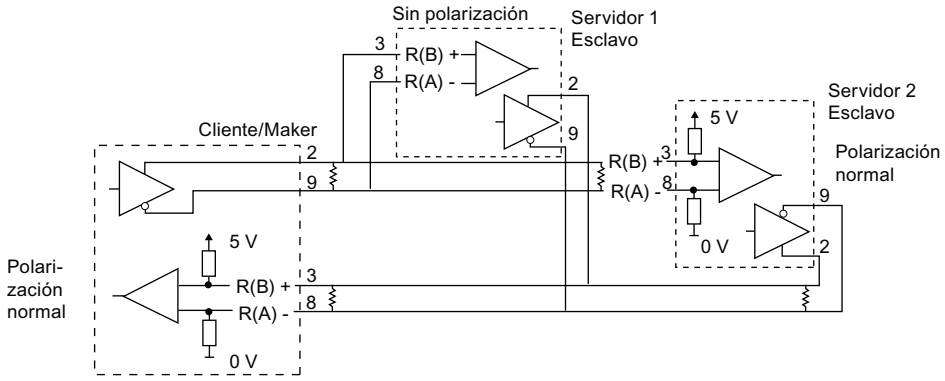
- Modo de operación: RS422
- Inicialización de la línea de recepción: Polarización directa (polarizada con R(B) > R(A) > 0 V)
- Rotura de cable: Sin detección de rotura de cable (transmisor habilitado solo durante la transmisión)



Caso 3: RS422: Sin detección de rotura de cable, sin polarización

- Modo de operación: RS422
- Inicialización de la línea de recepción: Sin polarización
- Rotura de cable: Sin detección de rotura de cable (transmisor habilitado solo durante la transmisión)

La polarización y la terminación las añade el usuario en los nodos finales de la red.



Configurar RS485

Para el modo RS485, solo hay un estado operativo. Según la selección que se realice en Inicialización de la línea de recepción, se podrían dar los casos que se indican a continuación.

- Modo semidúplex (RS485) de dos hilos. Inicialización de la línea de recepción:
  - Seleccione Ninguno si suministra la polarización y la terminación (caso 5).
  - Seleccione Polarización directa para utilizar la polarización y terminación internas (caso 4).

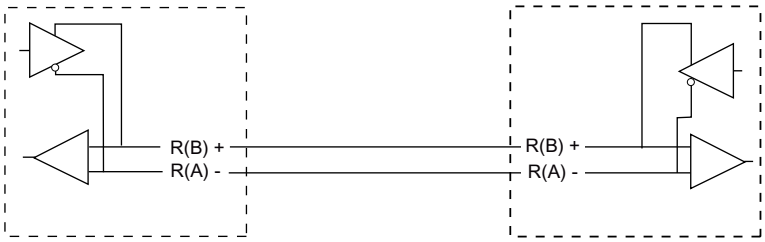
Caso 4: RS485: Polarización directa

- Modo de operación: RS485
- Inicialización de la línea de recepción: Polarización directa (polarizada con R(B) > R(A) > 0 V)



Caso 5: RS485: Sin polarización (polarización externa)

- Modo de operación: RS485
- Inicialización de la línea de recepción: Sin polarización (polarización externa necesaria)

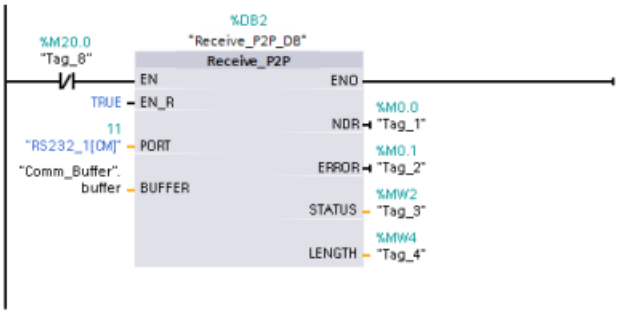


13.3.7.3 Programar el programa de STEP 7

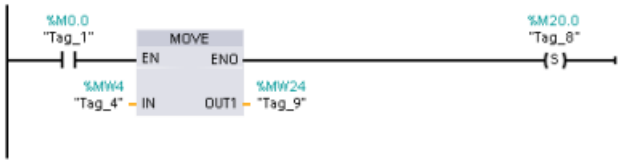
El programa de ejemplo utiliza un bloque de datos globales para el búfer de comunicación, una instrucción RCV_PTP (Página 1122) para recibir datos del emulador de terminal y una instrucción SEND_PTP (Página 1120) para enviar de regreso el búfer al emulador de terminal. Para programar el ejemplo, agregue la configuración del bloque de datos y el bloque de programa principal OB1 como se indica a continuación.

Bloque de datos global "Comm_Buffer": cree un bloque de datos global (DB) y asígnele el nombre "Comm_Buffer". Cree un valor en el bloque de datos llamado "buffer" con un tipo de datos de "matriz [0 .. 99] de byte".

Segmento 1: habilite la instrucción RCV_PTP siempre que SEND_PTP no esté activo. Tag_8 en MW20.0 indica cuando el envío ha finalizado en la red 4 y cuando el módulo de comunicación está por lo tanto listo para recibir un mensaje.

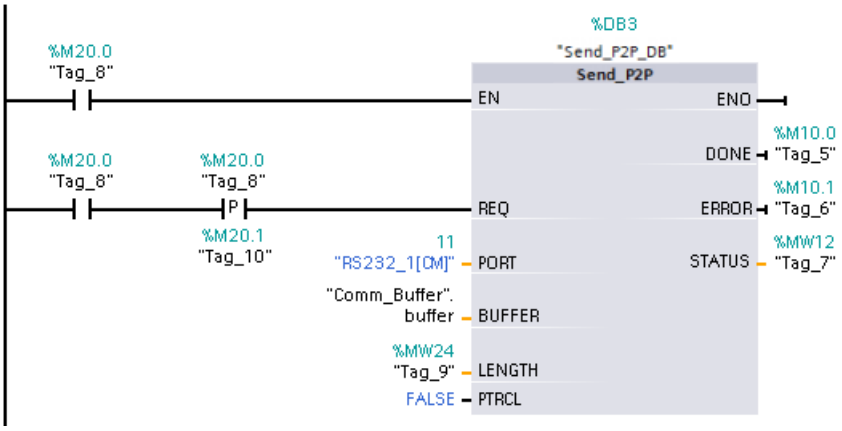


Segmento 2: utilice el valor NDR (Tag_1 en M0.0) activado por la instrucción RCV_PTP para hacer una copia del número de bytes recibidos y para activar una marca (Tag_8 en M20.0) con el fin de lanzar la instrucción SEND_PTP.

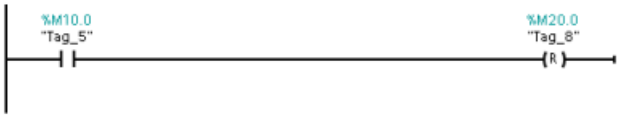


Segmento 3: habilite la instrucción SEND_PTP cuando la marca M20.0 esté activada. Utilice también esta marca para poner la entrada REQ en TRUE durante un ciclo. La entrada REQ le notifica a la instrucción SEND_PTP que debe transmitirse una nueva petición. La entrada REQ únicamente debe ponerse a TRUE durante una ejecución de SEND_PTP. La instrucción SEND_PTP se ejecuta en cada ciclo hasta que finalice la transmisión. La transmisión finaliza cuando el último

byte del mensaje ha sido transmitido desde el CM 1241. Cuando la transmisión ha finalizado, la salida DONE (Tag_5 en M10.0) se pone a TRUE durante una ejecución de SEND_PTP.



Segmento 4: vigile la salida DONE de SEND_PTP y desactive la marca de transmisión (Tag_8 en M20.0) cuando la transmisión haya finalizado. Cuando se desactiva la marca de transmisión, la instrucción RCV_PTP en el segmento 1 es habilitada para recibir el siguiente mensaje.



13.3.7.4 Configurar el emulador de terminal

Hay que configurar el emulador de terminal para soportar el programa de ejemplo. Se puede utilizar prácticamente cualquier tipo de emulador de terminal en su PC, p. ej. el HyperTerminal. Asegúrese de que el emulador de terminal esté en modo desconectado antes de editar los ajustes como se indica a continuación:

1. Ajuste el emulador de terminal para que utilice el puerto RS232 del PC (normalmente COM1).
2. Configure el puerto para 9600 baudios, 8 bits de datos, sin paridad (ninguna), 1 bit de parada y sin control de flujo.
3. Cambie los ajustes del emulador de terminal para emular un terminal ANSI.
4. Configure el emulador de terminal ASCII para enviar un salto de línea después de cada línea (después de que el usuario pulse la tecla Intro).
5. Active el eco de los caracteres escritos localmente para que el emulador de terminal visualice lo que se ha escrito.

13.3.7.5 Ejecutar el programa de ejemplo

Para poner en práctica el programa de ejemplo, proceda del siguiente modo:

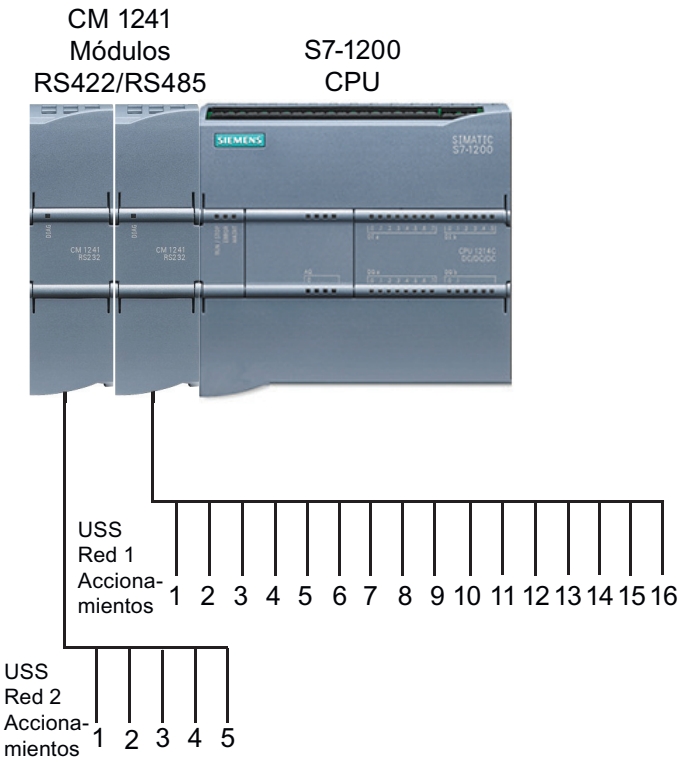
1. Cargue el programa STEP 7 en la CPU y asegúrese de que la misma esté en modo RUN.
2. Haga clic en el botón "conectar" en el emulador de terminal para transferir los cambios en la configuración y abrir una sesión de terminal en el CM 1241.
3. Escriba caracteres en el PC y pulse Intro.

El emulador de terminal envía los caracteres al CM 1241 y a la CPU. Luego, el programa de la CPU envía los caracteres de regreso al emulador de terminal.

13.4 Comunicación de interfaz serie universal (USS)

Las instrucciones USS controlan el funcionamiento de accionamientos motorizados que soportan el protocolo interfaz serie universal (USS). Las instrucciones USS se pueden utilizar para comunicarse con varios accionamientos a través de conexiones RS485 con módulos de comunicación CM 1241 RS485 o una placa de comunicación CB 1241 RS485. Pueden instalarse hasta tres módulos CM 1241 RS422/RS485 y una placa CB 1241 RS485 en una CPU S7-1200. Cada puerto RS485 puede operar hasta 16 accionamientos.

El protocolo USS utiliza una red maestro-esclavo para una comunicación vía bus serie. El maestro utiliza un parámetro de dirección para enviar un mensaje a un esclavo seleccionado. Un esclavo jamás puede transmitir sin primero haber recibido una petición correspondiente. No es posible una transferencia directa de mensajes entre los distintos esclavos. La comunicación USS funciona en modo semidúplex. La siguiente figura de USS muestra un diagrama de red para una aplicación con accionamientos a modo de ejemplo.



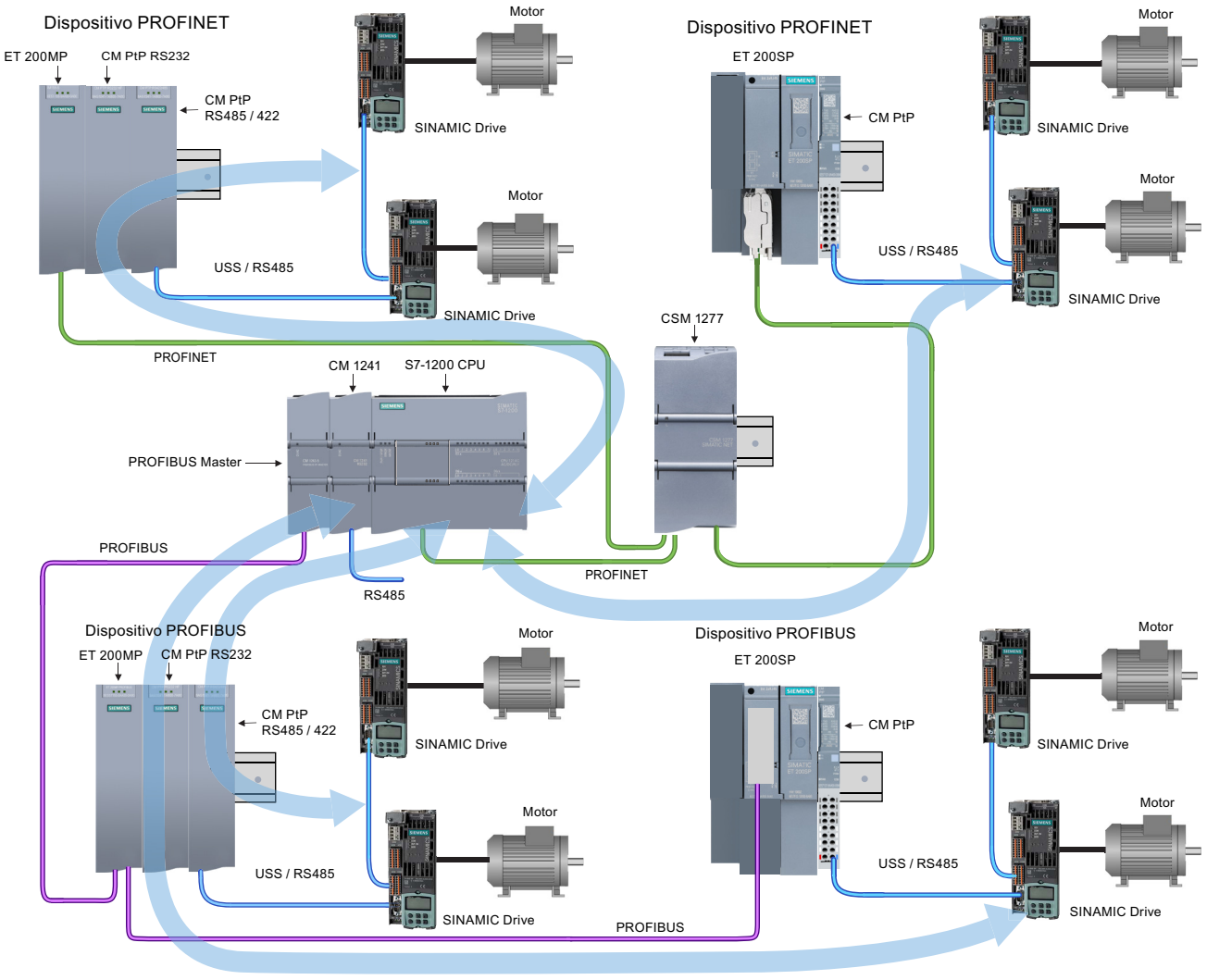
Comunicaciones USS mediante PROFIBUS o PROFINET

A partir de la versión V4.1 de la CPU S7-1200 junto con STEP 7 V13 SP1, la CPU amplía la capacidad de USS para usar un rack de periferia descentralizada PROFINET o PROFIBUS para comunicarse con varios dispositivos (lectores RFID, dispositivos GPS y otros):

- PROFINET (Página 594): la interfaz Ethernet de la CPU S7-1200 se conecta a un módulo de interfaz PROFINET. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.
- PROFIBUS (Página 777): un módulo de comunicación PROFIBUS se inserta en el lado izquierdo del rack con la CPU S7-1200. El módulo de comunicación PROFIBUS se conecta al rack que contiene un módulo de interfaz PROFIBUS. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.

Por este motivo, S7-1200 soporta dos juegos de instrucciones PtP:

- Instrucciones Legacy USS (Página 1131): estas instrucciones USS son anteriores a la versión V4.0 de S7-1200 y solo funcionan con la comunicación serie si se utiliza un módulo de comunicación CM 1241 o una Communication Board CB 1241.
- Instrucciones USS (Página 1000): Estas instrucciones ofrecen toda la funcionalidad de las instrucciones Legacy más la capacidad de conectarse a la periferia descentralizada de PROFINET y PROFIBUS. Las instrucciones USS permiten configurar la comunicación entre los módulos de comunicación PtP en el rack de periferia descentralizada y los dispositivos PtP. Los módulos CM 1241 del S7-1200 deben tener como mínimo la versión de firmware V2.1 para poder utilizar estas instrucciones USS.



La flecha azul indica el flujo de comunicación bidireccional entre dispositivos.

Nota

Con la versión V4.1 de S7-1200 se pueden emplear las instrucciones punto a punto para todos los tipos de comunicación punto a punto: serie, serie mediante PROFINET y serie mediante PROFIBUS. STEP 7 ofrece las instrucciones Legacy punto a punto solo por su compatibilidad con programas existentes. Las instrucciones Legacy siguen funcionando con todas las CPUs S7-1200. No es necesario convertir programas anteriores de un juego de instrucciones a otro.

13.4.1 Seleccionar la versión de las instrucciones USS

En STEP 7 están disponibles dos versiones de las instrucciones USS:

- La versión 2.0 (instrucciones Legacy) estaba disponible inicialmente en STEP 7 Basic/ Professional V13.
- La versión 2.1 y superiores está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No es posible utilizar las dos versiones de las instrucciones con el mismo módulo; en cambio, dos módulos distintos sí pueden utilizar diferentes versiones de las instrucciones.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



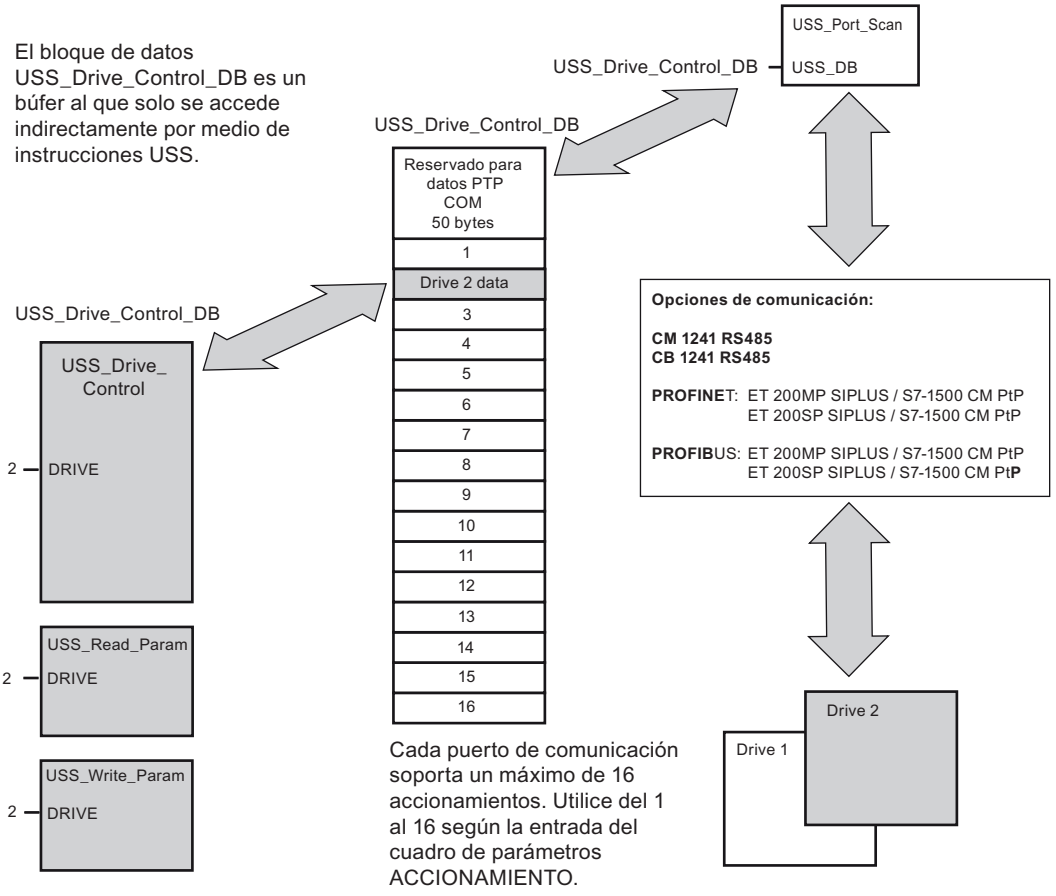
Para cambiar la versión de las instrucciones USS, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción USS en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción USS seleccionada. La nueva instancia del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción USS en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione una instancia del FB o la FC USS del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción USS.

13.4.2 Requisitos para utilizar el protocolo USS

Las cuatro instrucciones USS utilizan dos bloques de función (FB) y dos funciones (FC) que soportan el protocolo USS. Un bloque de datos de instancia (DB) USS_Port_Scan se utiliza para cada red USS. El bloque de datos de instancia USS_Port_Scan contiene un almacenamiento temporal y búferes para todos los accionamientos en la red USS. Las instrucciones USS comparten la información en este bloque de datos.



Todos los accionamientos (máx. 16) conectados a un solo puerto RS485 pertenecen a una misma red USS. Todos los accionamientos conectados a otro puerto RS485 pertenecen a diferentes redes USS. Cada red USS se gestiona utilizando un bloque de datos unívoco. Todas las instrucciones asociadas a una red USS deben compartir dicho bloque de datos. Esto incluye todas las instrucciones USS_Drive_Control, USS_Port_Scan, USS_Read_Param, y USS_Write_Param utilizadas para controlar todos los accionamientos de una red USS.

La instrucción USS_Drive_Control es un bloque de función (FB). Cuando la instrucción USS_Drive_Control se coloca en el editor de programas, es preciso indicar qué DB debe asignarse a ese FB en el cuadro de diálogo "Opciones de llamada". Si se trata de la primera instrucción USS_Drive_Control de este programa para esta red USS, es posible aceptar la asignación de DB predeterminada (o cambiar el nombre a discreción). Entonces se crea el nuevo DB. No obstante, si no es la primera instrucción USS_Drive_Control de este canal, es preciso utilizar la lista desplegable del cuadro de diálogo "Opciones de llamada" para seleccionar el nombre de DB que fue asignado previamente a esta red USS.

La instrucción USS_Port_Scan es un bloque de función (FB) que gestiona la comunicación real entre la CPU y los accionamientos vía el puerto de comunicación punto a punto (PtP) RS485. Cada llamada a este FB gestiona una comunicación con un accionamiento. El programa debe llamar este FB lo suficientemente rápido para impedir que se exceda el tiempo de comunicación por parte de los accionamientos. Este FB puede llamarse desde un OB de ciclo de programa principal o desde cualquier OB de alarma.

Las instrucciones USS_Read_Param y USS_Write_Param son funciones (FC). A estas FCs no se les asigna ningún DB cuando se colocan en el editor. En cambio, hay que asignar la referencia de DB apropiada a la entrada "USS_DB" de estas instrucciones. Haga doble clic en el campo de parámetro. A continuación, haga clic en el símbolo de ayuda de parámetros para ver los nombres de DB disponibles.

Por lo general, el FB USS_Port_Scan debe llamarse desde un OB de alarma cíclica. El tiempo de ciclo del OB de alarma cíclica debe ajustarse a aprox. la mitad del intervalo mínimo de llamada (p. ej., una comunicación de 1200 baudios debe utilizar un tiempo de ciclo de 350 ms o menos).

El FB USS_Drive_Control permite al programa acceder a un accionamiento determinado en la red USS. Sus entradas y salidas representan el estado y los controles del accionamiento. Si la red comprende 16 accionamientos, el programa debe tener como mínimo 16 llamadas de USS_Drive_Control, es decir, una para cada accionamiento. Estos bloques deben llamarse a la velocidad necesaria para controlar el funcionamiento del accionamiento.

El FB USS_Drive_Control solo se puede llamar desde un OB de ciclo de programa principal.

⚠ PRECAUCIÓN

Factores que se deben tener en cuenta al llamar las instrucciones USS de OB

USS_Drive_Control, USS_Read_Param y USS_Write_Param deben llamarse únicamente desde un OB de ciclo de programa principal. USS_Port_Scan FB puede llamarse desde un OB cualquiera (generalmente desde un OB de alarma cíclica).

No utilice las instrucciones USS_Drive_Control, USS_Read_Param y USS_Write_Param en un OB de prioridad más alta que la instrucción USS_Port_Scan correspondiente. Por ejemplo, no inserte USS_Port_Scan en el OB principal ni USS_Read_Param en un OB de alarma cíclica. Si no se impide la interrupción de la ejecución de USS_Port_Scan podrían producirse errores inesperados, que podrían provocar lesiones corporales.

Las FCs USS_Read_Param y USS_Write_Param leen y escriben los parámetros operativos del accionamiento remoto. Estos parámetros controlan el funcionamiento interno del accionamiento. Estos parámetros se definen en el manual del accionamiento. El programa puede contener un número cualquiera de estas funciones. No obstante, solo una petición de lectura o escritura puede estar activa en un accionamiento en un momento determinado. Las FCs USS_Read_Param y USS_Write_Param solo pueden llamarse desde un OB de ciclo de programa principal.

Calcular el tiempo necesario para la comunicación con el accionamiento

La comunicación con el accionamiento es asíncrona al ciclo del S7-1200. Por lo general, pueden transcurrir varios ciclos del S7-1200 antes de que finalice una transacción de comunicación con un accionamiento.

El intervalo de USS_Port_Scan es el tiempo necesario para una transacción con un accionamiento. La tabla siguiente muestra el intervalo de USS_Port_Scan mínimo para cada

velocidad de transferencia de comunicación. Si el FB USS_Port_Scan se llama más frecuentemente que el intervalo de USS_Port_Scan, no se incrementará el número de transacciones. El intervalo de timeout del accionamiento es el tiempo disponible para una transacción si, debido a errores de comunicación, se requieren 3 intentos para finalizar la transacción. De forma predeterminada, la librería del protocolo USS realiza automáticamente 2 reintentos por transacción.

Tabla 13-41 Calcular los requisitos de tiempo

Velocidad de transferencia	Intervalo mínimo calculado para la llamada de USS_Port_Scan (milisegundos)	Intervalo de timeout por accionamiento (milisegundos)
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 Instrucciones USS

13.4.3.1 USS_Port_Scan (procesar comunicación vía red USS)

Tabla 13-42 Instrucción USS_Port_Scan

KOP / FUP	SCL	Descripción

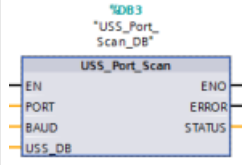
	`USS_Port_Scan(`
	`  PORT:=_uint_in_,`
	`  BAUD:=_dint_in_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	`  USS_DB:=_fbtref_inout_);`
		La instrucción USS_Port_Scan gestiona la comunicación en una red USS.

Tabla 13-43 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
PORT	IN	Port	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
BAUD	IN	DInt	Velocidad de transferencia utilizada para la comunicación USS.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_Drive_Control en el programa.

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	Si es TRUE (verdadero), esta salida indica que ha ocurrido un error y la salida STATUS es válida.
STATUS	OUT	Word	El valor de estado de la petición indica el resultado del ciclo o de la inicialización. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

Generalmente, el programa contiene solo una instrucción USS_Port_Scan por cada puerto de comunicación PtP. Cada llamada de este bloque de función (FB) gestiona una transferencia hacia o desde un accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo DB de instancia.

El programa debe ejecutar la instrucción USS_Port_Scan con suficiente frecuencia para impedir timeouts del accionamiento. USS_Port_Scan se llama generalmente desde un OB de alarma cíclica para impedir timeouts del accionamiento y para que las actualizaciones de datos USS más recientes estén disponibles para las llamadas de USS_Drive_Control.

Nota

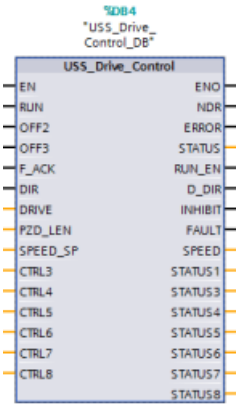
Cuando se utiliza la librería del protocolo USS y la instrucción USS_Port_Scan con una CB 1241, hay que poner la variable del bloque de datos LINE_PRE a 0 (sin estado inicial). El valor predeterminado 2 para la variable del bloque de datos LINE_PRE hace que la instrucción USS_Port_Scan devuelva un valor erróneo de 16#81AB. La variable del bloque de datos LINE_PRE se encuentra en el bloque de datos asociado a la instrucción USS_Port_Scan (normalmente se denomina USS_Port_Scan_DB).

Asegúrese de que el valor de arranque de LINE_PRE se cambia a 0 (cero).

13.4.3.2 USS_Drive_Control (intercambio de datos con el accionamiento)

Tabla 13-44 Instrucción USS_Drive_Control

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción USS_Drive_Control intercambia datos con el accionamiento creando peticiones e interpretando las respuestas del accionamiento. Para cada accionamiento debe utilizarse un bloque de función propio. No obstante, todas las funciones USS asociadas con una red USS y el puerto de comunicación PtP deben utilizar el mismo bloque de datos de instancia. Es preciso crear el nombre del DB cuando se inserta la primera instrucción USS_Drive_Control. Este DB creado al insertar la instrucción por primera vez se debe referenciar a continuación.
		STEP 7 crea el DB automáticamente al insertar la instrucción.

[1] KOP y FUP: Haga clic en el lado inferior del cuadro para ampliarlo y ver todos los parámetros. Los parámetros que aparecen atenuados son opcionales y no se requiere una asignación de parámetros.

Tabla 13-45 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
RUN	IN	Bool	Bit de arranque del accionamiento: Si es TRUE (verdadera), esta entrada habilita el accionamiento para que funcione a la velocidad predeterminada. Cuando RUN pasa a False mientras funciona un accionamiento, el motor se desacelerará hasta pararse. Este comportamiento difiere del de una caída de alimentación (OFF2) y del de un frenado del motor (OFF3).
OFF2	IN	Bool	Bit de parada eléctrica: Si es FALSE (falso), este bit hace que el accionamiento marche en inercia hasta parar sin frenar.
OFF3	IN	Bool	Bit de parada rápida: Si es FALSE (falso), este bit origina una parada rápida frenando el accionamiento, en vez de permitir que marche en inercia hasta parar.

Parámetro y tipo		Tipo de datos	Descripción
F_ACK	IN	Bool	Bit de acuse de fallo: Este bit se activa para resetear el bit de fallo en un accionamiento. El bit se activa tras haberse solucionado el fallo para indicar al accionamiento que no tiene que seguir notificando el fallo anterior.
DIR	IN	Bool	Control de sentido del accionamiento: Este bit se activa para indicar que el sentido es hacia delante (SPEED_SP positiva).
DRIVE	IN	USInt	Dirección del accionamiento: Esta entrada es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PZD_LEN	IN	USInt	Longitud de palabra: Este es el número de palabras de los datos PZD. Los valores válidos son 2, 4, 6 u 8 palabras. El valor predeterminado es 2.
SPEED_SP	IN	Real	Consigna de velocidad: Esta es la velocidad del accionamiento expresada como porcentaje de la frecuencia configurada. Un valor positivo indica el sentido hacia delante (si DIR es TRUE). El rango válido está comprendido entre 200,00 y -200,00.
CTRL3	IN	Word	Palabra de control 3: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL4	IN	Word	Palabra de control 4: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL5	IN	Word	Palabra de control 5: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL6	IN	Word	Palabra de control 6: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL7	IN	Word	Palabra de control 7: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL8	IN	Word	Palabra de control 8: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
NDR	OUT	Bool	Nuevos datos listos: Si es TRUE (verdadero), el bit indica que las salidas contienen datos de una petición de comunicación nueva.
ERROR	OUT	Bool	Ha ocurrido un error: Si es TRUE (verdadero), indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_Port_Scan.
STATUS	OUT	Word	El valor de estado de la petición indica el resultado del ciclo. Esta no es una palabra de estado devuelta del accionamiento.
RUN_EN	OUT	Bool	Funcionamiento habilitado: Este bit indica si está funcionando el accionamiento.
D_DIR	OUT	Bool	Sentido del accionamiento: Este bit indica si el accionamiento está funcionando hacia delante.
INHIBIT	OUT	Bool	Accionamiento inhibido: Este bit indica el estado del bit de inhibición del accionamiento.
FAULT	OUT	Bool	Fallo del accionamiento: Este bit indica que el accionamiento ha registrado un fallo. El usuario debe corregir el problema y activar el bit F_ACK para borrar este bit si está activado.

Parámetro y tipo		Tipo de datos	Descripción
SPEED	OUT	Real	Velocidad actual del accionamiento (valor escalado de la palabra de estado 2 del accionamiento) Valor de velocidad del accionamiento expresado como porcentaje de la velocidad configurada.
STATUS1	OUT	Word	Palabra de estado 1 del accionamiento: Este valor contiene bits de estado fijos de un accionamiento.
STATUS3	OUT	Word	Palabra de estado 3 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS4	OUT	Word	Palabra de estado 4 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS5	OUT	Word	Palabra de estado 5 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS6	OUT	Word	Palabra de estado 6 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS7	OUT	Word	Palabra de estado 7 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS8	OUT	Word	Palabra de estado 8 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.

Cuando la instrucción USS_Drive_Control se ejecuta por primera vez, el accionamiento que indica la dirección USS (parámetro DRIVE) se inicializa en el DB de instancia. Después de esta inicialización, las ejecuciones siguientes de USS_Port_Scan pueden iniciar la comunicación con el accionamiento en este número de accionamiento.

Si se modifica el número del accionamiento, la CPU debe cambiar de STOP a RUN con objeto de inicializar el DB de instancia. Los parámetros de entrada se configuran en el búfer de mensajes USS TX y las salidas se leen de un búfer de respuesta válido "anterior" (si existe). Durante la ejecución de USS_Drive_Control no se transmiten datos. Los accionamientos se comunican cuando USS_Port_Scan se ha ejecutado. USS_Drive_Control configura únicamente los mensajes que deben enviarse e interpreta los datos que puedan haberse recibido de una petición anterior.

El sentido de rotación del accionamiento se puede controlar utilizando la entrada DIR (Bool) o el signo (positivo o negativo) con la entrada SPEED_SP (Real). La tabla siguiente indica cómo interactúan estas entradas para determinar el sentido del accionamiento, suponiendo que el motor está cableado para la rotación adelante.

Tabla 13-46 Interacción de los parámetros SPEED_SP y DIR

SPEED_SP	DIR	Sentido de rotación del accionamiento
Valor > 0	0	Atrás
Valor > 0	1	Adelante
Valor < 0	0	Adelante
Valor < 0	1	Atrás

13.4.3.3 USS_Read_Param (leer datos del accionamiento)

Tabla 13-47 Instrucción USS_Read_Param

KOP / FUP	SCL	Descripción

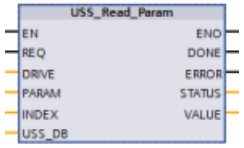
	`USS_Read_Param(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`VALUE=>_variant_out_,`
	`USS_DB:=_fbtref_inout_);`
		La instrucción USS_Read_Param lee un parámetro del accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo bloque de datos. USS_Read_Param debe llamarse desde un OB de ciclo de programa principal.

Tabla 13-48 Tipos de datos para los parámetros

Tipo de parámetro		Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), REQ indica que se desea una nueva petición de lectura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: DRIVE es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PARAM	IN	UInt	Número de parámetro: PARAM designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. En algunos accionamientos, el byte más significativo puede acceder a valores PARAM superiores a 2047. Encontrará más información sobre cómo acceder a un rango ampliado en el manual del accionamiento.
INDEX	IN	UInt	Índice de parámetro: INDEX designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_Drive_Control en el programa.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Valor del parámetro que se ha leído y que es válido solo si el bit DONE es TRUE (verdadero).
DONE[1]	OUT	Bool	Si es TRUE (verdadero), indica que la salida VALUE contiene el valor del parámetro de lectura solicitado anteriormente. Este bit se activa cuando USS_Drive_Control detecta los datos de respuesta de lectura del accionamiento. Este bit se desactiva en los casos siguientes: los datos de respuesta se solicitan mediante otra consulta USS_Read_Param o bien en la segunda de las dos llamadas siguientes de USS_Drive_Control.

Tipo de parámetro		Tipo de datos	Descripción
ERROR	OUT	Bool	Ha ocurrido un error: Si es TRUE (verdadero), ERROR indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_Port_Scan.
STATUS	OUT	Word	STATUS indica el resultado de la petición de lectura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

[1] El bit DONE indica que se han leído datos válidos del accionamiento de motor referenciado y que se han transmitido a la CPU. No indica que la librería USS sea capaz de leer inmediatamente otros parámetros. Hay que enviar una petición PKW vacía al accionamiento del motor, la cual debe ser acusada por la instrucción antes de que el canal de parámetros para el accionamiento en cuestión vuelva a estar disponible. Si se llama inmediatamente una FC USS_Read_Param o USS_Write_Param para el accionamiento de motor especificado se producirá un error "0x818A".

13.4.3.4 USS_Write_Param (modificar datos del accionamiento)

Nota

Operaciones de escritura EEPROM (para la EEPROM dentro de un accionamiento USS)

No utilice excesivamente la operación de escritura permanente en EEPROM. Minimice el número de operaciones de escritura en EEPROM para prolongar la vida útil de la EEPROM.

Tabla 13-49 Instrucción USS_Write_Param

KOP / FUP	SCL	Descripción

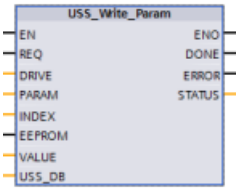
	`USS_Write_Param(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`EEPROM:=_bool_in_,`
	`VALUE:=_variant_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`USS_DB:=_fbtref_inout_);`
		La instrucción USS_Write_Param modifica un parámetro en el accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo bloque de datos.
		USS_Write_Param debe llamarse desde un OB de ciclo de programa principal.

Tabla 13-50 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), REQ indica que se desea una nueva petición de escritura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: DRIVE es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.

Parámetro y tipo		Tipo de datos	Descripción
PARAM	IN	UInt	Número de parámetro: PARAM designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. En algunos accionamientos, el byte más significativo puede acceder a valores PARAM superiores a 2047. Encontrará más información sobre cómo acceder a un rango ampliado en el manual del accionamiento.
INDEX	IN	UInt	Índice de parámetro: INDEX designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
EEPROM	IN	Bool	Guardar en la EEPROM del accionamiento: Si es TRUE (verdadero), una operación de escritura en los parámetros del accionamiento se guardará en la EEPROM de este. Si es FALSE (falso), la operación de escritura será temporal por lo que no se conservará tras desconectar y volver a conectar la alimentación del accionamiento.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Valor del parámetro en el que se debe escribir. Debe ser válido en la transición de REQ.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_Drive_Control en el programa.
DONE[1]	OUT	Bool	Si es TRUE (verdadero), DONE indica que la entrada VALUE se ha escrito en el accionamiento. Este bit se activa cuando USS_Drive_Control detecta los datos de respuesta de escritura del accionamiento. El bit se desactiva cuando los datos de respuesta se solicitan mediante otra consulta USS_Drive_Control o bien en la segunda de las dos llamadas siguientes de USS_Drive_Control.
ERROR	OUT	Bool	Si es TRUE (verdadero), ERROR indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_Port_Scan.
STATUS	OUT	Word	STATUS indica el resultado de la petición de escritura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

[1] El bit DONE indica que se han leído datos válidos del accionamiento de motor referenciado y que se han transmitido a la CPU. No indica que la librería USS sea capaz de leer inmediatamente otros parámetros. Hay que enviar una petición PKW vacía al accionamiento del motor, la cual debe ser acusada por la instrucción antes de que el canal de parámetros para el accionamiento en cuestión vuelva a estar disponible. Si se llama inmediatamente una FC USS_Read_Param o USS_Write_Param para el accionamiento de motor especificado se producirá un error "0x818A".

13.4.4 Códigos de estado USS

La salida STATUS de las funciones USS devuelve los códigos de estado de estas funciones.

Tabla 13-51 Códigos STATUS [1]

STATUS (W#16#....)	Descripción
0000	No hay error
8180	La longitud de la respuesta del accionamiento no concuerda con los caracteres recibidos del accionamiento. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8181	El parámetro VALUE no es un tipo de datos Word, Real o DWord.
8182	El usuario ha introducido un parámetro en formato Word y se ha recibido una respuesta en formato DWord o Real del accionamiento.
8183	El usuario ha introducido un parámetro en formato DWord o Real y se ha recibido una respuesta en formato Word del accionamiento.
8184	El telegrama de respuesta del accionamiento tiene una suma de verificación incorrecta. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8185	Dirección del accionamiento no válida (rango de direcciones válidas: de 1 a 16)
8186	La consigna de velocidad está fuera del rango válido (rango de SP de velocidad válido: -200% a 200%).
8187	Un número de accionamiento incorrecto ha respondido a la petición enviada. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8188	Se ha indicado una longitud de palabra PZD no permitida (rango válido = 2, 4, 6 u 8 palabras)
8189	Se ha indicado una velocidad de transferencia no permitida.
818A	Otra petición para este accionamiento está utilizando el canal de petición de parámetros.
818B	El accionamiento no ha respondido a las peticiones ni reintentos. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
818C	El accionamiento ha devuelto un error avanzado relativo a la petición de parámetros. La descripción de los errores avanzados aparece a continuación de esta tabla.
818D	El accionamiento ha devuelto un error de acceso no permitido relativo a la petición de parámetros. Para más información sobre la limitación de acceso a los parámetros, consulte el manual del accionamiento.
818E	El accionamiento no se ha inicializado. Este código de error se devuelve a USS_Read_Param o USS_Write_Param si USS_Drive_Control no se ha llamado por lo menos una vez para este accionamiento. De esta manera se impide que la inicialización de USS_Drive_Control en el primer ciclo sobrescriba una petición pendiente de lectura o escritura de los parámetros, puesto que inicializa el accionamiento como entrada nueva. Para corregir este error, llame USS_Drive_Control para este número de accionamiento.
80Ax-80Fx	Errores específicos que devuelven los FBs de comunicación PtP llamados por la librería USS - La librería USS no modifica estos códigos de error que se definen en las descripciones de la instrucción PtP.

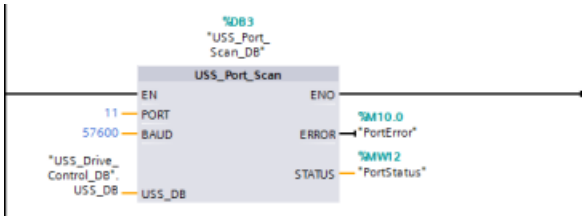
[1] Además de los errores de la instrucción USS indicados arriba, las instrucciones de comunicaciones PtP (Página 959) subyacentes pueden devolver errores.

Para muchos códigos STATUS se suministra información adicional en la variable "USS_Extended_Error" del DB de instancia USS_Drive_Control. Para los códigos STATUS hexadecimales 8180, 8184, 8187 y 818B, USS_Extended_Error contiene el número del accionamiento en que se ha presentado el error de comunicación. Para el código STATUS

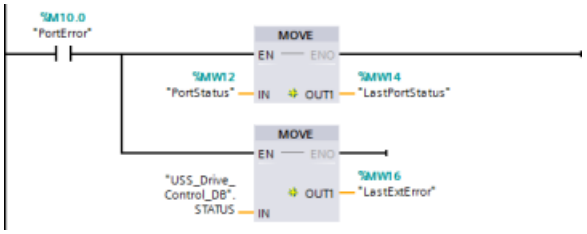
hexadecimal 818C, USS_Extended_Error contiene un código de error que devuelve el accionamiento cuando se utiliza una instrucción USS_Read_Param o USS_Write_Param.

Ejemplo: notificación de errores de comunicación

Los errores de comunicación (STATUS = 16#818B) se notifican solo en la instrucción USS_Port_Scan y no en la instrucción USS_Drive_Control. Si, por ejemplo, la red no ha sido finalizada correctamente, un accionamiento puede pasar a RUN, no obstante la instrucción USS_Drive_Control mostrará el valor 0 para todos los parámetros de salida. En este caso, el error de comunicación solo se puede detectar en la instrucción USS_Port_Scan. Debido a que este error solo es visible durante un ciclo, hay que agregar algunos elementos lógicos de detección, como se indica en el ejemplo a continuación. En este ejemplo, si el bit de error de la instrucción USS_Port_Scan es TRUE, entonces los valores STATUS y USS_Extended_Error se guardan en el área de marcas. El número de accionamiento se deposita en la variable USS_Extended_Error si el código STATUS tiene el valor hexadecimal 8180, 8184, 8187 u 818B.



Segmento 1 El estado del puerto "PortStatus"y los valores de los códigos de error avanzados de"USS_Drive_Control_DB".USS_Extended_Error solo son válidos durante un ciclo del programa. Los valores deben capturarse para un procesamiento posterior.



Segmento 2 El contacto "PortError" inicia el almacenamiento del valor "PortStatus" en "LastPortStatus" y del valor"USS_Drive_Control_DB".USS_Extended_Error en "LastExtError".

Acceso de lectura y escritura a los parámetros internos de un accionamiento

Los accionamientos USS soportan accesos de lectura y escritura a los parámetros internos de los accionamientos. Esta función permite controlar y configurar el accionamiento de forma remota. Las operaciones de acceso a los parámetros del accionamiento pueden fallar debido a errores tales como valores fuera de rango o peticiones no permitidas del modo actual del accionamiento. El accionamiento genera un código de error cuyo valor se devuelve en la variable "USS_Extended_Error". Este valor del código de error es válido únicamente para la última ejecución de una instrucción USS_Read_Param o USS_Write_Param. El código de error del accionamiento se deposita en la variable USS_Extended_Error si el STATUS code tiene el valor hexadecimal 818C. El valor del código de error de USS_Extended_Error depende del modelo de accionamiento. Los códigos de error avanzados para las operaciones de lectura y escritura de parámetros se describen en el manual del accionamiento.

13.4.5 Requisitos generales para la configuración del accionamiento USS

Los requisitos generales para la configuración del accionamiento USS constan de los puntos siguientes:

- Los accionamientos deben configurarse para utilizar 4 palabras PKW.
- Los accionamientos pueden configurarse para 2, 4, 6 u 8 palabras PZD.
- El número de palabras PZD en el accionamiento debe concordar con la entrada PZD_LEN de la instrucción USS_Drive_Control para ese accionamiento.
- La velocidad de transferencia en todos los accionamientos debe concordar con la entrada BAUD de la instrucción USS_Port_Scan.
- El accionamiento debe configurarse para control remoto.
- Para el accionamiento debe configurarse una consigna de frecuencia a USS en el puerto COM.
- La dirección del accionamiento debe configurarse en un rango de 1 a 16 y debe concordar con la entrada DRIVE del bloque USS_Drive_Control para ese accionamiento.
- El control de sentido del accionamiento debe configurarse para que utilice la polaridad de la consigna del accionamiento.
- La red RS485 se debe finalizar correctamente.

13.4.6 Ejemplo: conexión e instalación general del accionamiento USS

Conectar un accionamiento MicroMaster

Esta información sobre accionamientos SIEMENS MicroMaster se suministra a modo de ejemplo. Para otros accionamientos, consulte el manual respectivo para obtener instrucciones de configuración.

Para conectar un accionamiento MicroMaster 4 (MM4), inserte los extremos del cable RS485 en los dos terminales fijadores sin tornillos previstos para el protocolo USS. El cable PROFIBUS y los conectores estándar sirven para conectar el S7-1200.

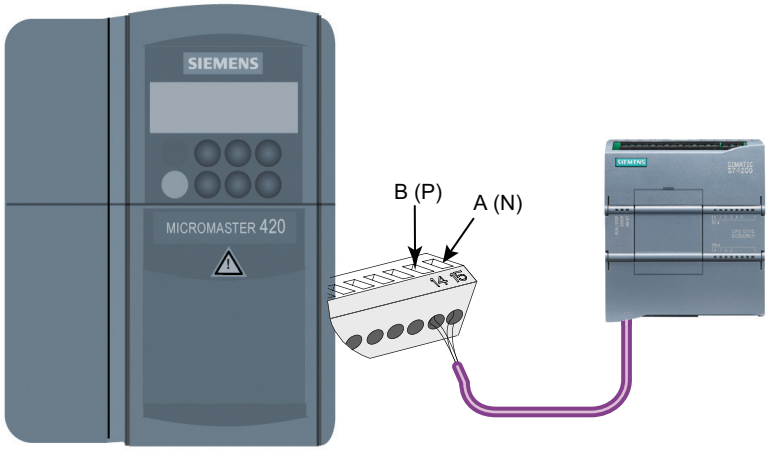
⚠ PRECAUCIÓN

Si se interconectan equipos con potenciales de referencia diferentes, podrían circular corrientes indeseadas por el cable de conexión.

Estas corrientes pueden causar errores de comunicación o deteriorar los equipos. Para evitar corrientes indeseadas, vigile que todos los equipos conectados con un cable de comunicación compartan un circuito de referencia, o bien que estén aislados entre sí. El blindaje debe conectarse a tierra, o bien al pin 1 del conector de 9 pines. Es recomendable conectar a tierra el terminal 20 V del accionamiento MicroMaster.

Los dos hilos del extremo opuesto del cable RS485 se deben insertar en los bloques de terminales del accionamiento MM4. Para conectar el cable a un accionamiento MM4, desmonte

la(s) tapa(s) del accionamiento para acceder al bloque de terminales. Para más información sobre cómo desmontar la(s) tapa(s), consulte el manual del accionamiento MM4.

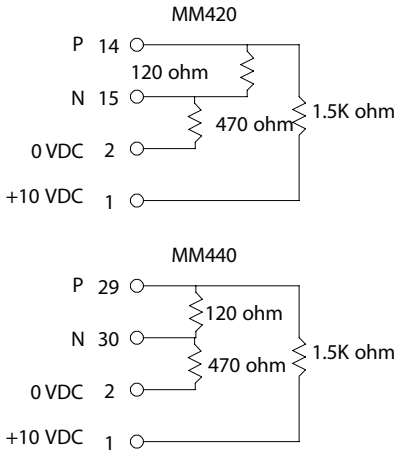


Las conexiones del bloque de terminales están etiquetadas de forma numérica. Utilizando un conector PROFIBUS en el lado del S7-1200, conecte el terminal A del cable al terminal 15 del accionamiento (si es un MM420), o bien al terminal 30 (si es un MM440). Conecte el terminal B del cable B (P) A (N) al terminal 14 (si es un MM420), o bien al terminal 29 (si es un MM440).

Si el S7-1200 está situado en un extremo de la red o si se trata de una conexión punto a punto, será preciso utilizar los terminales A1 y B1 (en vez de A2 y B2) del conector, puesto que estos permiten ajustar el cierre (p. ej. utilizando el conector DP con la referencia 6ES7972-0BA40-0X40).

⚠ **PRECAUCIÓN**
Monte las tapas del accionamiento correctamente antes de conectar la alimentación. Monte correctamente de nuevo las tapas del accionamiento antes de conectar la alimentación.

Si el accionamiento está situado en un extremo de la red, los resistores de cierre y polarización se deberán conectar también a los terminales correctos. El diagrama muestra ejemplos de conexión del accionamiento MM4 necesarios para cierre y polarización.



Configurar accionamientos MicroMaster 4

Antes de conectar un accionamiento al S7-1200, vigile que tenga los siguientes parámetros de sistema. Utilice el teclado del accionamiento para ajustar los parámetros:

1. Restablezca los ajustes de fábrica del accionamiento (opcional).	P0010=30 P0970=1
Si omite el paso 1, entonces asegúrese de que estos parámetros tengan los valores indicados.	Longitud USS PZD = P2012 índice 0=(2, 4, 6 u 8) Longitud USS PKW = P2013 índice 0=4
2. Habilite el acceso de lectura/escritura a todos los parámetros (modo avanzado).	P0003=3
3. Compruebe los ajustes del motor. Los ajustes varían dependiendo del (de los) motor(es) utilizados. Para configurar los parámetros P304, P305, P307, P310 y P311, deberá ajustar primero el parámetro P010 a "1" (modo de puesta en servicio rápida). Cuando haya terminado de ajustar los parámetros, ponga el parámetro P010 a 0. Los parámetros P304, P305, P307, P310 y P311 solo se pueden cambiar en el modo de puesta en servicio rápida.	P0304 = Tensión nominal del motor (V) P0305 = Intensidad nominal del motor (A) P0307 = Potencia nominal del motor (W) P0310 = Frecuencia nominal del motor (Hz) P0311 = Velocidad nominal del motor
4. Ajuste el modo de control (local o remoto).	P0700 índice 0=5
5. Seleccione la consigna de frecuencia a USS en el puerto COM.	P1000 índice 0=5
6. Tiempo de aceleración (opcional) Este es el tiempo (indicado en segundos) que el motor necesita para acelerar hasta la frecuencia máxima.	P1120=(de 0 a 650,00)
7. Tiempo de deceleración (opcional) Este es el tiempo (indicado en segundos) que el motor necesita para decelerar hasta una parada completa.	P1121=(de 0 a 650,00)
8. Ajuste la frecuencia de referencia del puerto serie:	P2000=(de 1 a 650 Hz)
9. Ajuste la normalización USS:	P2009 índice 0=0
10. Ajuste la velocidad de transferencia del puerto serie RS485:	P2010 índice 0= 4 (2400 bit/s) 5 (4800 bit/s) 6 (9600 bit/s) 7 (19200 bit/s 8 (38400 bit/s) 9 (57600 bit/s) 12 (115200 bit/s)
11. Introduzca la dirección del esclavo. Todos los accionamientos (31 como máximo) se pueden controlar a través del bus.	P2011 índice 0=(de 0 a 31)
12. Ajuste el timeout para el puerto serie. Este es el intervalo máximo admisible entre la recepción de dos telegramas de datos. Esta función se utiliza para desactivar el inversor en caso de un fallo de comunicación. La temporización comienza tras haberse recibido un telegrama de datos válido. Si no se recibe otro telegrama de datos dentro del período de tiempo indicado, el inversor se desactivará y se visualizará el código de error F0070. Ajustando el valor a cero se desconecta el control.	P2014 índice 0=(de 0 a 65.535 ms) 0=sin timeout
13. Transfiera los datos de la RAM a la EEPROM:	P0971=1 (iniciar la transferencia) Guardar los cambios de los parámetros en la EEPROM

13.5 Comunicación Modbus

13.5.1 Sinopsis de la comunicación Modbus RTU y Modbus TCP

Códigos de función Modbus

- Una CPU que actúe como maestro Modbus RTU (o cliente Modbus TCP) puede leer y escribir tanto datos como estados de E/S en un esclavo Modbus RTU (o servidor Modbus TCP) remoto. Los datos remotos se pueden leer y procesar en la lógica del programa.
- Una CPU que actúe como esclavo Modbus RTU (o servidor Modbus TCP) permite que un dispositivo de supervisión lea y escriba tanto datos como estados de E/S en la memoria de la CPU. Un maestro RTU (o cliente Modbus TCP) puede escribir nuevos valores en la memoria de la CPU esclavo/servidor que está disponible para la lógica del programa.

⚠ ADVERTENCIA

Prevenir riesgos de seguridad por ataques a la red física

Si un atacante puede acceder físicamente a las redes, posiblemente podrá leer y escribir datos.

Algunas formas de comunicación (intercambio E/S a través de PROFIBUS, PROFINET, AS-i u otro bus de periferia, GET/PUT, bloque T y módulos de comunicación (CM)) carecen de funciones de seguridad. Este tipo de comunicaciones deben protegerse limitando el acceso físico. Si un atacante puede acceder físicamente a las redes con estos tipos de comunicación, posiblemente podrá leer y escribir datos.

Si no consigue proteger estas formas de comunicación, podrían producirse la muerte o lesiones corporales graves.

Para más información sobre seguridad y recomendaciones, consulte las "Operational Guidelines for Industrial Security" en la página del Siemens Industry Online Support.

Tabla 13-52 Funciones de lectura de datos: Leer datos de programa y E/S remotos

Código de función Modbus	Funciones de lectura de esclavo (servidor), direccionamiento estándar
01	Leer bits de salida: De 1 a 2000 bits por petición
02	Leer bits de entrada: De 1 a 2000 bits por petición
03	Leer registros de retención: De 1 a 125 palabras por petición
04	Leer palabras de entrada: De 1 a 125 palabras por petición

Tabla 13-53 Funciones de escritura de datos: Escribir en E/S remota y modificar datos de programa

Código de función Modbus	Funciones de escritura de esclavo (servidor), direccionamiento estándar
05	Escribir un bit de salida: 1 bit por petición
06	Escribir un registro de retención: 1 palabra por petición
15	Escribir uno o más bits de salida: De 1 a 1968 bits por petición
16	Escribir uno o más registros de retención: De 1 a 123 palabras por petición

- Los códigos de función Modbus 08 y 11 proporcionan información de diagnóstico de comunicaciones con dispositivos esclavos.
- El código de función Modbus 0 envía un mensaje Broadcast a todos los esclavos (sin respuesta de los esclavos). La función Broadcast no está disponible para Modbus TCP porque las comunicaciones se basan en conexión.
- El código de función Modbus 23 puede escribir o leer uno o más registros de retención: De 1 a 121/125 palabras (escritura/lectura) por petición. Este código de función solo está disponible para Modbus TCP.

Tabla 13-54 Direcciones de estación de red Modbus

Estación		Dirección
Estación RTU	Dirección de estación estándar	1 a 247
	Dirección de estación avanzada	1 a 65535
Estación TCP	Dirección de estación	Dirección IP y número de puerto

Direcciones de memoria Modbus

El número de direcciones de memoria Modbus disponible depende del modelo de CPU, la cantidad de memoria de trabajo y la cantidad de memoria de CPU usada por otros datos de programa. En esta tabla, se indican valores nominales de rangos de direcciones.

Tabla 13-55 Direcciones de memoria Modbus

Estación		Rango de direcciones
Estación RTU	Dirección de memoria estándar	10K
	Dirección de memoria avanzada	64K
Estación TCP	Dirección de memoria estándar	10K

Comunicación Modbus RTU

Modbus RTU (Remote Terminal Unit) es un protocolo de comunicaciones en red estándar que utiliza conexiones eléctricas RS232 o RS485 para transferencia serie de datos entre dispositivos de red Modbus. Se pueden añadir puertos de red PtP (punto a punto) a una CPU con RS232, CM RS485 o CB RS485.

Modbus RTU utiliza una red maestro/esclavo en la que un solo dispositivo maestro inicia todas las comunicaciones y los esclavos solo pueden responder a una petición del maestro. El maestro envía una petición a una dirección de esclavo y solo esa dirección de esclavo responde a la orden.

Comunicación Modbus TCP

Modbus TCP (Transmission Control Protocol) es un protocolo de comunicaciones en red estándar que usa el conector PROFINET de la CPU para comunicaciones TCP/IP. No se necesitan módulos hardware de comunicación adicionales.

Modbus TCP utiliza conexiones para Open User Communications (OUC) como vía de comunicaciones Modbus. Puede haber varias conexiones cliente-servidor, además de la conexión entre STEP 7 y la CPU. Se admiten conexiones de cliente y de servidor mezcladas hasta el número máximo de conexiones que admita el modelo de CPU (Página 591).

Cada conexión MB_SERVER debe usar un DB de instancia y número de puerto IP únicos. Solo se admite 1 conexión por puerto IP. Cada MB_SERVER (con su DB de instancia y puerto IP únicos) se debe ejecutar individualmente para cada conexión.

Un cliente Modbus TCP (maestro) debe controlar la conexión cliente-servidor con el parámetro DISCONNECT. Las acciones de cliente Modbus básicas se indican a continuación.

1. Iniciar una conexión a una dirección IP y número de puerto IP particulares de un servidor (esclavo)
2. Iniciar transmisión de cliente de un mensaje Modbus y recibir las respuestas del servidor
3. Cuando se desee, iniciar la desconexión de cliente y servidor para permitir la conexión con otro servidor.

Instrucciones Modbus RTU en el programa

- Modbus_Comm_Load: Una ejecución de Modbus_Comm_Load se utiliza para configurar parámetros de puerto PtP como velocidad de transferencia, paridad y control de flujo. Tras configurar un puerto de la CPU para el protocolo Modbus RTU, este solo puede ser utilizado por la instrucción Modbus_Master o Modbus_Slave.
- Modbus_Master: la instrucción Modbus_Master permite que la CPU actúe como maestro Modbus RTU y se comunique con uno o varios esclavos Modbus.
- Modbus_Slave: la instrucción Modbus_Slave permite que la CPU actúe como esclavo Modbus RTU y se comunique con un dispositivo maestro Modbus.

Instrucciones Modbus TCP en el programa

- MB_CLIENT: establece la conexión TCP cliente-servidor, envía mensaje de orden, recibe respuesta y controla la desconexión desde el servidor.
- MB_SERVER: conecta a un cliente Modbus TCP bajo petición, recibe mensaje Modbus y envía respuesta.

Consulte también

Sitio Service & Support de Siemens (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

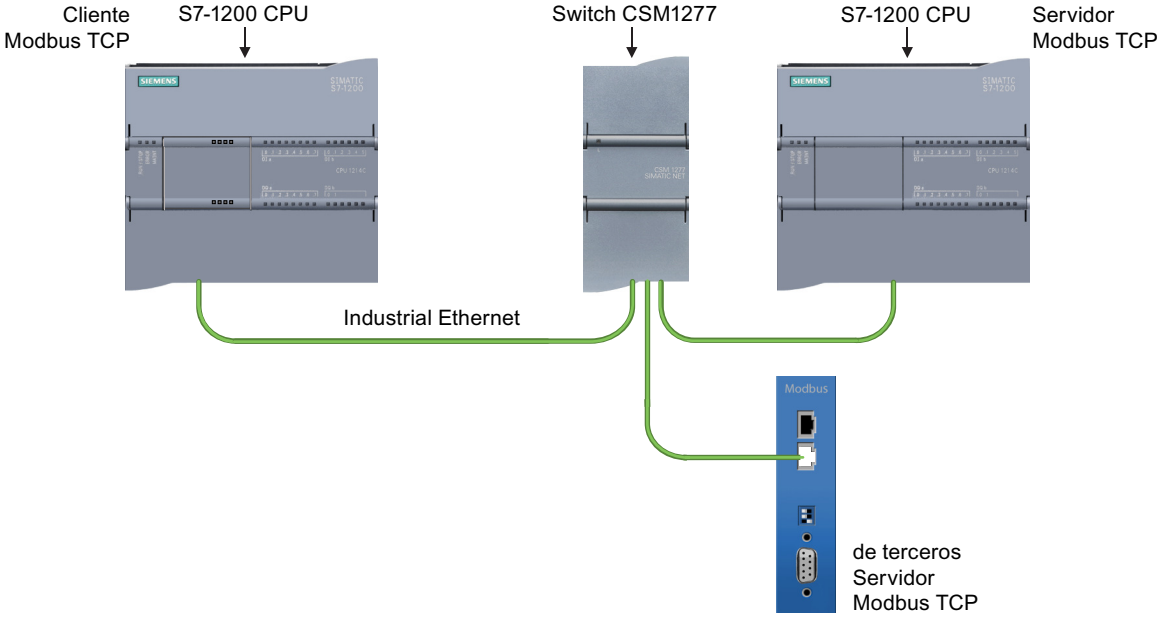
13.5.2 Modbus TCP

13.5.2.1 Vista general

A partir de la versión V4.1 de la CPU S7-1200 junto con STEP 7 V13 SP1, la CPU amplía la capacidad de Modbus TCP para utilizar instrucciones T-block mejoradas.

Por este motivo, S7-1200 soporta dos juegos de instrucciones PtP:

- Instrucciones Legacy Modbus TCP (Página 1142): Estas instrucciones Modbus TCP ya existían antes que la versión V4.0 de S7-1200.
- Instrucciones Modbus TCP (Página 1017): Estas instrucciones Modbus TCP ofrecen toda la funcionalidad de las instrucciones Legacy.



13.5.2.2 Seleccionar la versión de las instrucciones Modbus TCP

En STEP 7 hay disponibles las versiones siguientes de las instrucciones Modbus TCP:

- Versión antigua 2.1: compatible con todas las versiones de CPU y CM
- Versión antigua 3.1: compatible con todas las versiones de CPU y CM
- Versión 4.2: compatible con CPU V4.0 y superiores y con CM V2.1 y superiores
- Versión 5.1: compatible con CPU V4.2 y superiores y con CM V2.1 y superiores
- Versión antigua 6.0: compatible con CPU V4.2 y superiores y con CM V2.1 y superiores

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

En la Task Card Instrucciones, visualice las instrucciones MODBUS TCP en "Otros" del grupo "Comunicación".

Para cambiar la versión de las instrucciones Modbus TCP, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.



Cuando se utiliza el árbol de instrucciones para insertar una instrucción Modbus TCP en el programa, se crea una nueva instancia del FB en el árbol del proyecto. El nuevo FB instancia se puede ver en el árbol de proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción Modbus TCP en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione una instancia del FB Modbus TCP del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción Modbus TCP.

13.5.2.3 Instrucciones Modbus TCP

Instrucción MB_CLIENT (Comunicar como cliente TCP Modbus vía PROFINET)

Tabla 13-56 Instrucción MB_CLIENT

KOP / FUP	SCL	Descripción

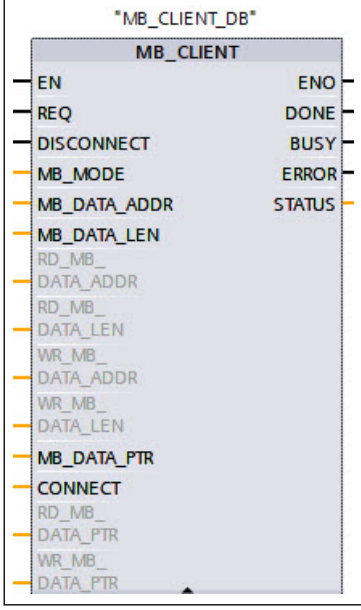
	`"MB_CLIENT_DB"(`
	`  REQ:=_bool_in_,`
	`  DISCONNECT:=_bool_in_,`
	`  MB_MODE:=_usint_in_,`
	`  MB_DATA_ADDR:=_udint_in_,`
	`  MB_DATA_LEN:=_uint_in_,`
	`  RD_MB_DATA_ADDR:=_uint_in_,`
	`  RD_MB_DATA_LEN:=_uint_in_,`
	`  WR_MB_DATA_ADDR:=_uint_in_,`
	`  WR_MB_DATA_LEN:=_uint_in_,`
	`  DONE=>_bool_out_,`
	`  BUSY=>_bool_out_,`
	`  ERROR=>_bool_out_,  STATUS=>_word_out_,`
	`  MB_DATA_PTR:=_variant_inout_,`
	`  CONNECT:=_variant_inout_,`
	`  RD_MB_DATA_PTR:=_variant_inout_,`
	`  WR_MB_DATA_PTR:`
	`=_variant_inout_);`
		MB_CLIENT se comunica como cliente Modbus TCP a través del puerto PROFINET de la CPU S7-1200. No se necesitan módulos hardware de comunicación adicionales.
		MB_CLIENT puede establecer una conexión cliente-servidor, enviar una petición de función Modbus, recibir una respuesta y controlar la desconexión de un servidor Modbus TCP.

Tabla 13-57 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	In	Bool	FALSE = Ninguna petición de comunicación Modbus TRUE = Petición de comunicación con un servidor Modbus TCP
DISCONNECT	IN	Bool	El parámetro DISCONNECT permite que su programa controle la conexión y desconexión con un dispositivo servidor Modbus. Si DISCONNECT = 0 y no hay ninguna conexión, entonces MB_CLIENT intenta establecer una conexión con la dirección IP y el número de puerto asignados. Si DISCONNECT = 1 y existe una conexión, entonces se intenta una operación de desconexión. Siempre que esta entrada esté habilitada, no se intentará ninguna otra operación.
MB_MODE	IN	USInt	Selección de modo: asigna el tipo de petición (lectura, escritura o diagnóstico). Consulte los detalles en la tabla de funciones Modbus que aparece más abajo.
MB_DATA_ADDR	IN	UDInt	Dirección inicial Modbus: asigna la dirección inicial de los datos a los que accederá MB_CLIENT. Para conocer las direcciones válidas, consulte la siguiente tabla de funciones Modbus.
MB_DATA_LEN	IN	UInt	Longitud de datos Modbus: asigna el número de bits o palabras a las que debe accederse en esta petición. Para conocer las longitudes válidas, consulte la siguiente tabla de funciones Modbus.
MB_DATA_PTR	IN_OUT	Variant	Puntero al registro de datos Modbus: El registro almacena temporalmente datos que van a un servidor Modbus o vienen de este. El puntero debe asignar una dirección de memoria M o un DB global no optimizado.
CONNECT	IN_OUT	Variant	Referencia a una estructura de bloque de datos que contiene parámetros de conexión en el tipo de datos del sistema "TCON_IP_v4". También se soportan los siguientes tipos de datos: TCON_IP_V4_SEC, TCON_QDN y TCON_QDN_SEC. Consulte "Parámetros para la conexión PROFINET" (Página 618).
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación MB_CLIENT en curso • 1 - Operación MB_CLIENT en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la ejecución de MB_CLIENT con un error. El código de error en el parámetro STATUS es válido solo durante el ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

Función Modbus 23

Descripción

La función Modbus 23 permite realizar las siguientes acciones en una tarea:

1. Se transfieren datos de la CPU al servidor Modbus y se escriben en uno o más registros de retención.
2. Se leen datos de uno o más registros de retención del servidor Modbus y se transfieren a la CPU.

La instrucción "MB_CLIENT" soporta la función Modbus 23 a partir de la versión V6.0 de la instrucción.

Parámetros

Cuando se utiliza la función Modbus 23, el parámetro MB_MODE debe tener el valor 123.

Los parámetros MB_DATA_ADDR, MB_DATA_LEN y MB_DATA_PTR no se utilizan y deben tener sus valores predeterminados.

Al utilizar la función Modbus23 se utilizan seis nuevos parámetros, que se describen en la tabla siguiente. Cada uno de estos parámetros comienza por "RD_" o "WR_" para indicar que pertenece a la tarea de lectura o escritura, respectivamente. Estos parámetros están ocultos por defecto. Al utilizar la función Modbus 23 hay que utilizar los seis parámetros sin excepciones. Si se utiliza una función Modbus diferente, estos seis parámetros deben tener el valor 0 o deben estar vacíos. De lo contrario se devolverá el valor STATUS 16#818D.

Tabla 13-58 Tipos de datos de los parámetros

Parámetro y tipo		Tipo de datos	Descripción
RD_MB_DATA_ADDR	IN	UInt	Dirección inicial en el dispositivo remoto a partir de la cual deben leerse datos. Valores permitidos: de 0 a 65535
RD_MB_DATA_LEN	IN	UInt	Número de registros que deben leerse del dispositivo remoto. Valores permitidos: de 1 a 125
WR_MB_DATA_ADDR	IN	UInt	Dirección inicial en el dispositivo remoto a partir de la cual deben escribirse datos. Valores permitidos: de 0 a 65535
WR_MB_DATA_LEN	IN	UInt	Número de registros que deben escribirse en el dispositivo remoto. Valores permitidos: de 1 a 121
RD_MB_DATA_PTR	IN_OUT	Variant	Puntero hacia un búfer de datos para los datos que deben leerse del servidor Modbus. Como tipos de datos se permiten los mismos tipos de datos que para MB_DATA_PTR.
WR_MB_DATA_PTR	IN_OUT	Variant	Puntero hacia un búfer de datos para los datos que deben escribirse en el servidor Modbus. Como tipos de datos se permiten los mismos tipos de datos que para MB_DATA_PTR.

Parámetro STATUS

El significado de los valores de STATUS 16#8383, 8189, 818A, 818B se amplía. Se agrega el valor de STATUS 16#818D.

Actualizar un proyecto, actualizar una instrucción

Cuando se actualiza un proyecto existente (p. ej., uno creado con TIA Portal V16) con instrucciones MB_CLIENT (p. ej., versión de la instrucción V5.2), la nueva versión de la instrucción no se utiliza automáticamente en el programa. Para utilizar la función Modbus 23 hay que actualizar la versión de la instrucción manualmente.

Parámetro REQ

FALSE = Ninguna petición de comunicación Modbus
TRUE = Petición de comunicación con un servidor Modbus TCP

Si ninguna instancia de MB_CLIENT está activa y el parámetro DISCONNECT=0, se inicia una nueva petición Modbus cuando REQ=1. Si la conexión no está todavía establecida, se establecerá una nueva.

Si la misma instancia de MB_CLIENT se ejecuta nuevamente con DISCONNECT=0 y REQ=1, antes de que finalice la petición actual no se realizarán más transmisiones Modbus. No obstante, en cuanto finalice la petición actual, se podrá procesar otra petición si MB_CLIENT se ejecuta con REQ=1.

Cuando finalice la petición de comunicación MB_CLIENT actual, el bit DONE será TRUE durante un ciclo. El bit DONE se puede usar como puerta temporal para secuenciar varias peticiones MB_CLIENT.

Nota

Coherencia de los datos de entrada durante el procesamiento de MB_CLIENT

Una vez que un cliente Modbus inicia una operación Modbus, todos los estados de entrada se guardan internamente y se comparan en cada llamada sucesiva. La comparación se utiliza para determinar si una llamada en particular originó la petición de cliente activa. Se puede realizar más de una llamada de MB_CLIENT con un DB de instancia común.

Es importante que las entradas no se cambien durante el tiempo en el que se procesa activamente una operación MB_CLIENT . Si no se observa esta regla, un MB_CLIENT no puede determinar la instancia activa.

Los parámetros MB_MODE y MB_DATA_ADDR seleccionan la función de comunicación Modbus

La instrucción MB_CLIENT utiliza una entrada MB_MODE, en lugar de un código de función. MB_DATA_ADDR asigna la dirección Modbus inicial de los datos remotos.

La combinación de MB_MODE y MB_DATA_ADDR determina el código de función utilizado en el mensaje Modbus real. La tabla siguiente muestra la correspondencia entre el parámetro MB_MODE, MB_DATA_ADDR y la función Modbus.

Tabla 13-59 Funciones Modbus

MB_MODE	Función Modbus	Longitud de datos	Operación y datos	MB_DATA_ADDR
0	01	De 1 a 2000	Leer bits de salida: De 1 a 2000 bits por petición	De 1 a 9999
101	01	De 1 a 2000	Leer bits de salida: De 1 a 2000 bits por petición	De 00000 a 65535
0	02	De 1 a 2000	Leer bits de entrada: De 1 a 2000 bits por petición	De 10001 a 19999
102	02	De 1 a 2000	Leer bits de entrada: De 1 a 2000 bits por petición	De 00000 a 65535

MB_MODE	Función Modbus	Longitud de datos	Operación y datos	MB_DATA_ADDR
0	03	De 1 a 125	Leer registros de retención: De 1 a 125 palabras por petición	De 40001 a 49999 o De 400001 a 465535
103	03	De 1 a 125	Leer registros de retención: De 1 a 125 palabras por petición	De 00000 a 65535
0	04	De 1 a 125	Leer palabras de entrada: De 1 a 125 palabras por petición	De 30001 a 39999
104	04	De 1 a 125	Leer palabras de entrada: De 1 a 125 palabras por petición	De 00000 a 65535
1	05	1	Escribir un bit de salida: Un bit por petición	De 1 a 9999
105	05	1	Escribir un bit de salida: Un bit por petición	De 00000 a 65535
1	06	1	Escribir un registro de retención: 1 palabra por petición	De 40001 a 49999 o De 400001 a 465535
106	06	1	Escribir un registro de retención: 1 palabra por petición	De 00000 a 65535
1	15	De 2 a 1968	Escribir varios bits de salida: De 2 a 1968 bits por petición	De 1 a 9999
1	16	De 2 a 123	Escribir varios registros de retención: de 2 a 123 palabras por petición	De 40001 a 49999 o De 400001 a 465535
2	15	De 1 a 1968	Escribir uno o más bits de salida: De 1 a 1968 bits por petición	De 1 a 9999
115	15	De 1 a 1968	Escribir uno o más bits de salida: De 1 a 1968 bits por petición	De 00000 a 65535
2	16	De 1 a 123	Escribir uno o más registros de retención: De 1 a 123 palabras por petición	De 40001 a 49999 o De 400001 a 465535
116	16	De 1 a 123	Escribir uno o más registros de retención: De 1 a 123 palabras por petición	De 00000 a 65535
11	11	0	Leer la palabra de estado de la comunicación del servidor y el contador de eventos. Esta palabra de estado indica ocupado (0 = no ocupado, 0xFFFF = ocupado). El contador de eventos se incrementa cada vez que se finaliza un mensaje correctamente. En esta función se ignoran los parámetros MB_DATA_ADDR y MB_DATA_LEN de MB_CLIENT.	
80	08	1	Comprobar el estado del servidor con el código de diagnóstico 0x0000 (test de Loopback, el servidor envía de regreso la petición) 1 palabra por petición	

MB_MODE	Función Modbus	Longitud de datos	Operación y datos	MB_DATA_ADDR
81	08	1	Inicializar el contador de eventos del servidor con el código de diagnóstico 0x000A 1 palabra por petición	
123	23	de 1 a 121 (escribir) de 1 a 125 (leer)	Escribir registros de retención del dispositivo remoto y leer registros de retención del dispositivo remoto en una tarea. Nota: "MB_CLIENT" soporta esta función Modbus a partir de la versión V6.0 de la instrucción. Los parámetros RD_MB_DATA_ADDR, RD_MB_DATA_LEN, WR_MB_DATA_ADDR, WR_MB_DATA_LEN, RD_MB_DATA_PTR y WR_MB_DATA_PTR se utilizan con este fin.	
de 3 a 10, de 12 a 79, de 82 a 100, de 107 a 114, de 117 a 255			Reservado	

Nota

MB_DATA_PTR asigna un búfer para guardar datos leídos/escritos hasta/desde un servidor Modbus TCP

El búfer de datos puede ubicarse en una dirección de memoria M o un DB global no optimizado.

Para un búfer en la memoria M utilice el formato de puntero Any. Este está en el formato P#"dirección de bit" "tipo de datos" "longitud"; por ejemplo: P#M1000.0 WORD 500.

El parámetro MB_DATA_PTR asigna un búfer de comunicación

- Funciones de comunicación de MB_CLIENT:
  - Leer y escribir datos de 1 bit de las direcciones del servidor Modbus (00001 a 09999)
  - Leer datos de 1 bit de las direcciones del servidor Modbus (de 10001 a 19999)
  - Leer datos de palabras de 16 bits de las direcciones del servidor Modbus (de 30001 a 39999) y (de 40001 a 49999)
  - Escribir datos de palabras de 16 bit de las direcciones del servidor Modbus (40001 a 49999)
- Los datos de tamaño de palabra o de bit se transfieren a/desde el búfer de memoria M o el DB asignado por MB_DATA_PTR.

- Si MB_DATA_PTR asigna un DB como búfer, el usuario debe asignar tipos de datos a todos los elementos de datos del DB.
  - El tipo de datos de 1 bit Bool representa una dirección de bit Modbus
  - Los tipos de datos de una palabra de 16 bits como WORD, UInt e Int representan una dirección de palabra Modbus
  - Los tipos de datos de palabra doble de 32 bits como DWORD, DInt y Real representan dos direcciones de palabra Modbus
- MB_DATA_PTR puede asignar elementos de DB complejos como:
  - Matrices
  - Estructuras con nombre en las que cada elemento es único
  - Estructuras complejas con nombre, en las que cada elemento tiene un nombre unívoco y un tipo de datos de 16 o 32 bits
- No es necesario que las áreas de datos de MB_DATA_PTR se encuentren en el mismo bloque de datos global (o área de memoria M). Se puede asignar un bloque de datos para lecturas Modbus, otro para escrituras Modbus o un bloque de datos para cada MB_CLIENT.

El parámetro CONNECT asigna datos utilizados para establecer una conexión PROFINET

Debe utilizar un bloque de datos global y almacenar la conexión necesaria antes de hacer referencia a este DB en el parámetro CONNECT.

1. Cree un nuevo DB global o utilice un DB global existente para almacenar los datos de CONNECT. Se puede utilizar un DB para almacenar varias estructuras de datos TCON_IP_v4. Cada conexión de cliente o servidor Modbus TCP usa una estructura de datos TCON_IP_v4. Se hace referencia a los datos de conexión en el parámetro CONNECT.
2. Denomine el DB y una variable estática con un nombre significativo. Por ejemplo, denomine el bloque de datos "Conexiones Modbus" y una variable estática "TCPactiva_1" (para la conexión Modbus TCP cliente 1).
3. En el editor de DB, asigne el tipo de datos del sistema "TCON_IP_v4" en la columna Tipo de datos, para la variable estática de ejemplo "TCPactiva_1".
4. Expanda la estructura TCON_IP_v4 para poder modificar los parámetros de conexión, como se muestra en la imagen siguiente.
5. Modifique los datos en la estructura TCON_IP_v4 para una conexión MB_CLIENT.
6. Introduzca la referencia de estructura de DB para el parámetro CONNECT de MB_CLIENT. En el ejemplo, esto sería "Conexiones Modbus".TCPactiva_1.

Modbus connections

	Nombre	Tipo de datos	Valor de arran...	Comentario
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

Modificar los datos de DB de TCON_IP_V4 para cada conexión MB_CLIENT

- **InterfaceID**: Por medio de la ventana de configuración de dispositivo, haga clic en la imagen del puerto PROFINET de la CPU. A continuación, haga clic en la pestaña de propiedades generales y utilice el identificador de hardware que verá allí.
- **ID**: Introduzca un número de identificación de la conexión entre 1 y 4095. La comunicación Modbus TCP se realiza utilizando las instrucciones TCON, TDISCON, TSEND y TRCV subyacentes para OUC (Open User Communication).
- **ConnectionType**: Para TCP/IP, use el valor predeterminado 16#0B (número decimal = 11).
- **ActiveEstablished**: Este valor debe ser 1 o TRUE. La conexión está activa en el MB_CLIENT que inicia la comunicación Modbus.
- **RemoteAddress**: Introduzca la dirección IP del servidor Modbus TCP de destino en los cuatro elementos de matriz ADDR. Por ejemplo, escriba 192.168.2.241, como en la imagen anterior.
- **RemotePort**: El valor predeterminado es 502. Este número es el número de puerto IP del servidor Modbus con el que MB_CLIENT intenta conectarse y comunicarse. Algunos servidores Modbus de terceros requieren que use otro número de puerto.
- **LocalPort**: Este valor debe ser 0 para una conexión MB_CLIENT.

Varias conexiones de cliente

Un cliente Modbus TCP puede soportar conexiones simultáneas hasta el número máximo de conexiones Open User Communication permitido por el PLC. El número total de conexiones en un PLC dado, clientes y servidores Modbus TCP incluidos, no debe sobrepasar el número máximo de conexiones Open User Communication admitidas.

Cada conexión de cliente concurrente debe cumplir estas reglas:

- Cada conexión MB_CLIENT debe usar un DB de instancia exclusivo
- Cada conexión MB_CLIENT debe asignar una dirección IP de servidor exclusiva
- Cada conexión MB_CLIENT debe asignar una ID de conexión exclusiva
- Según la configuración del servidor pueden ser necesarios, o no, números únicos de puerto IP

Debe usarse una ID de conexión diferente con cada DB de instancia. Resumiendo, el DB de instancia y la ID de conexión están emparejadas y deben ser únicas para cada conexión.

Tabla 13-60 Bloque de datos de instancia MB_CLIENT: variables estáticas accesibles para el usuario

Variable	Tipo de datos	Valor predeterminado	Descripción
Blocked_Proc_Timeout	Real	3,0	Tiempo (en segundos) que hay que esperar a una instancia cliente Modbus bloqueada antes de eliminar esta instancia como ACTIVA. Esto puede ocurrir, por ejemplo, si se ha lanzado una petición de cliente y la aplicación deja de ejecutar la función de cliente antes de haber finalizado por completo la petición. El límite máximo de S7-1200 es 55 segundos.
MB_Unit_ID	Word	255	Identificador de unidad Modbus: Un servidor Modbus TCP se direcciona a través de su dirección IP. Por lo tanto, el parámetro MB_UNIT_ID no se usa para el direccionamiento Modbus TCP. El parámetro MB_UNIT_ID corresponde a la dirección de esclavo en el protocolo Modbus RTU. Si un servidor Modbus TCP se usa como pasarela a un protocolo Modbus RTU, MB_UNIT_ID puede usarse para identificar el dispositivo esclavo conectado en la red serie. MB_UNIT_ID se usaría para reenviar la petición a la dirección del esclavo Modbus RTU correcta. Algunos dispositivos Modbus TCP pueden necesitar que el parámetro MB_UNIT_ID esté dentro de un rango restringido.
RCV_TIMEOUT	Real	2,0	Tiempo en segundos que el MB_CLIENT espera a que el servidor responda a una petición.
Conectado	Bool	0	Indica si la conexión al servidor asignado está conectada o desconectada: 1=conectada, 0=desconectada

Tabla 13-61 Errores de protocolo MB_CLIENT

STATUS* (W#16#)	Errores locales y/o remotos	Código de error en la respuesta de MB_SERVER (B#16#)	Descripción
80C8	Local	-	No hay respuesta del servidor en el período definido. Compruebe la conexión con el servidor Modbus. Este error solo se notifica una vez realizados los intentos de repetición configurados. Si la instrucción "MB_CLIENT" no recibe una respuesta con el ID de transacción transferido originalmente (consulte la variable estática MB_TRANSACTION_ID) dentro del período definido, se devuelve este código de error.
8380	Local	-	La trama Modbus recibida tiene un formato incorrecto o no se han recibido suficientes bytes.
8381	Remoto	01	El código de función no se soporta.

STATUS* (W#16#)	Errores locales y/o remotos	Código de error en la respuesta de MB_SERVER (B#16#)	Descripción
8382	Local	-	• La longitud de la trama Modbus en el encabezado no concuerda con el número de bytes recibidos. • El número de bytes no concuerda con el número de bytes realmente transferidos (solo las funciones 1-4). Esto ocurre, por ejemplo, cuando "MB_CLIENT" solicita un número de palabras impar pero "MB_SERVER" siempre envía un número de palabras par. • La dirección inicial en la trama recibida no concuerda con la dirección inicial guardada (funciones 5, 6, 15 y 16). • El número de palabras no concuerda con el número de palabras realmente transferidas (funciones 15 y 16).
	Remoto	03	Especificación de longitud no válida en la trama Modbus recibida. Compruebe el lado del servidor.
8383	Local	-	• Versión de instrucción < V6.0: Error al leer o escribir datos o acceso fuera del área de direcciones de MB_DATA_PTR. • Versión de instrucción >= V6.0: Error al leer o escribir datos o acceso fuera del área de direcciones de MB_DATA_PTR, RD_MB_DATA_PTR o WR_MB_DATA_PTR.
	Remoto	02	Error al leer o escribir datos o acceso fuera del área de direcciones del servidor.
8384	Local	-	• Se ha recibido un código de excepción no válido. • Se ha recibido un valor de datos distinto del enviado originalmente por el cliente (funciones 5, 6 y 8). • Se ha recibido un valor de estado no válido (función 11).
	Remoto	03	Error en el valor de datos para la función 5
8385	Local	-	• Código de diagnóstico no soportado. • Se ha recibido un código de subfunción distinto del enviado originalmente por el cliente (función 8).
	Remoto	03	Código de diagnóstico no soportado.
8386	Local	-	El código de función recibido no concuerda con el enviado originalmente.
8387	Local	-	El ID de protocolo de la trama Modbus TCP recibida por el servidor no es "0".
8388	Local	-	El servidor Modbus ha enviado una longitud de datos distinta de la solicitada. Este error se produce solo cuando se utilizan las funciones Modbus 5, 6, 15 o 16.
* Los códigos de estado se pueden mostrar como valores enteros o hexadecimales en el editor de programas. Encontrará información sobre cómo cambiar los formatos de visualización en "Consulte también".			

Tabla 13-62 Códigos de condición de ejecución de MB_CLIENT [1]

STATUS (W#16#)	Errores de parámetros de MB_CLIENT
7001	MB_CLIENT está esperando a que un servidor Modbus responda a una petición de conexión o de desconexión en el puerto TCP asignado. Esto solo se devuelve en la primera ejecución de una operación de conexión o de desconexión.
7002	MB_CLIENT está esperando a que un servidor Modbus responda a una petición de conexión o de desconexión en el puerto TCP asignado. Esto se devolverá en las ejecuciones sucesivas, mientras se espera a que finalice una operación de conexión o de desconexión.
7003	Una operación de desconexión ha finalizado correctamente (válido solo para un ciclo del PLC).
80C8	El servidor no ha respondido en el tiempo asignado. MB_CLIENT debe recibir una respuesta mediante la ID de transacción que se transmitió originalmente dentro del tiempo asignado o se devuelve este error. Compruebe la conexión al dispositivo servidor Modbus. Este error solo se devuelve una vez que se han ejecutado los intentos (si procede).
8188	El parámetro MB_MODE tiene un valor no válido.
8189	• Versión de instrucción < V6.0: Direccionamiento de datos no válido en el parámetro MB_DATA_ADDR. • Versión de instrucción >= V6.0: Direccionamiento de datos no válido en el parámetro MB_DATA_ADDR, RD_MB_DATA_ADDR o WR_MB_DATA_ADDR
818A	• Versión de instrucción < V6.0: Longitud de datos no válida en el parámetro MB_DATA_LEN. • Versión de instrucción >= V6.0: Longitud de datos no válida en el parámetro MB_DATA_LEN, RD_MB_DATA_LEN o WR_MB_DATA_LEN
818B	Puntero al área DATA_PTR no válido. Puede ser la combinación de MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Puntero DATA_PTR a un área de DB no optimizada (debe ser un área de memoria M o DB no optimizada)
818D	Uno o más parámetros no tienen su valor predeterminado pero no se usan con la función Modbus especificada. Ejemplo: Si MB_MODE tiene el valor 123, MB_DATA_ADDR y MB_DATA_LEN deben tener el valor 0 y MB_DATA_PTR debe estar vacío. Si MB_MODE tiene un valor diferente de 123, todos los parámetros que comienzan con "RD_" o "WR_" deben tener el valor 0 o estar vacíos.
8200	El puerto está ocupado porque está procesando una petición Modbus existente.
8380	La trama Modbus recibida es incorrecta o no se han recibido suficientes bytes.
8387	El parámetro de ID de conexión asignado es diferente de la ID utilizada para peticiones anteriores. Solo se puede usar una ID de conexión en cada DB de instancia de MB_CLIENT. Este código también se devuelve como error interno si la ID de protocolo de Modbus TCP recibida de un servidor no es 0.
8388	Un servidor Modbus ha devuelto una cantidad de datos diferente de la solicitada. Este código se aplica a las funciones Modbus 15 y 16.

[1] Además de los errores de MB_CLIENT indicados, las instrucciones de comunicación del bloque T subyacente (TCON, TDISCON, TSEND y TRCV) pueden devolver errores.

Consulte también

Conexiones de comunicación asíncronas (Página 591)

Instrucción MB_SERVER (Comunicar como servidor TCP Modbus vía PROFINET)

Tabla 13-63 Instrucción MB_SERVER

KOP / FUP	SCL	Descripción

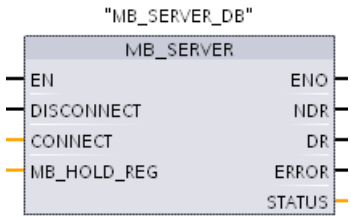
	`"MB_SERVER_DB"(`
	`DISCONNECT:=_bool_in_,`
	`CONNECT:=_variant_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	
	`MB_HOLD_REG:=_variant_inout_);`
		MB_SERVER se comunica como servidor Modbus TCP a través del puerto PROFINET de la CPU S7-1200. No se necesitan módulos hardware de comunicación adicionales.
		MB_SERVER puede aceptar una petición para conectarse a un cliente Modbus TCP, recibir una petición de función Modbus y enviar un mensaje de respuesta

Tabla 13-64 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
DISCONNECT	IN	Bool	MB_SERVER intenta establecer una conexión "pasiva" con un dispositivo interlocutor. Es decir, el servidor escucha pasivamente una petición de conexión TCP desde cualquier dirección IP que lo solicite. Si DISCONNECT = 0 y no existe una conexión, puede iniciarse una conexión pasiva. Si DISCONNECT = 1 y existe una conexión, se inicia una operación de desconexión. Este parámetro permite que su programa controle cuándo se acepta una conexión. Siempre que esta entrada esté habilitada, no se intentará ninguna otra operación.
CONNECT	IN	Variant	Referencia a una estructura de bloque de datos que contiene parámetros de conexión en el tipo de datos del sistema "TCON_IP_v4". También se soportan los siguientes tipos de datos: TCON_IP_V4, TCON_QDN y TCON_QDN_SEC. Consulte "Parámetros de la conexión PROFINET" (Página 618).
MB_HOLD_REG	IN_OUT	Variant	Puntero al registro de retención Modbus de MB_SERVER: El registro de retención debe ser una dirección de memoria M o bien un DB global no optimizado. Esta área de memoria se utiliza para retener los datos a los que un cliente Modbus puede acceder con las funciones de registro Modbus 3 (lectura), 6 (escritura), 16 (escritura) y 23 (escritura/lectura).
NDR	OUT	Bool	Nuevos datos listos: 0 = No hay datos nuevos, 1= Indica que el cliente Modbus ha escrito datos nuevos
DR	OUT	Bool	Lectura de datos: 0 = No se han leído datos, 1= Indica que el cliente Modbus ha leído datos.
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la ejecución de MB_SERVER con un error. El código de error en el parámetro STATUS es válido solo durante el ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

Nota

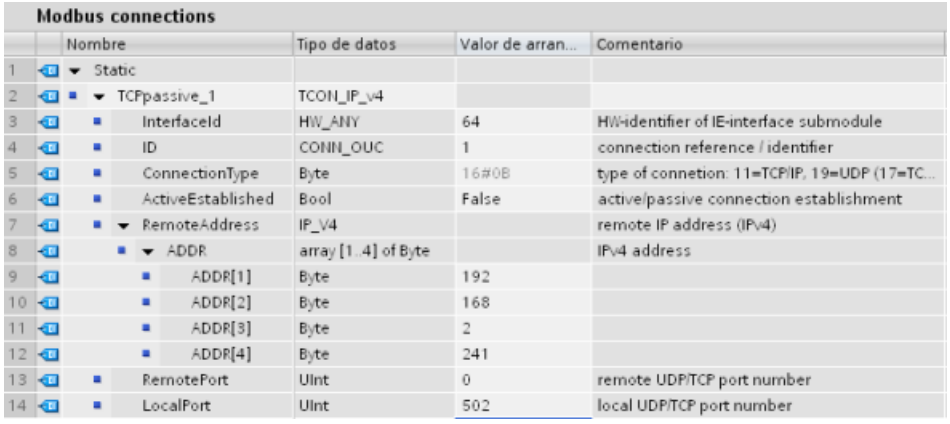
Requisito de versión del firmware de la CPU

Las instrucciones Modbus TCP descritas en este apartado del manual requieren una versión de firmware V4.1 o superior.

El parámetro CONNECT asigna datos utilizados para establecer una conexión PROFINET

Debe utilizar un bloque de datos global y almacenar la conexión necesaria antes de hacer referencia a este DB en el parámetro CONNECT.

1. Cree un nuevo DB global o utilice un DB global existente para almacenar los datos de CONNECT. Se puede utilizar un DB para almacenar varias estructuras de datos TCON_IP_v4. Cada conexión de cliente o servidor Modbus TCP usa una estructura de datos TCON_IP_v4. Se hace referencia a los datos de conexión en el parámetro CONNECT.
2. Denomine el DB y una variable estática con un nombre significativo. Por ejemplo, denomine el bloque de datos "Conexiones Modbus" y una variable estática "TCPpasiva_1" (para la conexión Modbus TCP de servidor 1).
3. En el editor de DB, asigne el tipo de datos del sistema "TCON_IP_v4" en la columna Tipo de datos, para la variable estática de ejemplo "TCPactiva_1".
4. Expanda la estructura TCON_IP_v4 para poder modificar los parámetros de conexión, como se muestra en la imagen siguiente.
5. Modifique los datos en la estructura TCON_IP_v4 para una conexión MB_SERVER.
6. Introduzca la referencia de estructura de DB para el parámetro CONNECT de MB_SERVER. En el ejemplo, esto sería "Conexiones Modbus".TCPpasiva_1.



Modbus connections

	Nombre	Tipo de datos	Valor de arran...	Comentario
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

Modificar los datos de DB de TCON_IP_V4 para cada conexión MB_SERVER

- **InterfaceID**: Por medio de la ventana de configuración de dispositivo, haga clic en la imagen del puerto PROFINET de la CPU. A continuación, haga clic en la pestaña de propiedades generales y utilice el identificador de hardware que verá allí.
- **ID**: Introduzca un número entre 1 y 4095 que sea exclusivo para esta conexión. La comunicación Modbus TCP se realiza utilizando las instrucciones TCON, TDISCON, TSEND y TRCV subyacentes para OUC (Open User Communication). Se permiten hasta ocho conexiones OUC simultáneas.
- **ConnectionType**: Para TCP/IP, use el valor predeterminado 16#0B (valor decimal = 11).
- **ActiveEstablished**: Este valor debe ser 0 o FALSE. La conexión es pasiva porque MB_SERVER espera una petición de comunicación de un cliente Modbus.
- **RemoteAddress**: Hay dos opciones.
  - Use 0.0.0.0 y MB_CLIENT responderá a una petición Modbus de cualquier cliente TCP.
  - Introduzca la dirección IP de un cliente Modbus TCP de destino y MB_CLIENT solo responderá a una petición que provenga de la dirección IP de ese cliente. Por ejemplo, escriba 192.168.2.241, como en la imagen anterior.
- **RemotePort**: Este valor debe ser 0 para una conexión MB_SERVER.
- **LocalPort**: El valor predeterminado es 502. Este es el número de puerto IP del cliente Modbus con el que MB_SERVER intenta conectarse y comunicarse. Algunos clientes Modbus de terceros requieren otro número de puerto.

Direcciones de imagen de proceso y Modbus

MB_SERVER permite los códigos de función Modbus de entrada (1, 2, 4, 5 y 15) para leer/escribir bits/palabras directamente en la imagen de proceso de entrada/salida. Para los códigos de función de transferencia de datos (3, 6 y 16), el parámetro MB_HOLD_REG debe definirse como un tipo de datos superior a un byte. La tabla siguiente muestra el mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

Tabla 13-65 Mapeo de las direcciones Modbus en la memoria imagen de proceso

Funciones Modbus						**S7-1200**	
Códigos	Función	Área de datos	Rango de direcciones			Área de datos	Dirección de la CPU
01	Leer bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
02	Leer bits	Entrada	10001	a	18192	Memoria imagen de proceso de las entradas	I0.0 a I1023.7
04	Leer palabras	Entrada	30001	a	30512	Memoria imagen de proceso de las entradas	IW0 a IW1022
05	Escribir bit	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
15	Escribir bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7

Los códigos de funciones de mensajes Modbus (3, 6 y 16) entrantes leen o escriben palabras en un registro de retención Modbus, que puede estar en la memoria M o en un bloque de datos. El tipo de registro de retención se especifica con el parámetro MB_HOLD_REG.

Nota

Asignación del parámetro MB_HOLD_REG

Los registros de retención Modbus definidos como array of word, entero, carácter ancho, entero sin signo, byte, entero corto, entero corto sin signo, carácter, palabra doble, entero doble, entero doble sin signo o real pueden colocarse en cualquier área de memoria.

Los registros de retención Modbus que se definan como estructuras deberán colocarse en DB no optimizados.

Para un registro de retención Modbus en memoria M, utilice el formato de puntero ANY. Es decir, en el formato P#"Bit Address" "Data Type" "Length". Un ejemplo sería P#M1000.0 WORD 500.

La tabla siguiente muestra ejemplos del mapeo de las direcciones Modbus en el registro de retención usado para los códigos de función 03 (leer palabras), 06 (escribir palabras) y 16 (escribir palabras). El límite superior real de las direcciones de DB está determinado por los límites máximos de memoria de trabajo y de memoria M del respectivo modelo de CPU.

Tabla 13-66 Ejemplos de mapeo de direcciones Modbus en direcciones de memoria de CPU

Dirección de ModBus	Ejemplos del parámetro 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]

Encabezado del protocolo de aplicación Modbus

El encabezado del protocolo de aplicación Modbus lo ocupan los siete primeros bytes de cada mensaje Modbus TCP. Dicho encabezado contiene el identificador de la transacción, el identificador del protocolo, la longitud y el identificador de la unidad. El mensaje de respuesta de la instrucción MB_SERVER contiene los mismos valores para el identificador de la transacción, el identificador del protocolo y el identificador de la unidad que los recibidos en el mensaje de petición Modbus. La instrucción MB_SERVER calcula el campo Longitud.

Varias conexiones de servidor

Se pueden crear varias conexiones de servidor. Un solo PLC puede establecer conexiones concurrentes con varios clientes Modbus TCP.

Un servidor Modbus TCP puede soportar conexiones simultáneas hasta el número máximo de conexiones Open User Communication permitido por el PLC. El número total de conexiones en un PLC dado, clientes y servidores Modbus TCP incluidos, no debe sobrepasar el número máximo de conexiones Open User Communication (Página 591). Las conexiones Modbus TCP pueden compartirse entre conexiones de cliente o de servidor.

Cada conexión de servidor concurrente debe cumplir estas reglas:

- Cada conexión MB_SERVER debe usar un DB de instancia exclusivo.
- Cada conexión MB_SERVER debe asignar un número de puerto IP exclusivo. Solo se admite 1 conexión por puerto.
- Cada conexión MB_SERVER debe asignar un ID de conexión unívoco.
- El MB_SERVER debe ser llamado individualmente para cada conexión (con su DB de instancia respectivo).

El ID de conexión debe ser unívoco para cada conexión individual. Debe usarse un ID de conexión unívoco con cada DB de instancia. El DB de instancia y el ID de conexión están emparejados y deben ser unívocos para cada conexión.

Tabla 13-67 Códigos de función de diagnóstico Modbus

Funciones de diagnóstico Modbus MB_SERVER		
Códigos	Subfunción	Descripción
08	0x0000	Devolver datos de consulta del test de eco: MB_SERVER reenvía a un cliente Modbus una palabra de datos que se ha recibido.
08	0x000A	Borrar contador de eventos de comunicación: La instrucción MB_SERVER borra el contador de eventos de comunicación utilizado para la función Modbus 11.
11		Consultar contador de eventos de comunicación: La instrucción MB_SERVER utiliza un contador de eventos de comunicación interno para registrar el número de peticiones de lectura y escritura Modbus correctas que se envían al servidor Modbus. El contador no incrementa con ninguna petición de Función 8, Función 11, ni con cualquier petición que derive en un error de comunicación. La función Broadcast no está disponible para Modbus TCP porque solo puede haber una conexión cliente-servidor a la vez.

Variables del bloque de datos (DB) de la instrucción MB_SERVER

Esta tabla muestra las variables estáticas públicas almacenadas en el bloque de datos de instancia MB_SERVER que se pueden utilizar en el programa.

Tabla 13-68 Variables estáticas públicas MB_SERVER

Variable	**Tipo de datos**	**Valor predeterminado**	**Descripción**
HR_Start_Offset	Word	0	Asigna la dirección inicial del registro de retención Modbus.
Request_Count	Word	0	Número de todas las peticiones recibidas por este servidor.
Server_Message_Count	Word	0	Número de peticiones recibidas para este servidor específico.
Xmt_Rcv_Count	Word	0	Número de transmisiones o recepciones con error detectado. También se incrementa con cada mensaje recibido que sea un mensaje Modbus no válido.
Exception_Count	Word	0	Errores específicos Modbus que requieren una excepción devuelta
Success_Count	Word	0	Número de peticiones recibidas para este servidor específico que no tienen errores de protocolo.

Variable	Tipo de datos	Valor predeterminado	Descripción
Connected	Bool	0	Indica si la conexión al cliente asignado está conectada o desconectada: 1=conectada, 0=desconectada
QB_Start	UInt	0	Dirección inicial de los bytes de salida en los que la CPU puede escribir (QB0 a QB65535)
QB_Count	UInt	65535	El número de bytes en los que puede escribir un dispositivo remoto. Si QB_Count = 0, un dispositivo remoto no podrá escribir en las salidas. Ejemplo: Para permitir la escritura solo en QB10 a QB17, QB_Start = 10 y QB_Count = 8.
QB_Read_Start	UInt	0	Dirección inicial de los bytes de salida en los que la CPU puede leer (QB0 a QB65535)
QB_Read_Count	UInt	65535	El número de bytes de salida en los que puede leer un dispositivo remoto. Si QB_Count = 0, un dispositivo remoto no podrá leer las salidas. Ejemplo: Para permitir la lectura solo en QB10 a QB17, QB_Start = 10 y QB_Count = 8.
IB_Read_Start	UInt	0	La dirección inicial de los bytes de entrada en los que la CPU puede leer (IB0 a IB65535)
IB_Read_Count	UInt	65535	El número de bytes de entrada en los que puede leer un dispositivo remoto. Si IB_Count = 0, un dispositivo remoto no podrá leer las entradas. Ejemplo: Para permitir la lectura solo en IB10 a IB17, IB_Start = 10 e IB_Count = 8.
NDR_immediate	Bool	FALSE	Significado idéntico al parámetro NDR (New Data Ready). MB_SERVER actualiza "NDR_immediate" en la misma llamada que procesa una petición de escritura Modbus TCP.
DR_immediate	Bool	FALSE	Significado idéntico al parámetro DR (Data Ready). MB_SERVER actualiza "DR_immediate" en la misma llamada que procesa una petición de escritura Modbus TCP.

El programa puede escribir datos en las operaciones de control del servidor Modbus y en las variables siguientes:

- HR_Start_Offset
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- IB_Read_Start
- IB_Read_Count

Los requisitos en cuanto a la versión para la disponibilidad de las variables de bloques de datos (DB) de la instrucción MB_SERVER son los siguientes:

Tabla 13-69 Requisitos para la disponibilidad de variables del bloque de datos (DB) de la instrucción MB_SERVER: instrucción, TIA Portal y CPU S7-1200

Versión de la instrucción MB_SERVER	Versión del TIA Portal	Versión del firmware (FW) de la CPU S7-1200	Variables de bloque de datos
4.2	V14 SP1	FW de la CPU V4.0 o posterior	QB_Start
			QB_Count
5.0 o posterior	V15 o posterior	FW de la CPU V4.2 o posterior	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count
			NDR_immediate
			DR_immediate

HR_Start_Offset

Las direcciones del registro de retención Modbus empiezan en 40001. Estas direcciones corresponden a la dirección de memoria del PLC inicial del registro de retención. No obstante, se puede usar la variable "HR_Start_Offset" para que defina la dirección inicial del registro de retención Modbus en un número diferente de 40001.

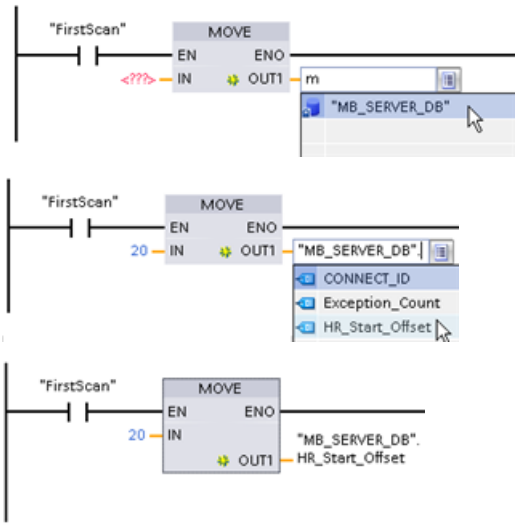
Por ejemplo, si el registro de retención empieza en MW100 y su longitud es de 100 palabras. Un offset de 20 determina que la dirección inicial del registro de retención sea 40021 en vez de 40001. Cualquier dirección inferior a 40021 o superior a 40119 tendrá como resultado un error de direccionamiento.

Tabla 13-70 Ejemplo de direccionamiento del registro de retención Modbus

HR_Start_Offset	Dirección	Mínimo	Máximo
0	Dirección Modbus (palabra)	40001	40099
	Dirección S7-1200	MW100	MW298
20	Dirección Modbus (palabra)	40021	40119
	Dirección S7-1200	MW100	MW298

HR_Start_Offset son datos de palabra en el bloque de datos de instancia de MB_SERVER que asigna la dirección inicial del registro de retención Modbus. Puede ajustar esta variable estática pública utilizando la lista desplegable de asistencia de parámetros, una vez que haya insertado MB_SERVER en el programa.

Por ejemplo, una vez que haya insertado MB_SERVER en un segmento KOP, puede ir a un segmento anterior y asignar HR_Start_Offset. La dirección inicial debe asignarse antes de ejecutar MB_SERVER.



Introducir una variable de servidor Modbus utilizando el nombre de DB predeterminado:

1. Posicione el cursor en el campo del parámetro y escriba un carácter m.
2. Seleccione "MB_SERVER_DB" en la lista desplegable de nombres de DB.
3. Seleccione "MB_SERVER_DB.HR_Start_Offset" en la lista desplegable de variables de DB.

Acceso a áreas de datos de bloques de datos (DB) en lugar de acceso directo a direcciones Modbus

A partir de la versión V5.0 de la instrucción MB_SERVER y la versión de firmware (FW) V4.2 de la CPU S7-1200 se puede acceder a áreas de datos de DB en lugar de acceder directamente a las memorias imagen de proceso y los registros de retención. Para hacerlo hay que deseleccionar las casillas de verificación "Depositar solo en la memoria de carga" y "Acceso optimizado al bloque" en la página de propiedades "Atributos" del DB global.

Si llega una petición Modbus y no ha definido un área de datos para el tipo de datos Modbus del código de función correspondiente, la instrucción MB_SERVER tratará la petición como en las versiones anteriores de la instrucción: se accede directamente a memorias imagen de proceso y registros de retención.

Si ha definido un área de datos para el tipo de datos Modbus del código de función, la instrucción MB_SERVER leerá o escribirá en esa área de datos. Si lee o escribe dependerá del tipo de orden.

Nota

Si se configura un área de datos, la instrucción MB_SERVER ignora los offsets o rangos configurados por las variables estáticas en el bloque de datos de instancia que corresponde al tipo de datos del área de datos. Dichos offsets y rangos solo son aplicables a la memoria imagen de proceso o a la memoria referenciada por MB_HOLD_REG. Los parámetros de inicio y longitud del área de datos proporcionan su propia forma de definir offsets y rangos.

Para una petición Modbus individual, solo se puede leer de o escribir en un área de datos. Si se desea, por ejemplo, leer registros de retención que se extienden a lo largo de varias áreas de datos, se requieren varias peticiones Modbus.

Estas son las normas para definir áreas de datos:

- Se pueden definir hasta ocho áreas de datos en distintos DB; cada DB solo puede contener un área de datos. Una petición MODBUS única solo puede leer en exactamente una única área de datos o escribir en exactamente un área de datos. Cada área de datos corresponde a un área de direcciones MODBUS. Las áreas de datos se definen en la variable estática "Data_Area_Array" del DB de instancia.
- Si desea utilizar menos de ocho áreas de datos, posicione las áreas de datos necesarias una detrás de otra sin dejar huecos. La primera entrada vacía de las áreas de datos finaliza la búsqueda del área de datos durante el procesamiento. Si define, por ejemplo, los elementos 1, 2, 4 y 5, "Data_Area_Array" solo reconocerá los elementos 1 y 2, mientras que el elemento 3 está vacío.

- El campo Data_Area_Array consta de ocho elementos: Data_Area_Array[1] a Data_Area_Array[8]
- Cada elemento de campo Data_Area_Array[x], 1 <= x <= 8, es un UDT del tipo MB_DataArea y está estructurado del siguiente modo:

Parámetro	Tipo de datos	Significado
data_ type	UInt	Identificador del tipo de datos MODBUS mapeado a esta área de datos: • 0: Identificador de un elemento de campo vacío o un área de datos no utilizada. En este caso, los valores de db, start y length son irrelevantes. • 1: Salida de la memoria imagen de proceso (utilizada con los códigos de función 1, 5 y 15) • 2: Entrada de la memoria imagen de proceso (utilizada con el código de función 2) • 3: Registro de retención (utilizado con los códigos de función 3, 6 y 16) • 4: Registro de entrada (utilizado con el código de función 4) Nota: Si ha definido un área de datos para un tipo de datos MODBUS, la instrucción MB_SERVER no podrá seguir accediendo directamente a este tipo de datos MODBUS. Si la dirección de una petición MODBUS para un tipo de datos de ese tipo no se corresponde con un área de datos definida, se devolverá un valor W#16#8383 en STATUS.
db	UInt	Número del bloque de datos al que se mapean el registro MODBUS o los bits definidos El número de DB debe ser único en las áreas de datos. El mismo número de DB no puede estar definido en varias áreas de datos. Hay que deseleccionar las casillas de verificación "Depositar solo en la memoria de carga" y "Acceso optimizado al bloque" en la página de propiedades "Atributos" del DB global. Las áreas de datos también comienzan por la dirección de byte 0 del DB. Valores permitidos: de 1 a 60999
start	UInt	Primera dirección MODBUS mapeada al bloque de datos comenzando por la dirección 0.0 Valores permitidos: de 0 a 65535
length	UInt	Número de bits (para los valores 1 y 2 de data_type) o el número de registros (para los valores 3 y 4 de data_type) Las áreas de direcciones MODBUS de un mismo tipo de datos MODBUS no pueden solaparse. Valores permitidos: de 1 a 65535

Ejemplos de definición de áreas de datos:

- Primer ejemplo: data_type = 3, db = 1, start = 10, length = 6
  La CPU mapea los registros de retención (data_type = 3) en el bloque de datos 1 (db = 1) y coloca la dirección Modbus 10 (start = 10) en la palabra de datos 0 y la última dirección Modbus válida 15 (length = 6) en la palabra de datos 5.
- Segundo ejemplo: data_type = 2, db = 15, start = 1700, length = 112
  La CPU mapea las entradas (data_type = 2) en el bloque de datos 15 (db = 15) y coloca la dirección Modbus 1700 (start = 1700) en la palabra de datos 0 y la última dirección Modbus válida 1811 (length = 112) en la palabra de datos 111.

Códigos de condición

Tabla 13-71 Códigos de condición de ejecución de MB_SERVER [1]

STATUS (W#16#)	Código de respuesta al servidor Modbus (B#16#)	Errores de protocolo Modbus
7001		MB_SERVER está esperando a que un cliente Modbus se conecte al puerto TCP asignado. Se devuelve este código en la primera ejecución de una operación de conexión o de desconexión.
7002		MB_SERVER está esperando a que un cliente Modbus se conecte al puerto TCP asignado. Se devolverá este código en las ejecuciones sucesivas, mientras se espera la finalización de una operación de conexión o de desconexión.
7003		Una operación de desconexión ha finalizado correctamente (válido solo para un ciclo del PLC).
8187		MB_HOLD_REG no es válido, podría apuntar a un DB optimizado o está apuntando a un área de menos de 2 bytes.
818C		Puntero MB_HOLD_REG a un área de DB no optimizada (debe ser un área de memoria M o DB global no optimizada) o el temporizador de procesos bloqueados supera el límite de 55 segundos. (específico de S7-1200)
8381	01	Código de función no soportado
8382	03	Error en la longitud de datos: • Especificación de longitud no válida en la trama Modbus recibida. • La longitud de trama introducida en el encabezado de la trama Modbus no concuerda con el número de bytes recibidos realmente. • El número de bytes introducido en el encabezado de la trama Modbus no concuerda con el número de bytes recibidos realmente (funciones 15 y 16).
8383	02	Error en la dirección de los datos o acceso fuera de los límites del área de direcciones de MB_HOLD_REG
8384	03	Error de valor de datos

STATUS (W#16#)	Código de respuesta al servidor Modbus (B#16#)	Errores de protocolo Modbus
8385	03	Código de diagnóstico de datos no soportado (código de función 08)
8389		Definición no válida del área de datos: • Valor de tipo de datos no válido • Número de DB no válido o no existe: – Valor de DB no válido – El número de DB no existe – El número de DB ya se utiliza en otra área de datos – DB con acceso optimizado – El DB no está en la memoria de trabajo • Valor de longitud no válido • Solapamiento de rangos de direcciones MODBUS que pertenecen al mismo tipo de datos MODBUS

[1] Además de los errores de MB_SERVER indicados, las instrucciones de comunicaciones del bloque T subyacente (TCON, TDISCON, TSEND y TRCV) pueden devolver errores.

MB_RED_CLIENT (comunicación redundante vía PROFINET como cliente Modbus TCP)

Esta instrucción puede utilizarse para establecer una conexión entre una CPU S7-1200 y un dispositivo que soporte el protocolo Modbus TCP.

Tabla 13-72 Instrucción MB_RED_CLIENT

KOP / FUP	SCL	Descripción

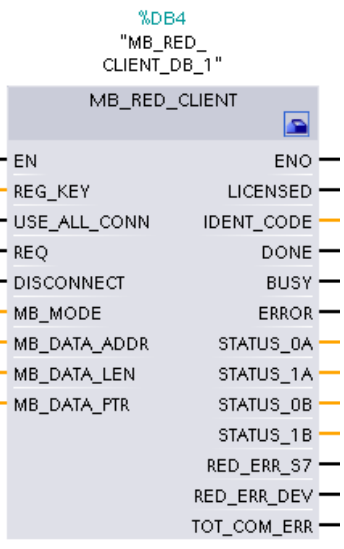
	`"MB_RED_CLIENT_DB"(`
	`REG_KEY:=_string_in_,`
	`USE_ALL_CONN:=_bool_in_`
	`REQ:=_bool_in_,`
	`DISCONNECT:=_bool_in_,`
	`MB_MODE:=_usint_in_,`
	`MB_DATA_ADDR:=_udint_in_,`
	`MB_DATA_LEN:=_uint_in_,`
	`LICENSED=>_bool_out_`
	`IDENT_CODE=>_string_out_`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`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_);`
		La instrucción MB_RED_CLIENT se comunica como cliente Modbus TCP a través de la conexión PROFINET.
		La instrucción MB_RED_CLIENT se utiliza para establecer una conexión redundante entre el cliente y el servidor, enviar órdenes Modbus, recibir respuestas y controlar la finalización de la conexión desde el cliente Modbus TCP.

Tabla 13-73 Tipos de datos de los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REG_KEY [1]	IN	STRING[17]	Código de registro para la licencia La instrucción MB_RED_CLIENT necesita una licencia en cada CPU.
USE_ALL_CONN	IN	Bool	Especifique el número de conexiones configuradas por las que debe enviarse la trama: • 0: enviar trama por una conexión, conmutar a la conexión siguiente solo en caso de error • 1: enviar trama por todas las conexiones configuradas
REQ	IN	Bool	Consulta Modbus al servidor Modbus TCP El parámetro REQ está controlado por niveles. Esto significa que mientras la entrada esté activada (REQ = TRUE), la instrucción enviará peticiones de comunicación. Si la conexión todavía no se ha establecido, se establece ahora y la trama Modbus se envía de inmediato. Los cambios en los parámetros de entrada no serán efectivos hasta que el servidor haya respondido o se haya emitido un mensaje de error. Si el parámetro REQ se activa de nuevo durante una petición Modbus entrante, no se lleva a cabo ninguna transmisión adicional posterior.
DISCONNECT	IN	Bool	Con este parámetro se controla el establecimiento y la finalización de la conexión con el servidor Modbus: • 0: establecer conexión de comunicación con el interlocutor configurado en el parámetro CONNECT (consulte el parámetro CONNECT). • 1: desconectar la conexión de comunicación. No se ejecuta ninguna otra función mientras finaliza la conexión. El valor 0003 se genera en el parámetro STATUS_x una vez finalizada correctamente la conexión.
MB_MODE [2]	IN	USInt	Selecciona el modo de la petición Modbus (lectura, escritura o diagnóstico) o selección directa de una función Modbus
MB_DATA_ADDR [2]	IN	UDInt	Dirección Modbus en función de MB_MODE
MB_DATA_LEN	IN	UInt	Longitud de datos: número de bits o registros para el acceso de datos
MB_DATA_PTR [2]	IN_OUT	Variant	Puntero a un búfer de datos para los datos que deben recibirse del servidor Modbus o que deben enviarse al servidor Modbus.
LICENSED [1]	OUT	Bool	• 0: la instrucción no tiene licencia. • 1: la instrucción tiene licencia.
IDENT_CODE [1]	OUT	STRING[18]	Identificador de la licencia. Utilice este string para solicitar el código de registro REG_KEY.
DONE	OUT	Bool	El bit en el parámetro de salida DONE se pone a "1" en cuanto la tarea Modbus activada finaliza sin errores en una conexión como mínimo.
BUSY	OUT	Bool	• 0: no hay ninguna petición Modbus en curso. • 1: se está procesando una petición Modbus. El parámetro de salida BUSY no se activa durante el establecimiento y la finalización de la conexión.
ERROR	OUT	Bool	• 0: no hay error • 1: la tarea Modbus activada no pudo transmitirse correctamente en alguna de las conexiones configuradas. La causa del error se indica en el parámetro STATUS_x.
STATUS_0A [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 0A.

Parámetro y tipo		Tipo de datos	Descripción
STATUS_1A [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 1A.
STATUS_0B [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 0B.
STATUS_1B [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 1B.
RED_ERR_S7 [3]	OUT	Bool	• 0: no hay errores de redundancia en SIMATIC. • 1: error de redundancia en SIMATIC.
RED_ERR_S7 [3]	OUT	Bool	• 0: no hay errores de redundancia en el lado del interlocutor del enlace. • 1: error de redundancia en el lado del interlocutor del enlace.
RED_ERR_S7 [3]	OUT	Bool	• 0: se ha establecido por lo menos una conexión configurada. • 1: pérdida total de la comunicación, todas las conexiones configuradas han finalizado.

[1] Encontrará más información en el apartado "Obtener una licencia" más adelante.

[2] Encontrará más información en el apartado "Parámetros de entrada: MB_MODE, MB_DATA_ADDR, MB_DATA_LEN y MB_DATA_PTR" más adelante.

[3] Encontrará más información en el apartado "Parámetros de salida: STATUS_x, RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR" más adelante.

Nota

Datos de entrada coherentes durante una llamada de MB_RED_CLIENT

Cuando se llama una instrucción de cliente Modbus, los valores de los parámetros de entrada se guardan internamente. Los valores no tienen que cambiarse mientras se procesa la trama.

Nota

Requisitos de la versión del firmware de la CPU

Las instrucciones Modbus TCP descritas en este apartado del manual requieren una versión de firmware V4.2 o superior.

Para usar la instrucción no se requiere un módulo de hardware adicional.

Varias conexiones de cliente

Las CPU pueden procesar varias conexiones de cliente Modbus TCP. El número máximo de conexiones depende de la CPU utilizada y puede consultarse en los datos técnicos de la CPU. El número total de conexiones de una CPU, incluidas las de los clientes y el servidor Modbus TCP, no debe sobrepasar el número máximo de conexiones soportadas.

Con conexiones de cliente individuales deben recordarse las reglas siguientes:

- Cada conexión MB_RED_CLIENT debe usar un DB de instancia exclusivo.
- Para cada conexión MB_RED_CLIENT debe especificarse una dirección IP de servidor unívoca.
- Cada conexión MB_RED_CLIENT requiere una ID de conexión unívoca. La ID de conexión debe ser unívoca en toda la CPU.

Operación y redundancia

Los nodos de comunicación pueden diseñarse como autónomos o redundantes. Si uno de los interlocutores está diseñado como autónomo, nos referiremos a ello como redundancia unilateral. Si ambos interlocutores están diseñados como redundantes, lo designaremos como redundancia bilateral:

- Redundancia unilateral:
  - Descripción: debe configurarse una conexión en cada caso para cada conexión entre los interlocutores. Los puntos de conexión de **SIMATIC S7** se denominan **0** y **1**; los puntos de conexión del **interlocutor de la comunicación** se denominan **A** y **B**. La CPU R o CPU H 1 recurre al punto de conexión 0, mientras que la CPU R o CPU H 2 recurre al punto de conexión 1.
  - Configuración: si el S7 está diseñado de forma redundante, se crea una conexión desde el punto de conexión 0 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A**) y otra conexión desde el punto de conexión 1 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **A** => conexión **1A**). La figura muestra las denominaciones de las conexiones:

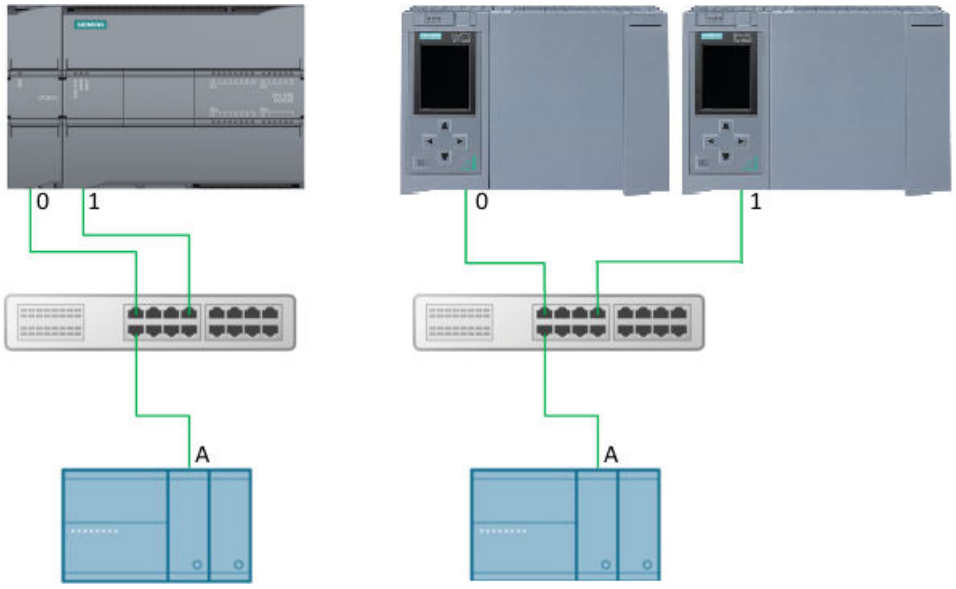


Figura 13-1 Redundancia unilateral S7

– Si el S7 está diseñado como autónomo y el interlocutor está diseñado de forma redundante, se crea una conexión desde el punto de conexión 0 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A**) y otra conexión desde el punto de conexión 0 del S7 al punto nodal B del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **B** => conexión **0B**). La figura muestra las denominaciones de las conexiones:

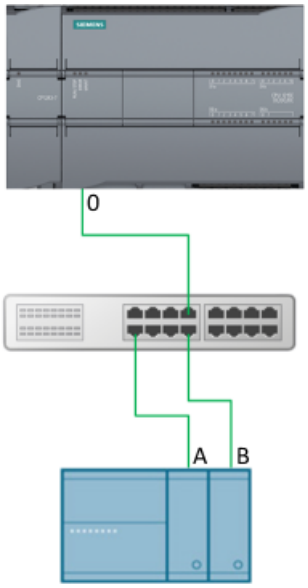


Figura 13-2 Interlocutor de redundancia unilateral

- Redundancia bilateral:
  – Descripción: debe configurarse una conexión en cada caso para cada conexión entre los interlocutores. Los puntos de conexión del SIMATIC S7 se denominan **0** y **1**; los puntos de conexión del **interlocutor de la comunicación** se denominan **A** y **B**.
  La CPU R o CPU H 1 recurre al punto de conexión 0, mientras que la CPU R o CPU H 2 recurre al punto de conexión 1.

– Configuración: en el caso de la redundancia bilateral, se crean dos conexiones desde el punto de conexión 0 (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A** y conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **B** => conexión **0B**) y dos conexiones desde el punto de conexión 1 del S7 a los puntos nodales A y B del interlocutor (conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **A** => conexión **1A** y conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **B** => conexión **1B**). La figura muestra las denominaciones de las conexiones:

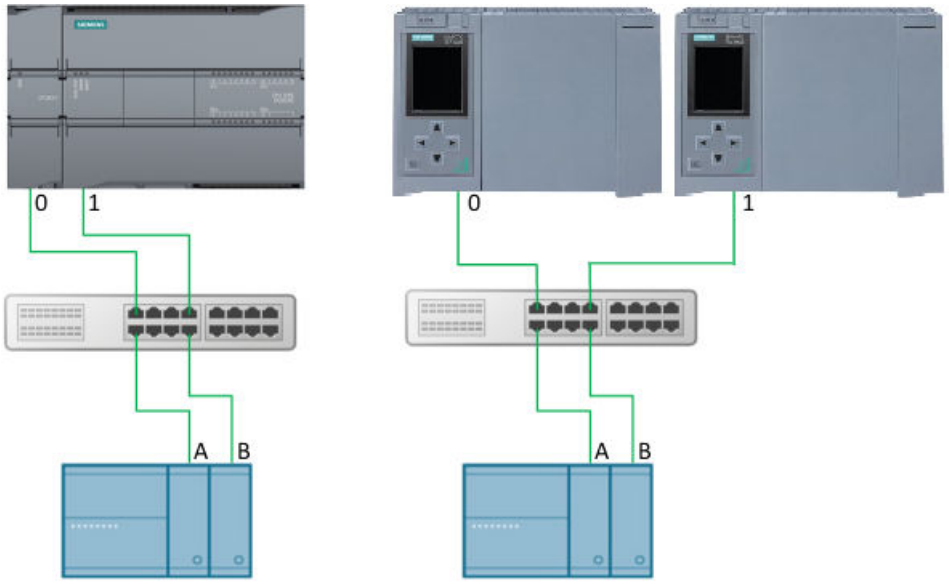


Figura 13-3 Redundancia bilateral

- Procesamiento de tramas: las tramas pueden enviarse por una o todas las conexiones configuradas:
  - Enviar tramas por una conexión: la trama MODBUS se envía a través de una conexión (la que está actualmente activa) con el ajuste USE_ALL_CONN = FALSE. En caso de que se exceda el tiempo (no hay respuesta del servidor) o de fallo de conexión, se intenta enviar la trama utilizando otras conexiones configuradas (como máximo 4). En este caso la secuencia es 0A, 1A, 0B y 1B. Si una trama se transmite correctamente por una conexión, dicha conexión se marca como "activa" y el tráfico de tramas posterior se ejecuta por esta conexión. En caso de que falle la conexión activa, se intenta enviar de nuevo la trama por todas las conexiones configuradas. Si todos los intentos de transmisión fallan, ERROR y STATUS_x se ajustan en correspondencia.
    Si se ha recibido una trama de respuesta se ejecuta un control de plausibilidad. Si el control es correcto se llevan a cabo las acciones requeridas y la tarea se ejecuta sin errores; la salida DONE se activa. Si se detectan errores durante el control, la tarea finaliza sin errores, el bit ERROR se activa y se muestra un número de error en STATUS_x. En este caso no se intenta volver a enviar la trama en la próxima conexión configurada. Solo se conmuta a las otras conexiones configuradas si se detecta un fallo de conexión o no se recibe respuesta alguna.
  - Enviar tramas por todas las conexiones: la trama MODBUS se envía por todas las conexiones configuradas y establecidas con el ajuste USE_ALL_CONN = TRUE. Se lleva a cabo un control de plausibilidad cuando se ha recibido la respuesta en una de las conexiones. Si el control es correcto se llevan a cabo las acciones requeridas. Si se ha recibido una trama de respuesta válida en por lo menos una conexión, la salida DONE se activa.
- Salidas de redundancia RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR:

– Los bits de redundancia RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR se activan en función de los estados de las salidas de estado:

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

Figura 13-4 Visualización de los bits de alarma para la configuración de redundancia en ambos lados

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

Figura 13-5 Visualización de los bits de alarma para la configuración de redundancia en un lado

Nota

Números de puerto para cliente y servidor

El cliente Modbus utiliza un número de puerto que empieza por 2000. El servidor Modbus suele direccionarse a través del número de puerto 502.

Parametrización

Puede utilizarse la instrucción MB_RED_CLIENT **V1.0** y **V1.1** para el S7-1200. La CPU implementa las conexiones a través de la interfaz local de la CPU o el CM/CP. La CPU configura y establece las conexiones utilizando la estructura TCON_IP_V4.

Configuración de MB_RED_CLIENT: Los ajustes siguientes se realizan en el cuadro de diálogo de configuración de la instrucción MB_RED_CLIENT:

- Parámetros de las conexiones 0A, 1A, 0B y 1B (encontrará más información sobre la configuración de la redundancia en el apartado anterior "Operación y redundancia").
- Parámetro interno (opcional)

El cuadro de diálogo de configuración se abre con la instrucción MB_RED_CLIENT o mediante objetos tecnológicos:

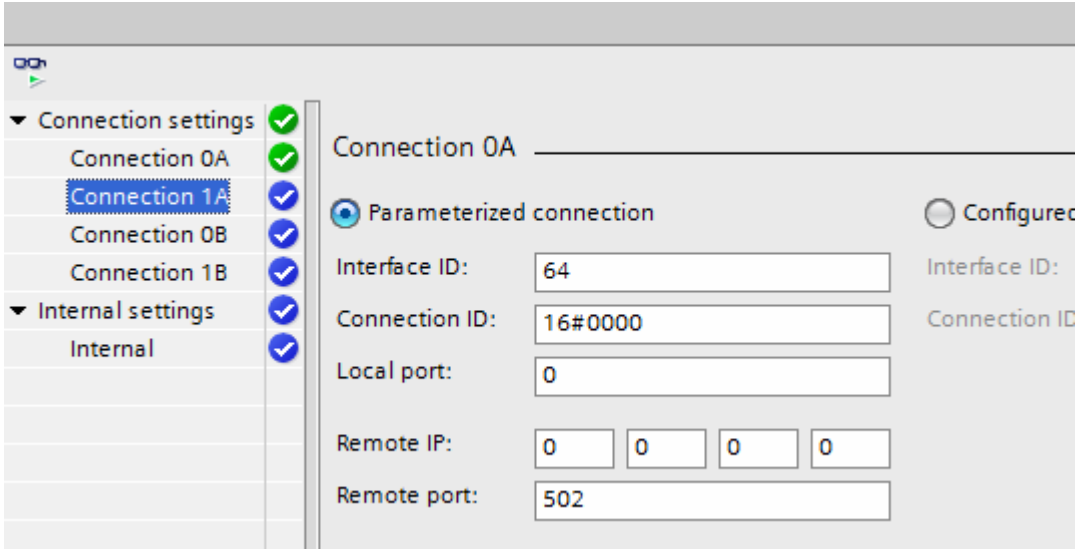


Figura 13-6 Conexión de cliente parametrizada

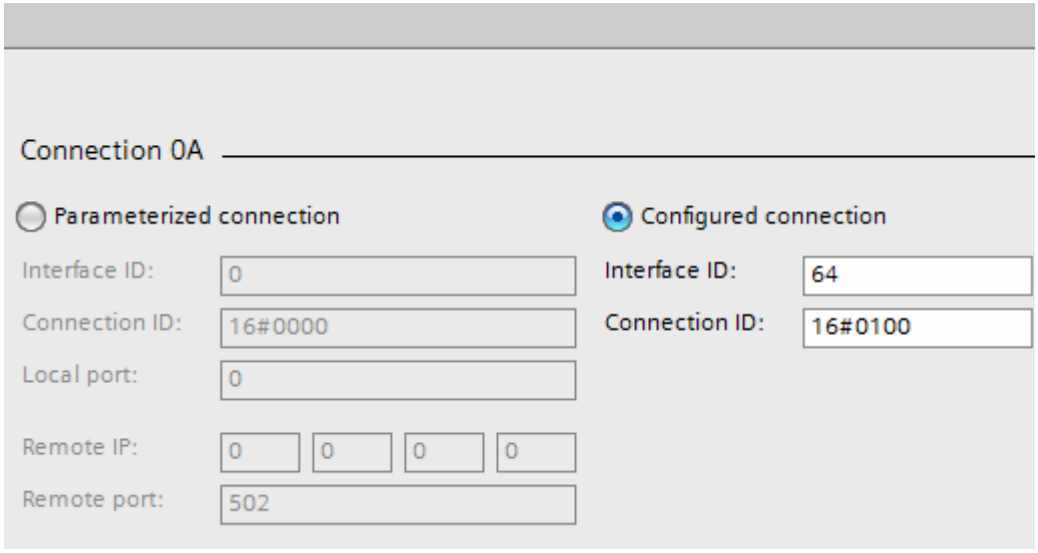


Figura 13-7 Conexión de cliente configurada

Variable	Valor inicial	Descripción
Conexiones configuradas		
Interface ID	64	Identificador de hardware de la interfaz PN utilizada
Connection ID	16#0000	ID de las conexiones utilizadas Estos ID de conexión deben ser unívocas en toda la CPU.
Local port	0	Número de puerto local del cliente. De forma predeterminada no se introduce ningún número de puerto para el cliente.
Remote IP	0.0.0.0	Dirección IP remota del servidor
Remote port	502	Número de puerto remoto del servidor El puerto predeterminado para el servidor Modbus/TCP es el 502.
Conexiones configuradas		
Interface ID	64	Identificador de hardware de la interfaz PN utilizada
Connection ID	16#0000	ID de las conexiones utilizadas Estas conexiones se configuran en la vista de redes.

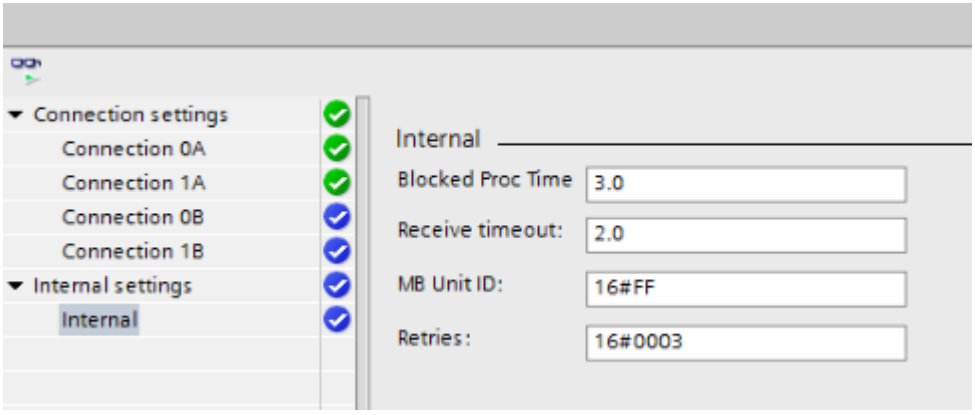


Figura 13-8 Parámetro interno (opcional)

Variable	Tipo de datos	Valor inicial	Descripción
Blocked Proc Time	REAL	3.0	Tiempo de espera en segundos antes de que la variable estática ACTIVE se desactive si hay una instancia Modbus bloqueada. Esto puede suceder, por ejemplo, si se genera una petición de cliente y la ejecución de la función de cliente se cancela antes de que se haya terminado de ejecutar la petición. El tiempo de espera debe estar comprendido entre 0,5 y 55 s.
Receive timeout	REAL	2.0	Intervalo de tiempo en segundos en el que la instrucción "MB_RED_CLIENT" espera una respuesta del servidor. Debe estar comprendido entre 0,5 y 55 s.

Variable	Tipo de datos	Valor inicial	Descripción
MB_Unit_ID	BYTE	255	Detección de dispositivos Modbus: Un servidor Modbus TCP se direcciona a través de su dirección IP. Por este motivo, el parámetro MB_UNIT_ID no se utiliza para el direccionamiento de Modbus TCP. El parámetro MB_UNIT_ID corresponde al campo de dirección de esclavo en el protocolo Modbus RTU. Si un servidor Modbus/TCP se usa como pasarela para un protocolo Modbus RTU, MB_UNIT_ID puede usarse para identificar el dispositivo esclavo en la red serie. En este caso, el parámetro MB_UNIT_ID se usaría para reenviar la petición a la dirección del esclavo Modbus RTU correcta. Tenga en cuenta que algunos dispositivos Modbus/TCP pueden necesitar el parámetro MB_UNIT_ID para la inicialización dentro de un rango de valores limitado.
Retries	WORD	3	Número de intentos de transmisión realizados por la instrucción MB_RED_CLIENT antes de que devuelva el error W#16#80C8.

Nota

Variable MB_Transaction_ID

Si la ID de transacción en la respuesta del servidor Modbus TCP no concuerda con la ID de transacción de la tarea de MB_RED_CLIENT, la instrucción MB_RED_CLIENT espera la respuesta del servidor Modbus TCP con la ID de transacción correcta durante el tiempo RCV_TIMEOUT * RETRIES; una vez transcurrido dicho tiempo, devuelve el error W#16#80C8.

Obtener una licencia

La instrucción MB_RED_CLIENT está sujeta a una tasa y hay que obtener una licencia individualmente para cada CPU. Para obtener una licencia deben seguirse estos dos pasos:

- Visualizar la licencia IDENT_CODE
- Introducir la clave de registro REG_KEY: hay que asignar la clave de registro REG_KEY en cada instrucción MB_RED_CLIENT. Guarde la REG_KEY en un bloque de datos global desde el que todas las instrucciones MB_RED_CLIENT reciban la clave de registro que necesitan.

Procedimiento para visualizar la licencia IDENT_CODE:

1. Parametrice la instrucción MB_RED_CLIENT de acuerdo con sus necesidades en un OB cíclico. Descargue el programa en la CPU y ponga la CPU en RUN.
2. Abra el DB de instancia de la instrucción Modbus y haga clic en el botón "Observar todos".

3. El DB de instancia muestra una cadena de caracteres de 18 dígitos en la salida IDENT_CODE.

	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

Figura 13-9 Licencia

4. Copie esta cadena utilizando la función de copiar y pegar desde el bloque de datos y péguela en el formulario (que se le envió por correo electrónico tras pedir el producto o que está incluido en el CD).
5. Envíe el formulario a Customer Support (https://support.industry.siemens.com/my/ww/en/requests/#createRequest) por medio de una solicitud de servicio. A continuación recibirá la clave de registro para su CPU.

Procedimiento para introducir la clave de registro REG_KEY:

1. Inserte un nuevo bloque de datos global con un nombre simbólico unívoco, por ejemplo "License_DB", mediante "Agregar nuevo bloque...".
2. Cree un parámetro REG_KEY en este bloque con el tipo de datos STRING[17].

LICENSE_DB

	Name	Data type	Offset	Start value
1	Static			
2	REG_KEY	String[17]	0.0	''

Figura 13-10 REG KEY

3. Copie la clave de registro de 17 dígitos que se ha transmitido a la columna "Valor de arranque" con la función de copiar y pegar.
4. En el OB cíclico, introduzca el nombre del DB de licencia y el nombre de la cadena (por ejemplo, License_DB.REG_KEY) en el parámetro REG_KEY de la instrucción MB_RED_CLIENT.
5. Descargue los bloques modificados en la CPU. Puede introducir la clave de registro en runtime; no es necesario pasar de STOP a RUN.
6. La comunicación Modbus/TCP por medio de la instrucción MB_RED_CLIENT ya tiene licencia para esta CPU, el bit de salida LICENSED es TRUE.

Procedimiento para corregir la falta de licencia o una licencia incorrecta:

- Si se introduce una clave de licencia incorrecta o no se introduce ninguna clave, el ERROR LED de la CPU parpadea. Además, en el S7-1200 la CPU realiza una entrada cíclica en el búfer de diagnóstico en relación con la falta de licencia.

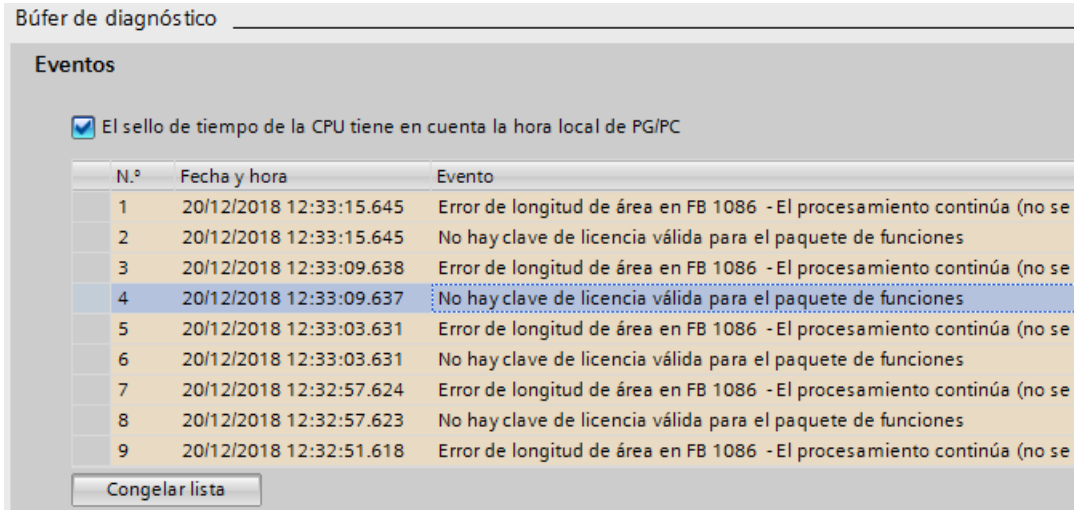


Figura 13-11 Búfer de diagnóstico

- En caso de que falte la clave de registro o que esta sea incorrecta, la CPU procesa la comunicación Modbus TCP; sin embargo, la CPU siempre indica "W#16#0A90" (no hay clave de licencia válida para paquete funcional) en la salida STATUS_x. El bit de salida LICENSED es FALSE.

Parámetros de entrada MB_MODE, MB_DATA_ADDR, MB_DATA_LEN y MB_DATA_PTR

La combinación de los parámetros MB_MODE, MB_DATA_ADDR y MB_DATA_LEN define el código de función utilizado en el mensaje Modbus actual:

- **MB_MODE** contiene la información sobre si debe leerse o escribirse:
  Lectura: MB_MODE = 0, 101, 102, 103 y 104
  Escritura: MB_MODE = 1, 2, 105, 106, 115 y 116 (nota: si MB_MODE = 2 no se distingue entre las funciones Modbus 15 y 05 o entre las funciones Modbus 16 y 06.)
- **MB_DATA_ADDR** contiene información sobre lo que debe leerse o escribirse, además de información de la dirección desde la que la instrucción MB_RED_CLIENT calcula la dirección remota.
- **MB_DATA_LEN** contiene el número de valores que debe leerse o escribirse.

En la tabla siguiente se indica la relación entre los parámetros de entrada MB_MODE, MB_DATA_ADDR y MB_DATA_LEN de la instrucción MB_RED_CLIENT y la función Modbus:

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Función Modbus	Función y tipo de datos
0	de 1 a 9.999	de 1 a 2.000	01	Leer entre 1 y 2.000 bits de salida en la dirección remota de 0 a 9.998
0	de 10.001 a 19.999	de 1 a 2.000	02	Leer entre 1 y 2.000 bits de entrada en la dirección remota de 0 a 9.998

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Función Modbus	Función y tipo de datos
0	• de 40.001 a 49.999 • de 400.001 a 465.535	de 1 a 125	03	• Leer entre 1 y 125 registros de retención en la dirección remota de 0 a 9.998 • Leer entre 1 y 125 registros de retención en la dirección remota de 0 a 65.534
0	de 30.001 a 39.999	de 1 a 125	04	Leer entre 1 y 125 palabras de entrada en la dirección remota de 0 a 9.998
1	de 1 a 9.999	1	05	Escribir 1 bit de salida en la dirección remota de 0 a 9.998
1	• de 40.001 a 49.999 • de 400.001 a 465.535	1	06	• Escribir 1 registro de retención en la dirección remota de 0 a 9.998 • Escribir 1 registro de retención en la dirección remota de 0 a 65.534
1	de 1 a 9.999	de 2 a 1.968	15	Leer entre 2 y 1.968 bits de salida en la dirección remota de 0 a 9.998
1	• de 40.001 a 49.999 • de 400.001 a 465.535	de 2 a 123	16	• Escribir entre 2 y 123 registros de retención en la dirección remota de 0 a 9.998 • Escribir entre 2 y 123 registros de retención en la dirección remota de 0 a 65.534
2	de 1 a 9.999	de 1 a 1.968	15	Escribir entre 1 y 1.968 bits de salida en la dirección remota de 0 a 9.998
2	• de 40.001 a 49.999 • de 400.001 a 465.535	de 1 a 123	16	• Escribir entre 1 y 123 registros de retención en la dirección remota de 0 a 9.998 • Escribir entre 1 y 123 registros de retención en la dirección remota de 0 a 65.534
11	La instrucción no evalúa los parámetros MB_DATA_ADDR y MB_DATA_LEN cuando se ejecuta esta función.		11	Leer la palabra de estado y el contador de eventos del servidor: • La palabra de estado refleja el estado de procesamiento (0 - no se procesa, 0xFFFF - se procesa). • El contador de eventos se incrementa cuando la orden Modbus se ha ejecutado correctamente. Si se ha producido un error durante la ejecución de una función Modbus, el servidor envía un mensaje pero el contador de eventos no se incrementa.

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Función Modbus	Función y tipo de datos
80	-	1	08	Comprobar el estado del servidor con el código de diagnóstico 0x0000 (prueba de bucle de retorno - el servidor devuelve la orden): 1 WORD por llamada
81	-	1	08	Inicializar el contador de eventos del servidor con el código de diagnóstico 0x000A: 1 WORD por llamada
101	de 0 a 65.535	de 1 a 2.000	01	Leer entre 1 y 2.000 bits de salida en la dirección remota de 0 a 65.535
102	de 0 a 65.535	de 1 a 2.000	02	Leer entre 1 y 2.000 bits de entrada en la dirección remota de 0 a 65.535
103	de 0 a 65.535	de 1 a 125	03	Leer entre 1 y 125 registros de retención en la dirección remota de 0 a 65.535
104	de 0 a 65.535	de 1 a 125	04	Leer entre 1 y 125 palabras de entrada en la dirección remota de 0 a 65.535
105	de 0 a 65.535	1	05	Escribir 1 bit de salida en la dirección remota de 0 a 65.535
106	de 0 a 65.535	1	06	Escribir 1 registro de retención en la dirección remota de 0 a 65.535
115	de 0 a 65.535	de 1 a 1.968	15	Escribir entre 1 y 1.968 bits de salida en la dirección remota de 0 a 65.535
116	de 0 a 65.535	de 1 a 123	16	Escribir entre 1 y 123 registros de retención en la dirección remota de 0 a 65.535
de 3 a 10, de 12 a 79, de 82 a 100, de 107 a 114, de 117 a 255				Reservado

Ejemplo:

Variable	Significado
MB_MODE = 1 MB_DATA_ADDR = 1 MB_DATA_LEN = 1	Escribe 1 bit de salida con el código de función 5, empezando por la dirección remota 0.
MB_MODE = 1 MB_DATA_ADDR = 1 MB_DATA_LEN = 2	Escribe 2 bits de salida con el código de función 15, empezando por la dirección remota 0.
MB_MODE = 104 MB_DATA_ADDR = 17834 MB_DATA_LEN = 125	Lee 125 palabras de entrada con el código de función 4, empezando por la dirección remota 17.834.

MB_DATA_PTR:

El parámetro MB_DATA_PTR es un puntero a un búfer de datos para los datos que deben recibirse del servidor Modbus o que deben enviarse al servidor Modbus. Es posible utilizar un bloque de datos global o un área de memoria (M) como búfer de datos.

Para un búfer en el área de memoria (M) debe utilizarse un puntero en formato ANY, tal como se indica a continuación: "P#dirección de bit" "tipo de datos" "longitud" (ejemplo: P#M1000.0 WORD 500)

Dependiendo del área de memoria en la que se encuentre el búfer de datos, MB_DATA_PTR puede hacer referencia a diferentes estructuras de datos:

- Cuando se utiliza un DB global con acceso optimizado, MB_DATA_PTR puede hacer referencia a una variable con un tipo de datos simple o un array de tipos de datos simples. Se soportan los siguientes tipos de datos:

Tipo de datos	Longitud en bits
Bool	1
Byte, SInt, USInt, Char	8
Word, Int, WChar, UInt	16
DWord, DInt, UDInt, Real	32

  Todos los tipos de datos soportados pueden utilizarse para todas las funciones Modbus. Así, por ejemplo, MB_RED_CLIENT también puede escribir un bit recibido en una variable del tipo byte en una dirección concreta sin cambiar otros bits de dicho byte. Por consiguiente, no es necesario tener un array de bits para ejecutar funciones orientadas a bits.

- Si se utiliza un área de marcas o un DB global con acceso estándar como área de memoria, ya no hay restricciones en cuanto a los tipos de datos simples para MB_DATA_PTR; así pues, MB_DATA_PTR puede hacer referencia también a estructuras de datos complejas como tipos de datos PLC (UDT) y tipos de datos de sistema (SDT).

Nota

Utilizar un área de marcas como búfer de datos

Si se utiliza un área de marcas como búfer de datos para MB_DATA_PTR, es necesario observar esta variable. En el caso de las CPU SIMATIC S7-1200 CPU, son 8 KB.

Parámetros de salida: STATUS_x, RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR

La CPU visualiza mensajes de error en las salidas de estado de la instrucción MB_RED_CLIENT :

Nota

Los códigos de estado de los errores pueden visualizarse en el editor de programas como valor entero o hexadecimal:

1. Abra el bloque deseado en el editor de programas.
2. Active el estado de programación haciendo clic en "Activar/desactivar vigilancia". (Si todavía no ha establecido una conexión online, se abre el cuadro de diálogo "Establecer conexión online". En este cuadro de diálogo puede establecer una conexión online.)
3. Seleccione la variable que desee vigilar y seleccione el formato de visualización deseado en "Formato de visualización" del menú contextual.

- Parámetro STATUS_x (información de estado general):

STATUS (W#16#)	Descripción
0000	Instrucción ejecutada sin errores.
0001	Conexión establecida.
0003	Conexión deshecha.
0A90	La instrucción MB_RED_CLIENT no tiene licencia. Encontrará más información en el apartado "Obtener una licencia" más arriba.
0AFF	La conexión no está configurada y no se utiliza. La conexión "0A" debe configurarse.
7000	No hay ninguna tarea activa y no se ha establecido ninguna conexión (REQ= 0, DISCONNECT= 1).
7001	Se ha iniciado el establecimiento de la conexión.
7002	Llamada intermedia. Se está estableciendo la conexión.
7003	Se está deshaciendo la conexión.
7004	Conexión establecida y vigilada. No se está procesando ninguna tarea.
7005	Se están transmitiendo datos.
7006	Se están recibiendo datos.

- Parámetro STATUS_x (error de protocolo)

STATUS (W#16#)	Descripción
80C8	No hay respuesta del servidor en el período definido. Compruebe la conexión con el servidor Modbus. Este error solo se notifica una vez realizados los intentos de repetición configurados. Si la instrucción MB_RED_CLIENT no recibe una respuesta con el ID de transacción transferido originalmente (consulte la variable estática MB_TRANSACTION_ID) dentro del período definido, se emite este código de error.
8380	La trama Modbus recibida tiene un formato incorrecto o no se han recibido suficientes bytes.
8382	• La longitud de la trama Modbus en el encabezado no concuerda con el número de bytes recibidos. • El número de bytes no concuerda con el número de bytes realmente transferidos (solo las funciones 1-4). • La dirección de inicio en la trama recibida no concuerda con la dirección de inicio guardada (funciones 5, 6, 15 y 16). • El número de palabras no concuerda con el número de palabras realmente transferidas (funciones 15 y 16).
8383	Error al leer o escribir datos o acceso fuera del área de direcciones de MB_DATA_PTR. Encontrará más información en el apartado "MB_DATA_PTR" más arriba.
8384	• Se ha recibido un código de excepción no válido. • Se ha recibido un valor de datos distinto del enviado originalmente por el cliente (funciones 5, 6 y 8). • Se ha recibido un valor de estado no válido (función 11).
8385	• Código de diagnóstico no soportado. • Se ha recibido un código de subfunción distinto del enviado originalmente por el cliente (función 8).

STATUS (W#16#)	Descripción
8386	El código de función recibido no concuerda con el enviado originalmente.
8387	El ID de protocolo de la trama Modbus TCP recibida por el servidor no es "0".
8388	El servidor Modbus ha enviado una longitud de datos distinta de la procesada. Este error se produce solo cuando se utilizan las funciones Modbus 5, 6, 15 o 16.

- Parámetro STATUS_x (error de parámetro)

STATUS (W#16#)	Descripción
80B6	Tipo de conexión no válido; solo se soportan conexiones TCP.
80BB	El parámetro ActiveEstablished tiene un valor no válido. Solo se permite la iniciativa local para el cliente (ActiveEstablished = TRUE).
8188	El parámetro MB_MODE tiene un valor no válido.
8189	Direccionamiento de datos no válido en el parámetro MB_DATA_ADDR
818A	Longitud de datos no válida en el parámetro MB_DATA_LEN
818B	El parámetro MB_DATA_PTR tiene un puntero no válido. También deberían comprobarse los valores de los parámetros MB_DATA_ADDR y MB_DATA_LEN. (Encontrará más información sobre "MB_DATA_ADDR" en el apartado "MB_DATA_ADDR" más arriba.)
818C	Tiempo excedido en el parámetro BLOCKED_PROC_TIMEOUT o RCV_TIMEOUT (ver variables estáticas de la instrucción). BLOCKED_PROC_TIMEOUT y RCV_TIMEOUT deben estar entre 0,5 s y 55,0 s.
8200	• La CPU está procesando una orden Modbus distinta a través del puerto. • Otra instancia de MB_RED_CLIENT con los mismos parámetros de conexión está procesando una orden Modbus existente.

Nota

Códigos de error de instrucciones de comunicación usadas internamente

Con la instrucción MB_RED_CLIENT, además de los errores listados en las tablas, pueden producirse errores provocados por las instrucciones de comunicación (TCON, TDISCON, TSEND, TRCV, T_DIAG y TRESET) que utiliza la instrucción.

La CPU asigna los códigos de error por medio del bloque de datos de instancia de la instrucción MB_RED_CLIENT. La CPU visualiza los códigos de error para la instrucción en cuestión en STATUS, en la sección "Static".

El significado de los códigos de error se explica en la documentación de la correspondiente instrucción de comunicación.

Nota

Error de comunicación al enviar o recibir datos

Si se produce un error de comunicación al enviar o recibir datos (80C4 (Error de comunicación temporal.) La conexión especificada está interrumpida temporalmente.), 80C5 (El partner remoto ha deshecho activamente la conexión.) y 80A1 (La conexión especificada está desconectada o todavía no se ha establecido.)), la CPU finaliza la conexión existente.

Esto significa que todos los valores de STATUS devueltos pueden verse cuando finaliza la conexión y que el código de STATUS que ha provocado la finalización de la conexión solo se emite cuando esta ya ha finalizado.

Ejemplo: si se produce un error de comunicación cuando se reciben datos, primero se emite el STATUS 7003 (ERROR=false) y luego 80C4 (ERROR=true).

MB_RED_SERVER (comunicación vía PROFINET como servidor Modbus TCP)

Esta instrucción puede utilizarse para establecer una conexión entre una CPU S7-1200 y un dispositivo que soporte el protocolo Modbus TCP.

Tabla 13-74 Instrucción MB_RED_SERVER

KOP / FUP	SCL	Descripción

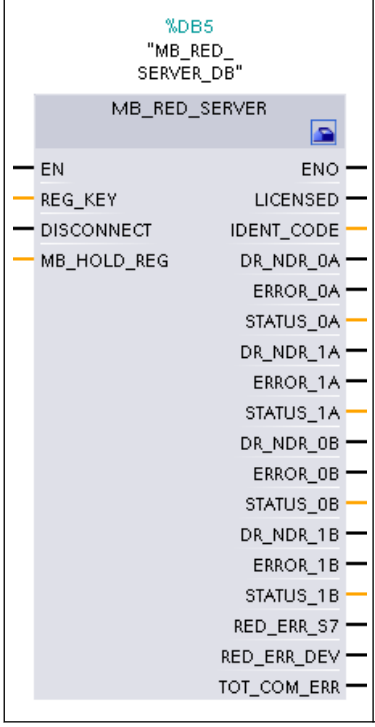
	`"MB_RED_SERVER_DB"(`
	`DISCONNECT:=_bool_in_,`
	`LICENSED=>_bool_out_`
	`IDENT_CODE=>_string_out_`
	`DR_NDR_0A=>_bool_out,`
	`ERROR_0A=>_bool_out,`
	`STATUS_0A=>_word_out_,`
	`DR_NDR_1A=>_bool_out,`
	`ERROR_1A=>_bool_out,`
	`STATUS_1A=>_word_out_,`
	`DR_NDR_0B=>_bool_out,`
	`ERROR_0B=>_bool_out,`
	`STATUS_0B=>_word_out_,`
	`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_);`
		La instrucción MB_RED_SERVER se comunica como servidor Modbus TCP a través de la conexión PROFINET.
		La instrucción MB_RED_SERVER procesa órdenes de conexión de un cliente Modbus TCP, recibe y procesa órdenes Modbus y envía respuestas.

Tabla 13-75 Tipos de datos de los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REG_KEY [1]	IN	STRING[17]	Código de registro para la licencia La instrucción MB_RED_SERVER necesita una licencia en cada CPU.
DISCONNECT	IN	Bool	La instrucción MB_RED_SERVER se utiliza para entrar en una conexión pasiva con un módulo asociado. El servidor responde a una orden de conexión desde las direcciones IP que se indican como especificadas o no especificadas en las descripciones de las conexiones. Este parámetro puede utilizarse para controlar cuándo se acepta una orden de conexión: • 0: la CPU establece una conexión pasiva cuando no hay ninguna conexión de comunicación. • 1: inicialización de la finalización de la conexión. Si la entrada se activa, la CPU no procesa más órdenes de clientes y se inicia la finalización de la conexión. El valor "0003" se genera en el parámetro STATUS_x una vez finalizada correctamente la conexión.
MB_HOLD_REG [2]	IN_OUT	Variant	Puntero hacia el registro de retención Modbus de la instrucción MB_RED_SERVER MB_HOLD_REG siempre debe hacer referencia a un área de memoria que tenga más de dos bytes. El registro de retención contiene los valores a los que puede acceder un cliente Modbus por medio de las funciones Modbus 3 (lectura), 6 (escritura), 16 (escritura múltiple) y 23 (lectura y escritura en una tarea).
LICENSED [1]	OUT	Bool	• 0: la instrucción no tiene licencia. • 1: la instrucción tiene licencia.
IDENT_CODE [1]	OUT	STRING[18]	Identificador de la licencia. Utilice este string para solicitar el código de registro REG_KEY.
DR_NDR_0A	OUT	Bool	"Data Read" o "New Data Ready" a la conexión 0A: • 0: no hay datos nuevos • 1: el cliente Modbus ha leído o escrito datos nuevos
ERROR_0A	OUT	Bool	Si se produce un error durante una llamada de la instrucción MB_RED_SERVER a la conexión 0A, la salida del parámetro ERROR_0A se pone a "1". El parámetro STATUS_0A indica información detallada sobre la causa del problema.
STATUS_0A [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 0A.
DR_NDR_1A	OUT	Bool	"Data Read" o "New Data Ready" a la conexión 1A: • 0: no hay datos nuevos • 1: el cliente Modbus ha leído o escrito datos nuevos
ERROR_1A	OUT	Bool	Si se produce un error durante una llamada de la instrucción MB_RED_SERVER a la conexión 1A, la salida del parámetro ERROR_1A se pone a "1". El parámetro STATUS_1A indica información detallada sobre la causa del problema.
STATUS_1A [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 1A.
DR_NDR_0B	OUT	Bool	"Data Read" o "New Data Ready" a la conexión 0B: • 0: no hay datos nuevos • 1: el cliente Modbus ha leído o escrito datos nuevos

Parámetro y tipo		Tipo de datos	Descripción
ERROR_0B	OUT	Bool	Si se produce un error durante una llamada de la instrucción MB_RED_SERVER a la conexión 0B, la salida del parámetro ERROR_0B se pone a "1". El parámetro STATUS_0B indica información detallada sobre la causa del problema.
STATUS_0B [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 0B.
DR_NDR_1B	OUT	Bool	"Data Read" o "New Data Ready" a la conexión 1B: • 0: no hay datos nuevos • 1: el cliente Modbus ha leído o escrito datos nuevos
ERROR_1B	OUT	Bool	Si se produce un error durante una llamada de la instrucción MB_RED_SERVER a la conexión 1B, la salida del parámetro ERROR_1B se pone a "1". El parámetro STATUS_1B indica información detallada sobre la causa del problema.
STATUS_1B [3]	OUT	Word	Información de estado detallada de la instrucción en la conexión 1B.
RED_ERR_S7 [3]	OUT	Bool	• 0: no hay errores de redundancia en SIMATIC. • 1: error de redundancia en SIMATIC.
RED_ERR_S7 [3]	OUT	Bool	• 0: no hay errores de redundancia en el lado del interlocutor del enlace. • 1: error de redundancia en el lado del interlocutor del enlace.
RED_ERR_S7 [3]	OUT	Bool	• 0: se ha establecido por lo menos una conexión configurada. • 1: pérdida total de la comunicación, todas las conexiones configuradas han finalizado.

[1] Encontrará más información en el apartado "Obtener una licencia" más adelante.

[2] Encontrará más información en el apartado "Parámetro de entrada MB_HOLD_REG " más adelante.

[3] Encontrará más información en el apartado "Parámetros de salida: ERROR_x, RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR" más adelante.

Nota

Información de seguridad

Tenga en cuenta que cada cliente de la red recibe acceso de lectura y escritura a las entradas y salidas de la memoria imagen de proceso y al bloque de datos o área de marcas definidos en el registro de retención Modbus. La opción está disponible para restringir el acceso a una dirección IP y para prevenir operaciones de lectura y escritura no autorizadas. No obstante, tenga en cuenta que la dirección compartida también puede emplearse para un acceso no autorizado.

Nota

Requisitos de la versión del firmware de la CPU

Las instrucciones Modbus TCP descritas en este apartado del manual requieren una versión de firmware V4.2 o superior.

Para usar la instrucción no se requiere un módulo de hardware adicional.

Varias conexiones de servidor

Las CPU pueden:

- procesar varias conexiones de servidor
- aceptar varias conexiones de distintos clientes a la vez en un puerto de servidor

El número máximo de conexiones depende de la CPU utilizada y puede consultarse en los datos técnicos de la CPU. El número total de conexiones de una CPU, incluidas las de los clientes y el servidor Modbus TCP, no debe sobrepasar el número máximo de conexiones soportadas.

Para las conexiones de servidor deben recordarse las reglas siguientes:

- Cada conexión MB_RED_SERVER debe usar un DB de instancia exclusivo.
- Se requiere una conexión/un ID de conexión unívoco para cada cliente que quiera conectarse al puerto de servidor.
- Los ID de conexión deben ser unívocos en toda la CPU.

Mapeo de las direcciones Modbus en la memoria imagen de proceso

La instrucción MB_RED_SERVER permite que las funciones Modbus entrantes (1, 2, 4, 5 y 15) accedan directamente en modo de lectura y escritura a las entradas y salidas de la memoria imagen de proceso de la CPU (se utilizan los tipos de datos BOOL y WORD).

Para las CPU S7-1200, el espacio de direcciones para la memoria imagen de proceso de las entradas y las salidas es de 1 KB.

En la tabla siguiente se muestra el espacio de direcciones de las funciones Modbus indicadas anteriormente:

Función Modbus					
Código de función	**Función**	**Área de datos**	**Espacio de direcciones**		
01	Lectura: bits	Output	0	a	65.535
02	Lectura: bits	Input	0	a	65.535
04	Lectura: WORD	Input	0	a	65.535
05	Escritura: bit	Output	0	a	65.535
15	Escritura: bits	Output	0	a	65.535

Las órdenes Modbus entrantes con los códigos de función 3, 6, 16 y 23 escriben o leen los registros de retención Modbus (el registro de retención se especifica con el parámetro MB_HOLD_REG o mediante Data_Area_Array).

Funciones Modbus

La tabla siguiente lista todas las funciones Modbus que soporta la instrucción MB_RED_SERVER:

Código de función	**Descripción**
01	Leer bits de salida
02	Leer bits de entrada
03	Leer un registro de retención

Código de función	Descripción
04	Leer palabras de entrada
05	Escribir un bit de salida
06	Escribir un registro de retención
08	Función de diagnóstico: • Prueba de eco (subfunción 0x0000): la instrucción MB_RED_SERVER recibe una palabra de datos y la devuelve sin modificar al cliente Modbus. • Inicializar contador de eventos (subfunción 0x000A): la instrucción MB_RED_SERVER inicializa los siguientes contadores de eventos: "Success_Count", "Xmt_Rcv_Count", "Exception_Count", "Server_Message_Count" y "Request_Count".
11	Función de diagnóstico: recoger el contador de eventos de la comunicación La instrucción MB_RED_SERVER utiliza un contador de eventos de comunicación interno para registrar el número de órdenes de lectura y escritura ejecutadas correctamente que se han enviado al servidor Modbus. El contador de eventos no se incrementa con las funciones 8 u 11. Lo mismo es aplicable a órdenes que provocan un error de comunicación, por ejemplo, si se ha producido un error de protocolo; el código de función en la orden Modbus recibida no se soporta.
15	Escribir bits de salida
16	Escribir un registro de retención
23	Escribir un registro de retención y leer un registro de retención con una orden

Operación y redundancia

Los nodos de comunicación pueden diseñarse como autónomos o redundantes. Si uno de los interlocutores está diseñado como autónomo, nos referiremos a ello como redundancia unilateral. Si ambos interlocutores están diseñados como redundantes, lo designaremos como redundancia bilateral.

- Redundancia unilateral:
  - debe configurarse una conexión en cada caso para cada conexión entre los interlocutores. Los puntos de conexión del **SIMATIC S7** se denominan **0** y **1**; los puntos de conexión del **interlocutor de la comunicación** se denominan **A** y **B**.
    La CPU R o CPU H 1 recurre al punto de conexión 0, mientras que la CPU R o CPU H 2 recurre al punto de conexión 1.
  - Configuración: si el S7 está diseñado de forma redundante, se crea una conexión desde el punto de conexión 0 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A**) y otra conexión desde el punto de conexión 1 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **A** => conexión **1A**). La figura muestra las denominaciones de las conexiones:

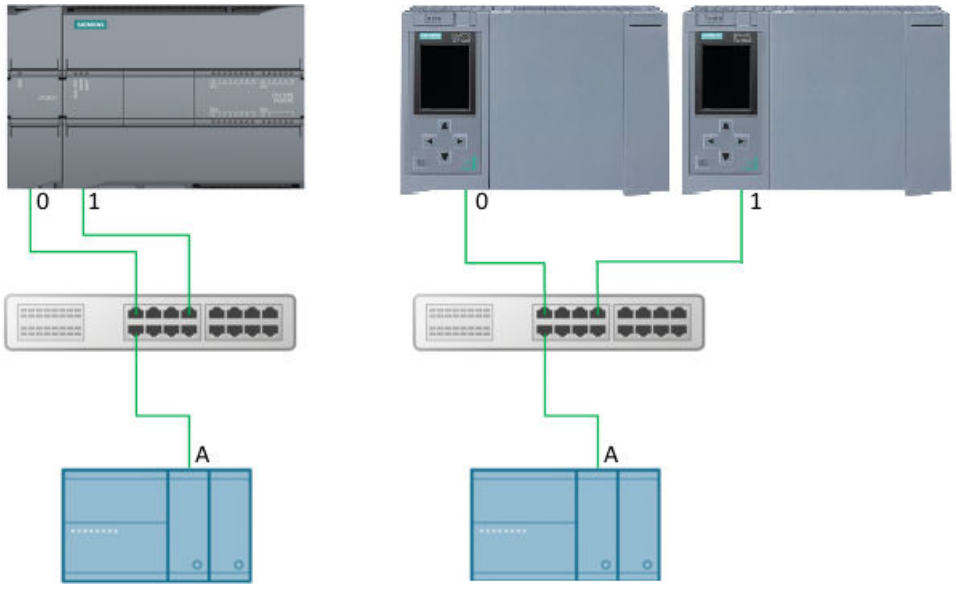


Figura 13-12 Redundancia unilateral S7

– Si el S7 está diseñado como autónomo y el interlocutor está diseñado de forma redundante, se crea una conexión desde el punto de conexión 0 del S7 al punto nodal A del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A**) y otra conexión desde el punto de conexión 0 del S7 al punto nodal B del interlocutor (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **B** => conexión **0B**). La figura muestra las denominaciones de las conexiones:

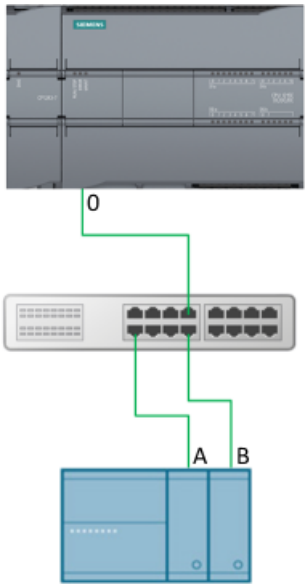


Figura 13-13 Interlocutor de redundancia unilateral

- Redundancia bilateral:
  – Descripción: debe configurarse una conexión en cada caso para cada conexión entre los interlocutores. Los puntos de conexión del SIMATIC S7 se denominan **0** y **1**; los puntos de conexión del **interlocutor de la comunicación** se denominan **A** y **B**.
  La CPU R o CPU H 1 recurre al punto de conexión 0, mientras que la CPU R o CPU H 2 recurre al punto de conexión 1.

– Configuración: en el caso de la redundancia bilateral, se crean dos conexiones desde el punto de conexión 0 (conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **A** => conexión **0A** y conexión desde el punto de conexión **0** del S7 al interlocutor/nodo **B** => conexión **0B**) y dos conexiones desde el punto de conexión 1 del S7 a los puntos nodales A y B del interlocutor (conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **A** => conexión **1A** y conexión desde el punto de conexión **1** del S7 al interlocutor/nodo **B** => conexión **1B**). La figura muestra las denominaciones de las conexiones:

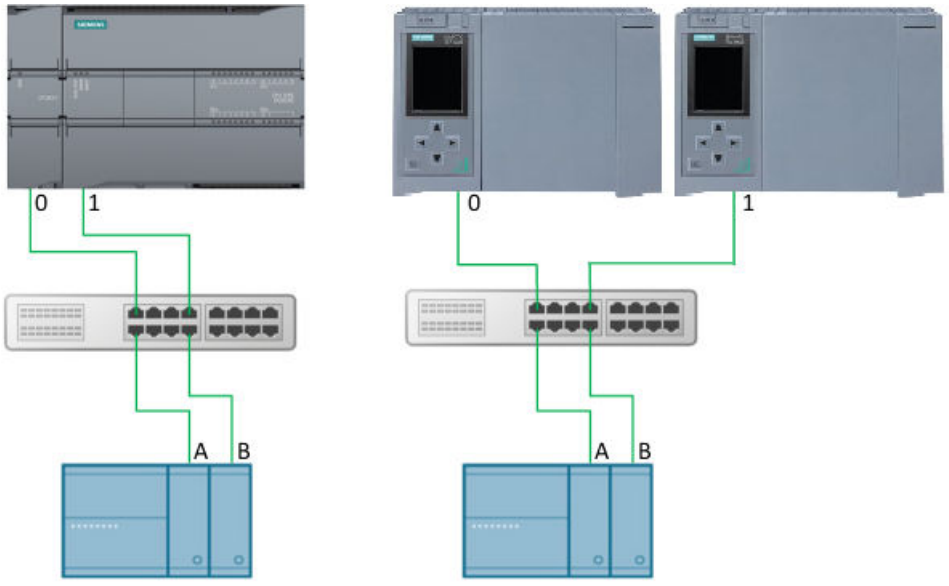


Figura 13-14 Redundancia bilateral

- Procesamiento de tramas: las tramas pueden recibirse por todas las conexiones configuradas. El cliente puede enviar tramas a través de una conexión o de todas ellas. Si en una conexión se ha recibido una trama, la CPU muestra el estado en la salida DR_NDR_x o ERROR_x correspondiente. Cada conexión se ejecuta independientemente y no influye en la visualización de las demás conexiones.
- Salidas de redundancia RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR:
  - Los bits de redundancia RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR se activan en función de los estados de las salidas de estado:

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

Figura 13-15 Visualización de los bits de alarma para la configuración de redundancia en ambos lados

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

Figura 13-16 Visualización de los bits de alarma para la configuración de redundancia en un lado

Nota

Números de puerto para cliente y servidor

El cliente Modbus utiliza un número de puerto que empieza por 2000. El servidor Modbus suele direccionarse a través del número de puerto 502. Dependiendo de la CPU, es posible configurar el puerto 502 para varias conexiones (puerto múltiple). Si el puerto 502 se ha configurado para dos o más conexiones, los clientes de los que provienen las órdenes se distribuyen aleatoriamente entre las conexiones de servidor existentes en caso de conexiones no especificadas. El primer cliente que quiere conectarse a la instrucción "MB_RED_SERVER" no se asigna automáticamente a la conexión 0A. Una vez se ha realizado la asignación de las órdenes de los clientes a las conexiones de servidor, la asignación permanece intacta durante el intercambio de tramas, hasta que finaliza la conexión.

Parametrización

Puede utilizarse la instrucción MB_RED_SERVER **V1.0** y **V1.1** para el S7-1200. La CPU implementa las conexiones a través de la interfaz local de la CPU o el CM/CP. La CPU configura y establece las conexiones utilizando la estructura TCON_IP_V4.

Configuración de MB_RED_SERVER: Los ajustes siguientes se realizan en el cuadro de diálogo de configuración de la instrucción MB_RED_SERVER:

- Parámetros de las conexiones 0A, 1A, 0B y 1B (encontrará más información sobre la configuración de la redundancia en el apartado anterior "Operación y redundancia").
- Parámetro interno (opcional)

El cuadro de diálogo de configuración se abre con la instrucción MB_RED_SERVER o mediante objetos tecnológicos:

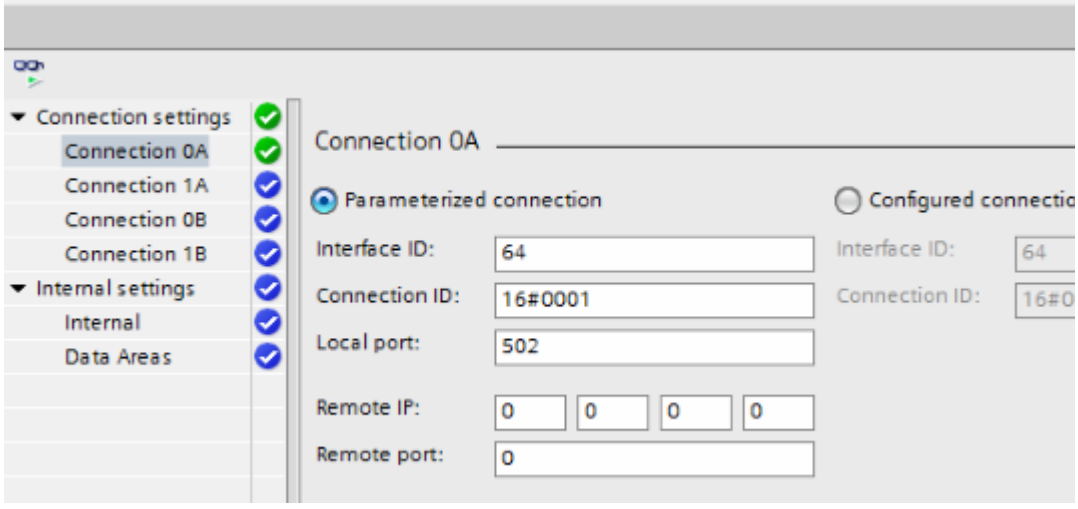


Figura 13-17 Conexión de servidor parametrizada

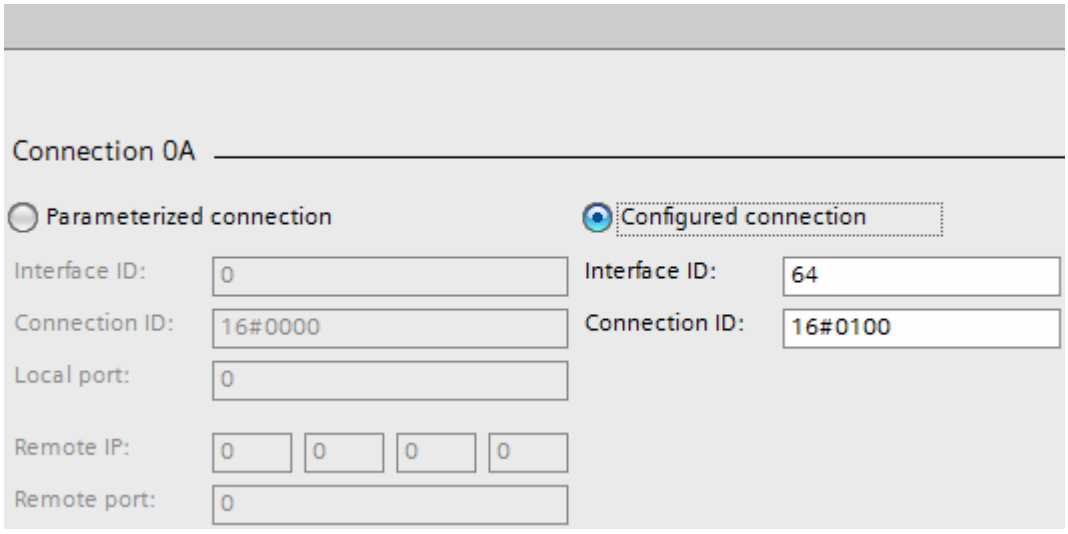


Figura 13-18 Conexión de servidor configurada

Variable	Valor inicial	Descripción
Conexiones configuradas		
Interface ID	64	Identificador de hardware de la interfaz PN utilizada
Connection ID	16#0000	ID de las conexiones utilizadas. Estos ID de conexión deben ser unívocos en toda la CPU.
Local port	502	Número de puerto local del bloque de servidor. El puerto predeterminado para el servidor Modbus/TCP es el 502.
Remote IP	0.0.0.0	Dirección IP remota del cliente. En el ajuste predeterminado no hay ninguna dirección IP introducida para el cliente.
Remote port	0	Número de puerto remoto del cliente. De forma predeterminada no se introduce ningún número de puerto para el cliente.
Conexiones configuradas		
Interface ID	64	Identificador de hardware de la interfaz PN utilizada
Connection ID	16#0000	ID de las conexiones utilizadas. Estas conexiones se configuran en la vista de redes.

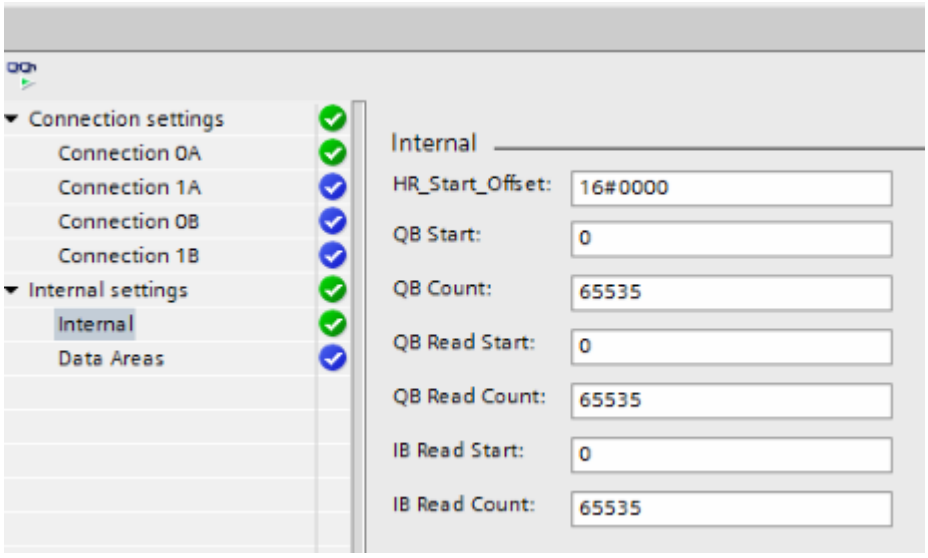


Figura 13-19 Parámetro interno (opcional)

Variable	Tipo de datos	Valor inicial	Descripción
HR_Start_Offset	WORD	0	Asignar la dirección inicial del registro de retención Modbus.
QB_Start	UINT	0	Dirección inicial del rango de direcciones permitido de las salidas en las que puede escribir el maestro Modbus (bytes 0 a 65535)
QB_Count	UINT	0	Número de bytes de salida en los que puede escribir el maestro Modbus. *Ejemplo:* • QB_Start=0 y QB_Count=10: el maestro Modbus puede escribir en los bytes de salida 0 a 9. • QB_Count=0: el maestro Modbus no puede escribir en ningún byte de salida.
QB_Read_Start	UINT	0	Dirección inicial del rango de direcciones permitido de las salidas que puede leer el maestro Modbus (bytes 0 a 65535)
QB_Read_Count	UINT	0	Número de bytes de salida que puede leer el maestro Modbus. *Ejemplo:* • QB_Read_Start=0 y QB_Read_Count=10: el maestro Modbus puede leer los bytes de salida 0 a 9. • QB_Read_Count=0: el maestro Modbus no puede leer ningún byte de salida.
IB_Read_Start	UINT	0	Dirección inicial del rango de direccionamiento permitido de las entradas que puede leer el maestro Modbus (bytes 0 a 65535)

Variable	Tipo de datos	Valor inicial	Descripción
IB_Read_Count	UINT	0	Número de bytes de entrada que puede leer el maestro Modbus. *Ejemplo:* IB_Read_Start=0 e IB_Read_Count=10: el maestro Modbus puede leer los bytes de entrada 0 a 9. IB_Read_Count=0: el maestro Modbus no puede leer ningún byte de entrada.
Data_Area_Array	ARRAY [1..8]		
data_type	UINT	0	Tipo de datos: de 0 a 4
db	UINT	0	Número del bloque de datos
start	UINT	0	Primera dirección Modbus en el bloque de datos
length	UINT	0	Número de valores Modbus en el bloque de datos

Direccionamiento mediante la variable estática HR_Start_Offset

Las direcciones del registro de retención Modbus empiezan en 0.

Ejemplo: El registro de retención empieza en MW100 y tiene una longitud de 100 palabras.

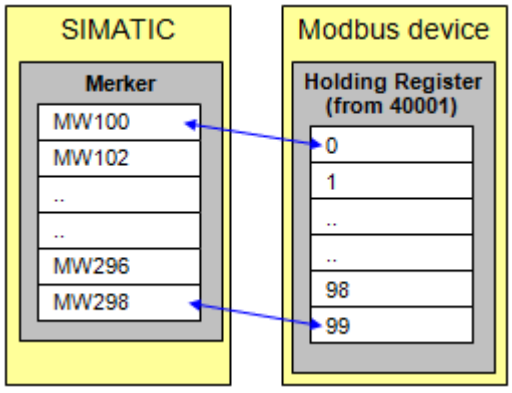


Figura 13-20 HR_Start_Offset_0

La variable HR_Start_Offset puede definirse de forma que el registro de retención Modbus tenga una dirección inicial diferente de 0.

Ejemplo: un valor de ajuste de 20 en el parámetro HR_Start_Offset significa que la dirección inicial del registro de retención se mueve de 0 a 20. Esto provoca un error siempre que se direccione el registro de retención por debajo de la dirección 20 y por encima de la dirección 119.

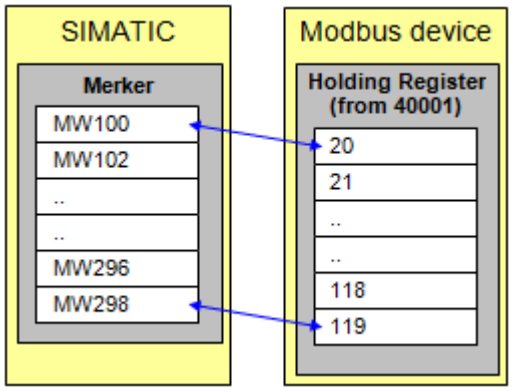


Figura 13-21 HR_Start_Offset_20

Data_Area_Array [1..8]

Son ocho áreas de datos disponibles para mapear las direcciones MODBUS en la memoria SIMATIC S7. Si el área de datos se define con el tipo de datos "registro de retención", el parámetro MB_HOLD_REG no se evalúa. En su lugar, el maestro Modbus escribe o lee el registro Modbus y los bits en los bloques de datos en función del tipo de tarea. La CPU puede seguir procesando estos valores en la ejecución posterior del programa.

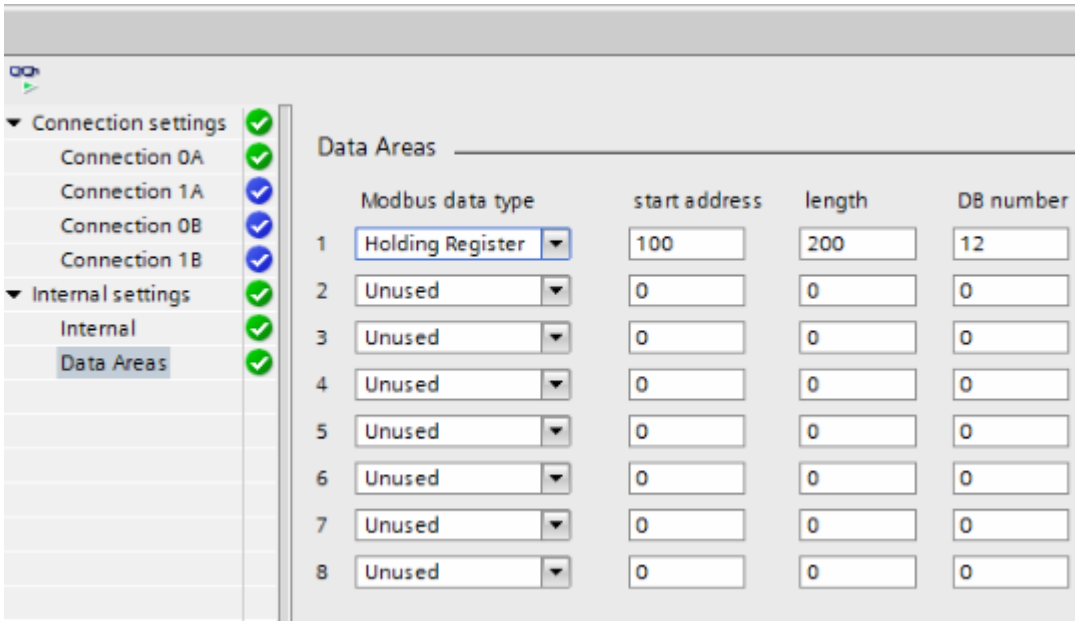


Figura 13-22 Áreas de datos del servidor

Solo es posible leer o escribir en un DB con cualquier tarea. El acceso a registros o valores de bit ubicados en varios DB, aunque los números sean consecutivos sin huecos, debe dividirse en dos tareas. Recuerde esto durante la configuración. Es posible mapear más áreas Modbus (registros o valores de bit) en un bloque de datos que las que puede procesar el maestro Modbus con una trama.

data_type

El parámetro data_type especifica qué tipos de datos MODBUS mapea el maestro Modbus en este bloque de datos. Si se introduce el valor "0" en data_type, el maestro Modbus no utiliza la

correspondiente área de datos. Si el maestro Modbus debe utilizar varias Data_Area, hay que definirlas una tras otra. El maestro Modbus no procesa entradas después de que data_type = 0.

Identi-ficador	Tipo de datos	Descripción
0	Área no utilizada	
1	Bits de salida (bobinas)	bit
2	Bits de entrada (entradas)	bit
3	Registro de retención	Word
4	Palabras de entrada (registro de entrada)	Word

db

El parámetro db especifica el bloque de datos que mapea los registros MODBUS o valores de bit definidos más adelante. La CPU no permite el número de DB 0 porque está reservado para el sistema.

start, length

start especifica la primera dirección Modbus que el maestro Modbus mapea en la palabra de datos 0 del DB. El parámetro length define la longitud de todas las direcciones MODBUS que el maestro Modbus mapea en el bloque de datos. Las áreas de datos definidas no deben solaparse. El parámetro length no debe ser 0.

Ejemplo: mapeo de direcciones con Data_Area_Array

Área de datos 1	data_type	3: registro de retención
	db	11
	start	0
	length	500
Área de datos 2	data_type	3: registro de retención
	db	12
	start	720
	length	181
Área de datos 3	data_type	4: palabras de entrada
	db	13
	start	720
	length	281
Área de datos 4	data_type	1: bits de salida
	db	14
	start	640
	length	611
Área de datos 5	data_type	2: bit de entrada
	db	15
	start	1700
	length	601

Área de datos 6	data_type	1: bits de salida
	db	16
	start	1700
	length	601
Área de datos 7	data_type	no se utiliza
	db	0
	start	0
	length	0
Área de datos 8	data_type	no se utiliza
	db	0
	start	0
	length	0

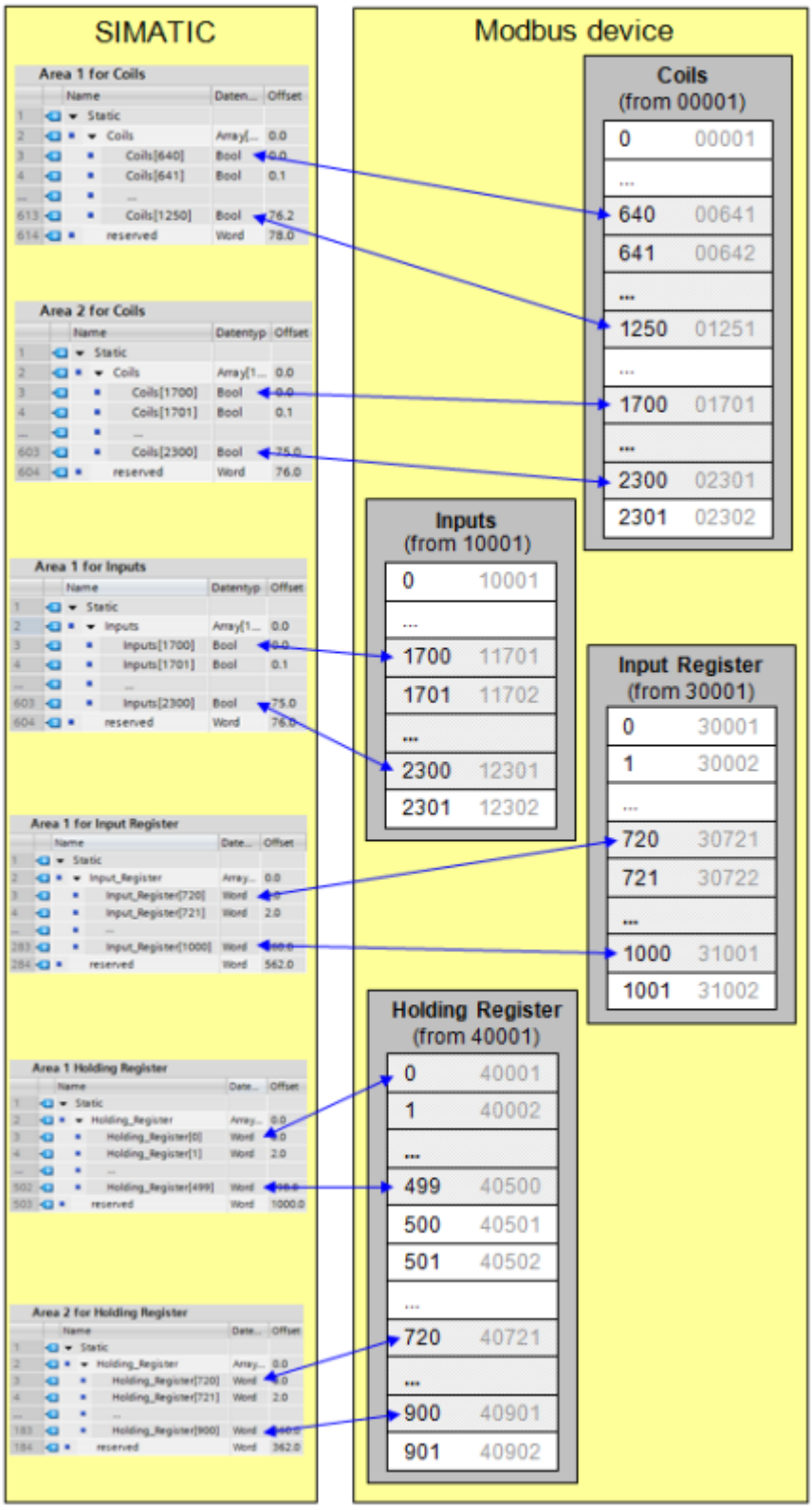


Figura 13-23 Generación de direcciones

Obtener una licencia

La instrucción MB_RED_SERVER está sujeta a una tasa y hay que obtener una licencia individualmente para cada CPU. Para obtener una licencia deben seguirse estos dos pasos:

- Visualizar la licencia IDENT_CODE
- Introducir la clave de registro REG_KEY: hay que asignar la clave de registro REG_KEY en cada instrucción MB_RED_SERVER. Guarde la REG_KEY en un bloque de datos global desde el que todas las instrucciones MB_RED_SERVER reciban la clave de registro que necesitan.

Procedimiento para visualizar la licencia IDENT_CODE:

1. Parametrice la instrucción MB_RED_SERVER de acuerdo con sus necesidades en un OB cíclico. Descargue el programa en la CPU y ponga la CPU en RUN.
2. Abra el DB de instancia de la instrucción Modbus y haga clic en el botón "Observar todos".
3. El DB de instancia muestra una cadena de caracteres de 18 dígitos en la salida IDENT_CODE.

	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

Figura 13-24 Licencia

4. Copie esta cadena utilizando la función de copiar y pegar desde el bloque de datos y péguela en el formulario (que se le envió por correo electrónico tras pedir el producto o que está incluido en el CD).
5. Envíe el formulario a Customer Support (https://support.industry.siemens.com/my/ww/en/requests/#createRequest) por medio de una solicitud de servicio. A continuación recibirá la clave de registro para su CPU.

Procedimiento para introducir la clave de registro REG_KEY:

1. Inserte un nuevo bloque de datos global con un nombre simbólico unívoco, por ejemplo "License_DB", mediante "Agregar nuevo bloque...".
2. Cree un parámetro REG_KEY en este bloque con el tipo de datos STRING[17].

LICENSE_DB

	Name	Data type	Offset	Start value
1	Static			
2	REG_KEY	String[17]	0.0	''

Figura 13-25 REG_KEY

3. Copie la clave de registro de 17 dígitos que se ha transmitido a la columna "Valor de arranque" con la función de copiar y pegar.
4. En el OB cíclico, introduzca el nombre del DB de licencia y el nombre de la cadena (por ejemplo, License_DB.REG_KEY) en el parámetro REG_KEY de la instrucción MB_RED_SERVER .
5. Descargue los bloques modificados en la CPU. Puede introducir la clave de registro en runtime; no es necesario pasar de STOP a RUN.
6. La comunicación Modbus/TCP por medio de la instrucción MB_RED_SERVER ya tiene licencia para esta CPU; el bit de salida LICENSED es TRUE.

Procedimiento para corregir la falta de licencia o una licencia incorrecta:

- Si se introduce una clave de licencia incorrecta o no se introduce ninguna clave, el ERROR LED de la CPU parpadea. Además, en el S7-1200 la CPU realiza una entrada cíclica en el búfer de diagnóstico en relación con la falta de licencia.

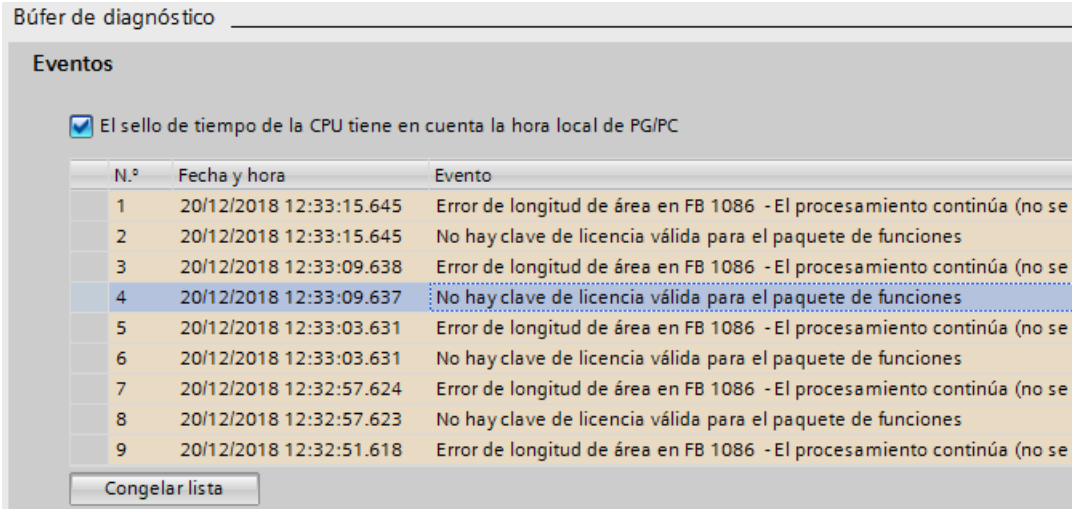


Figura 13-26 Diagnostic_buffer

- En caso de que falte la clave de registro o que esta sea incorrecta, la CPU procesa la comunicación Modbus TCP; sin embargo, la CPU siempre indica "W#16#0A90" (no hay clave de licencia válida para paquete funcional) en la salida STATUS_x. El bit de salida LICENSED es FALSE.

Parámetro de entrada MB_HOLD_REG

El parámetro MB_HOLD_REG es un puntero a un búfer de datos que almacena los datos a los que un cliente Modbus tiene acceso de lectura o escritura. Es posible utilizar un bloque de datos global (D) o una marca (M) como área de memoria:

- El límite superior para el número de direcciones en el bloque de datos (D) viene determinado por el tamaño máximo de un DB de su CPU.
- El límite superior para el número de marcas (M) viene determinado por el área de marcas máxima de su CPU.

Las figuras siguientes muestran algunos ejemplos del mapeo de direcciones Modbus al registro de retención para las funciones Modbus 3 (leer varios WORD), 6 (escribir un WORD), 16 (escribir varios WORD) y 23 (escribir y leer varios WORD).

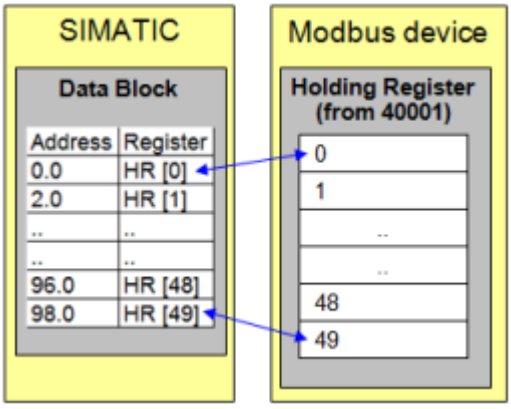


Figura 13-27 MB_HOLD_REG: bloque de datos con offset 0

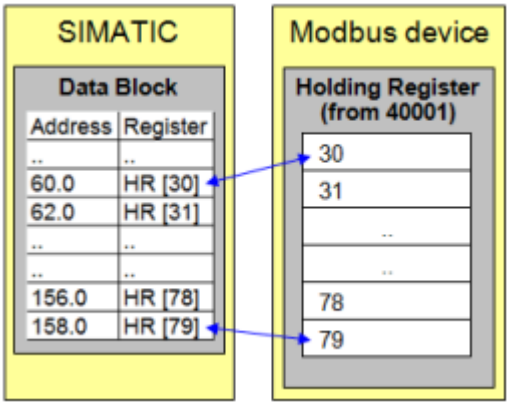


Figura 13-28 MB_HOLD_REG: bloque de datos con offset 60

Data_Area_Array [1..8]: para utilizar los parámetros opcionales Data_Area_Array [1..8], consulte el apartado "Parametrización" más arriba.

Parámetros de salida: ERROR_x, STATUS_x, RED_ERR_S7, RED_ERR_DEV y TOT_COM_ERR

La CPU muestra mensajes de error en las salidas de estado de la instrucción MB_RED_SERVER:

Nota

Los códigos de estado de los errores pueden visualizarse en el editor de programas como valor entero o hexadecimal:

1. Abra el bloque deseado en el editor de programas.
2. Active el estado de programación haciendo clic en "Activar/desactivar vigilancia". (Si todavía no ha establecido una conexión online, se abre el cuadro de diálogo "Establecer conexión online". En este cuadro de diálogo puede establecer una conexión online.)
3. Seleccione la variable que desee vigilar y seleccione el formato de visualización deseado en "Formato de visualización" del menú contextual.

Parámetro STATUS_x (información de estado general):

STATUS (W#16#)	Descripción
0000	Instrucción ejecutada sin errores.
0001	Conexión establecida.
0003	Conexión deshecha.
0A90	La instrucción MB_RED_SERVER no tiene licencia. Encontrará más información en el apartado "Obtener una licencia" más arriba.
0AFF	La conexión no está configurada y no se utiliza. La conexión 0A debe configurarse.
7000	No hay ninguna llamada activa y no se ha establecido ninguna conexión (REQ= 0, DISCONNECT= 1).
7001	Primera llamada. Se ha iniciado el establecimiento de la conexión.
7002	Llamada intermedia. Se está estableciendo la conexión.
7003	Se está deshaciendo la conexión.
7005	Se están transmitiendo datos.
7006	Se están recibiendo datos.

Parámetro STATUS_x (error de protocolo)

STATUS (W#16#)	Código de error en el mensaje de error de MB_RED_SERVER(B#16#)	Descripción
8380	-	La trama Modbus recibida tiene un formato incorrecto o no se han recibido suficientes bytes.
8381	01	El código de función no se soporta.
8382	03	Error en la longitud de datos: • Especificación de longitud no válida en la trama Modbus recibida • La longitud de trama introducida en el encabezado de la trama Modbus no concuerda con el número de bytes recibidos realmente. • El número de bytes introducido en el encabezado de la trama Modbus no concuerda con el número de bytes recibidos realmente (funciones 15 y 16).
8383	02	Error en la dirección de los datos o acceso fuera del área de direcciones del registro de retención (parámetro MB_HOLD_REG). Encontrará más información en el apartado "MB_HOLD_REG" más arriba.
8384	03	Error en el valor de los datos (función 05)
8385	03	El código de diagnóstico no se soporta (solo con la función 08).

Parámetro STATUS_x (error de parámetro)

STATUS (W#16#)	Descripción
80BB	El parámetro ActiveEstablished tiene un valor no válido Solo se permite un establecimiento de conexión pasivo para el servidor (active_established = FALSE).
8187	El parámetro MB_HOLD_REG tiene un puntero no válido. Área de datos demasiado pequeña.
8389	Definición no válida del área de datos: • El valor data_type no es válido • Número de DB no válido o no existe: – El valor db no es válido – El número de DB no existe – El número de DB ya se utiliza en otra área de datos – DB con acceso optimizado – El DB no está en la memoria de trabajo • El valor length no es válido • Solapamiento de rangos de direcciones MODBUS que pertenecen al mismo tipo de datos MODBUS

Nota

Códigos de error de instrucciones de comunicación usadas internamente

Con la instrucción MB_RED_SERVER, además de los errores listados en las tablas, pueden producirse errores provocados por las instrucciones de comunicación ("TCON", "TDISCON", "TSEND", "TRCV", "T_DIAG" y "T_RESET") que utiliza la instrucción.

Los códigos de error se asignan por medio del bloque de datos de instancia de la instrucción MB_RED_SERVER. Los códigos de error se muestran para la instrucción en cuestión en STATUS, en la sección "Static" de las diferentes instancias.

El significado de los códigos de error se explica en la documentación de la correspondiente instrucción de comunicación.

Nota

Error de comunicación al enviar o recibir datos

Si se produce un error de comunicación al enviar o recibir datos (80C4 (Error de comunicación temporal. La conexión especificada está interrumpida temporalmente.), 80C5 (El partner remoto ha deshecho activamente la conexión.), 80A1 (La conexión especificada está desconectada o todavía no se ha establecido.)), la CPU finaliza la conexión existente.

Esto significa también que todos los valores de STATUS devueltos pueden verse cuando finaliza la conexión y que el código de STATUS que ha provocado la finalización de la conexión solo se emite cuando esta ya ha finalizado.

Ejemplo: si se produce un error de comunicación cuando se reciben datos, primero se emite el STATUS 7003 (ERROR=false) y luego 80C4 (ERROR=true).

13.5.2.4 Ejemplos de Modbus TCP

Ejemplo: MB_SERVER conexiones TCP múltiples

Puede haber múltiples conexiones de servidor Modbus TCP. Para conseguirlo, se debe ejecutar MB_SERVER independientemente para cada conexión. Cada conexión debe usar un DB de instancia, una ID de conexión y un puerto IP independientes. El S7-1200 solo permite una conexión por puerto IP.

Para un rendimiento óptimo, se debe ejecutar MB_SERVER cada ciclo de programa para cada conexión.

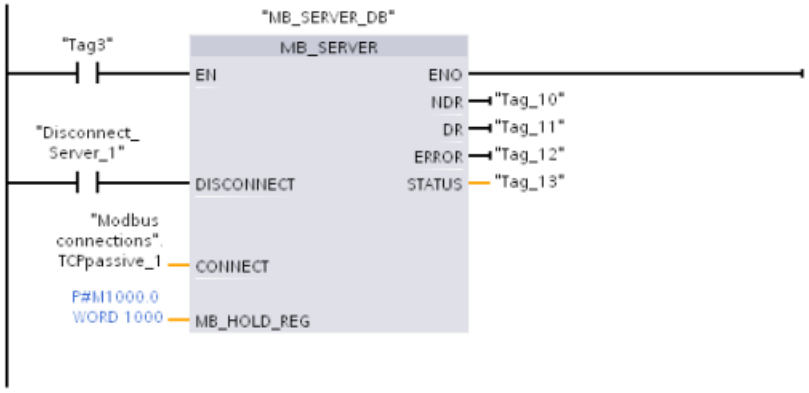
El parámetro CONNECT utiliza el tipo de datos de sistema TCON_IP_V4. En el caso del ejemplo, estas estructuras de datos se encuentran en un DB denominado "Conexiones Modbus". El DB "Conexiones Modbus" contiene dos estructuras TCON_IP_V4, "TCPpasiva_1" (para la conexión 1) y "TCP_pasiva_2" (para la conexión 2). La ID de propiedades de conexión y el puerto local descritos en los comentarios de la red son elementos de datos almacenados en la estructura de datos CONNECT.

Los datos CONNECT de TCON_IP_V4 también contienen una dirección IP en la matriz ADDR de RemoteAddress. Las asignaciones de direcciones IP en TCPpasiva_1 y TCP_pasiva_2 no afectan al establecimiento de conexiones de servidor TCP, pero determinan a qué clientes Modbus TCP se les permite comunicarse mediante las conexiones a cada MB_SERVER. MB_SERVER queda pasivamente a la espera de un mensaje del cliente Modbus y compara la dirección IP del mensaje entrante con la dirección IP almacenada en la matriz ADDR de RemoteAddress correspondiente.

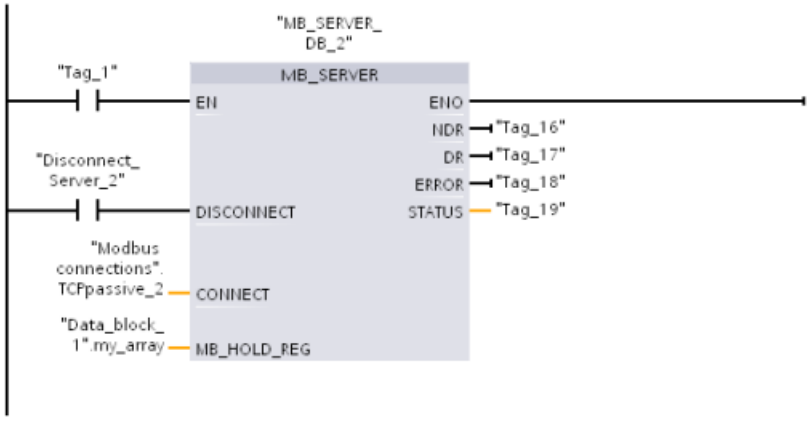
Hay tres variaciones de dirección IP de MB_SERVER posibles para las dos instrucciones MB_SERVER:

- **Dirección IP = 0.0.0.0**
  Cada MB_SERVER responderá a todos los clientes Modbus TCP que usen cualquier dirección IP.
- **Dirección IP = la misma dirección IP en TCPpasiva_1 y TCPpasiva_2**
  Ambas conexiones de MB_SERVER solo responden a clientes Modbus que tengan origen en esta dirección IP.
- **Dirección IP = número de IP diferente en TCP_pasiva_1 y TCP_pasiva_2**
  Cada MB_SERVER solo responde a clientes Modbus con origen en la dirección IP almacenada en sus datos de TCON_IP_V4.

Segmento 1: Conexión n.º 1, DB de instancia= "MB_SERVER_DB", en "Conexiones Modbus.TCPpasiva_1" (ID = 1 y LocalPort = 502)



Segmento 2: Conexión n.º 2, DB de instancia= "MB_SERVER_DB_1", en "Conexiones Modbus.TCPpasiva_2" (ID = 2 y LocalPort = 503)



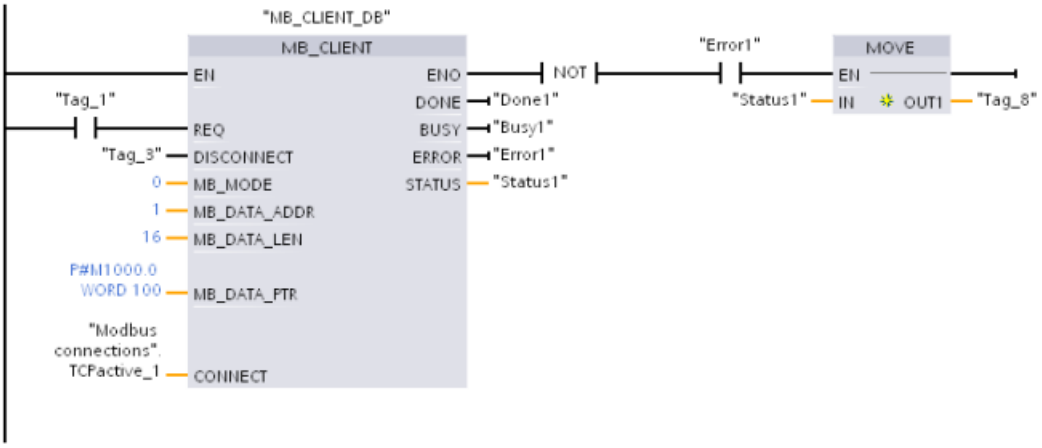
Ejemplo: MB_CLIENT 1: Peticiones múltiples con conexión TCP común

Se pueden enviar varias peticiones de cliente Modbus por la misma conexión. Para ello, se deben utilizar los mismos DB de instancia, ID de conexión y número de puerto.

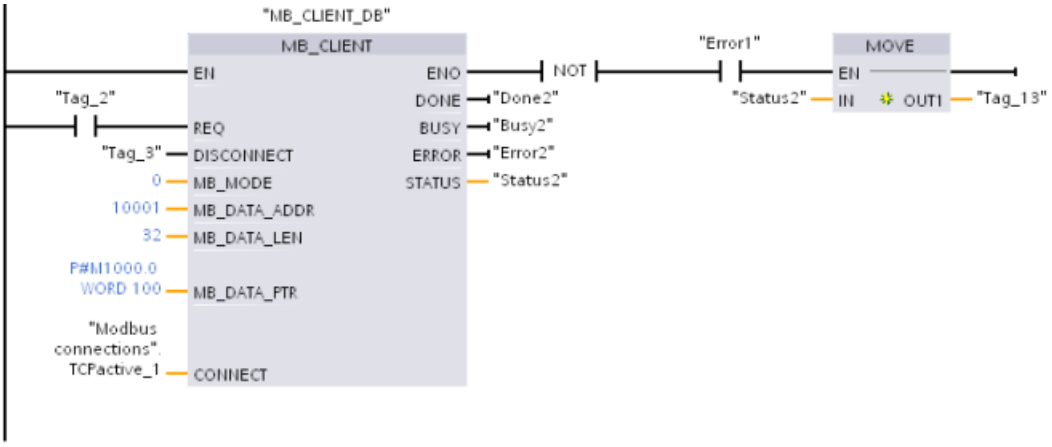
Puesto que ambos cuadros MB_CLIENT utilizan la misma estructura de datos TCON_IP_v4 del parámetro CONNECT ("Conexiones Modbus".TCPactiva_1), la ID de conexión, el número de puerto y la dirección IP son idénticos. Los datos de dirección IP de CONNECT asignan la dirección IP del servidor Modbus TCP de destino.

Solo puede estar activo un MB_CLIENT a la vez. Una vez que se termina la ejecución de un cliente, puede empezar la ejecución del siguiente. La lógica del programa es la responsable de la lógica de la secuencia de ejecución. En el ejemplo se muestra que ambos clientes leen los datos remotos de un único cliente Modbus y transfieren los datos a la CPU del cliente Modbus (la memoria M empieza en M1000.0). De forma opcional, se captura un error devuelto.

Segmento 1: Función Modbus 1: leer 16 bits de salida de un servidor Modbus TCP con la dirección IP asignada en "Conexiones Modbus".TCPactiva_1.



Segmento 2: Función Modbus 2: leer 32 bits de entrada de un servidor Modbus TCP con la dirección IP asignada en "Conexiones Modbus".TCPactiva_1.



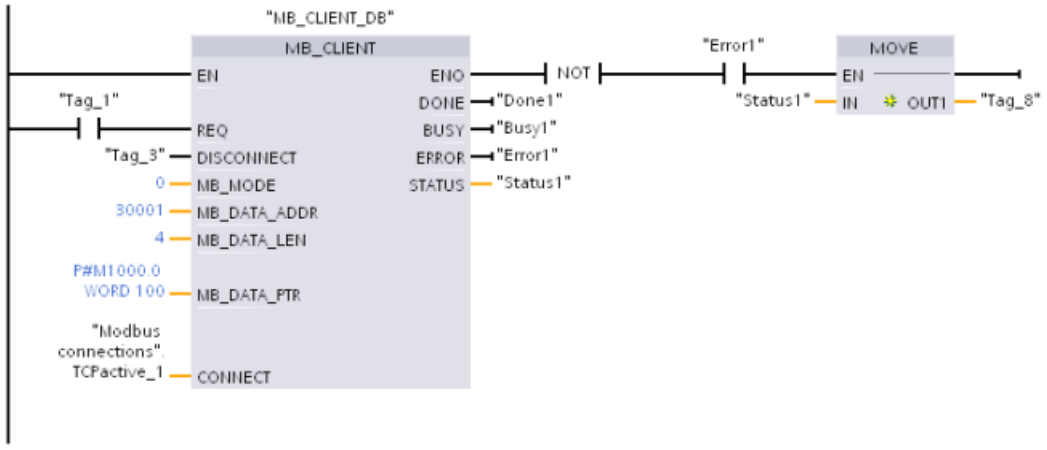
Ejemplo: MB_CLIENT 2: Peticiones múltiples con varias conexiones TCP

Las peticiones de cliente Modbus TCP se pueden enviar por varias conexiones. Para ello, se deben utilizar diferentes DBs de instancia e IDs de conexión.

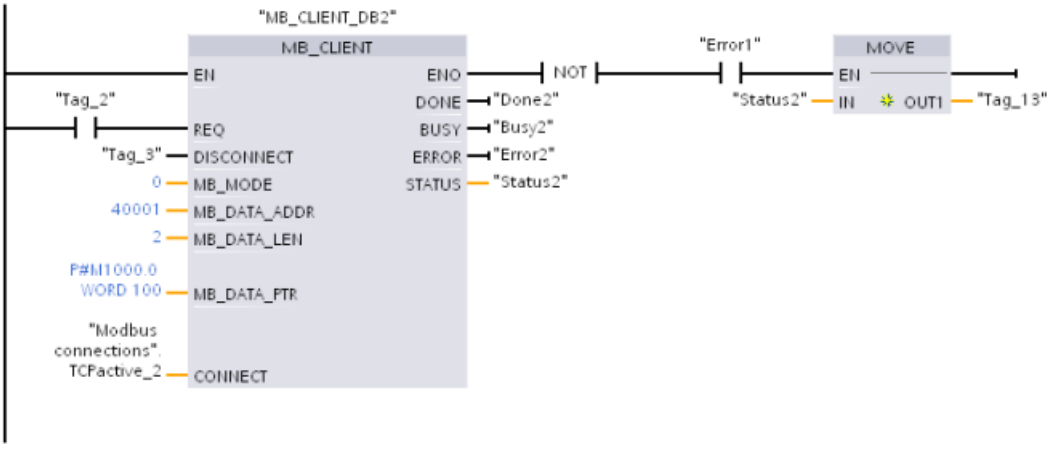
El número de RemotePort (puerto IP) debe ser diferente si las conexiones se establecen con el mismo servidor Modbus. Si las conexiones se realizan con servidores diferentes, no hay restricciones en número de puertos IP.

En el ejemplo se muestra cómo dos clientes Modbus TCP transfieren los datos remotos de dos servidores Modbus TCP diferentes a la misma área de memoria de CPU local, empezando en la dirección M1000.0. Además, de forma opcional, se captura un error devuelto.

Segmento 1: Función Modbus 4: leer palabras de la memoria imagen de proceso de las entradas de un servidor Modbus TCP
Parámetro CONNECT = "Conexiones Modbus".TCPactiva_1: **ID** de conexión = 1, **RemoteAddress** = 192.168.2.241, **RemotePort** = 502



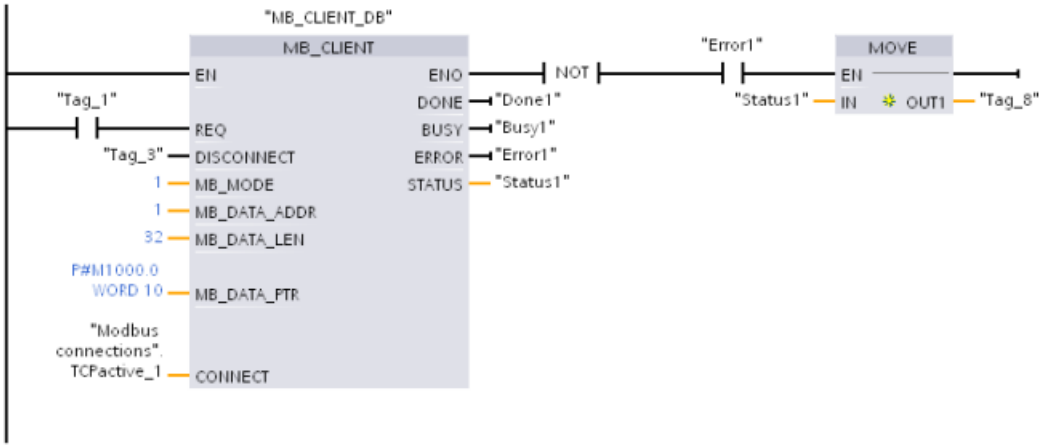
Segmento 2: Función Modbus 3: leer palabras de registro de retención de un servidor Modbus TCP
Parámetro CONNECT = "Conexiones Modbus".TCPactiva_2: **ID** de conexión = 2, **RemoteAddress** = 192.168.2.242, **RemotePort** = 502



Ejemplo: MB_CLIENT 3: Petición de escritura de imagen de salida

En este ejemplo se muestra una petición de cliente Modbus que transfiere datos de bits de la memoria de CPU local (empezando en M1000.0) a un servidor Modbus TCP remoto.

Segmento 1: Función Modbus 15: escribir bits de salida en un servidor Modbus

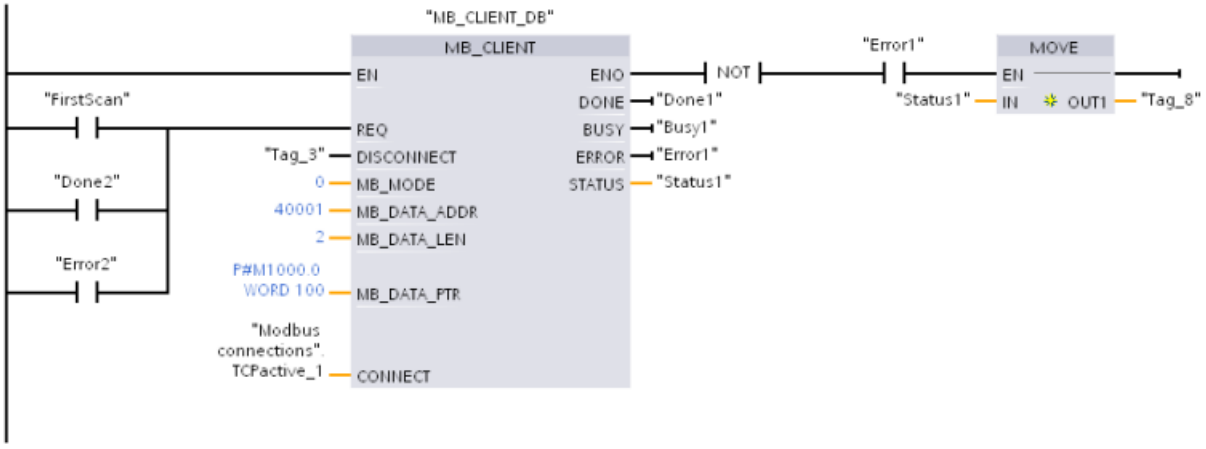


Ejemplo: MB_CLIENT 4: Coordinación de varias peticiones

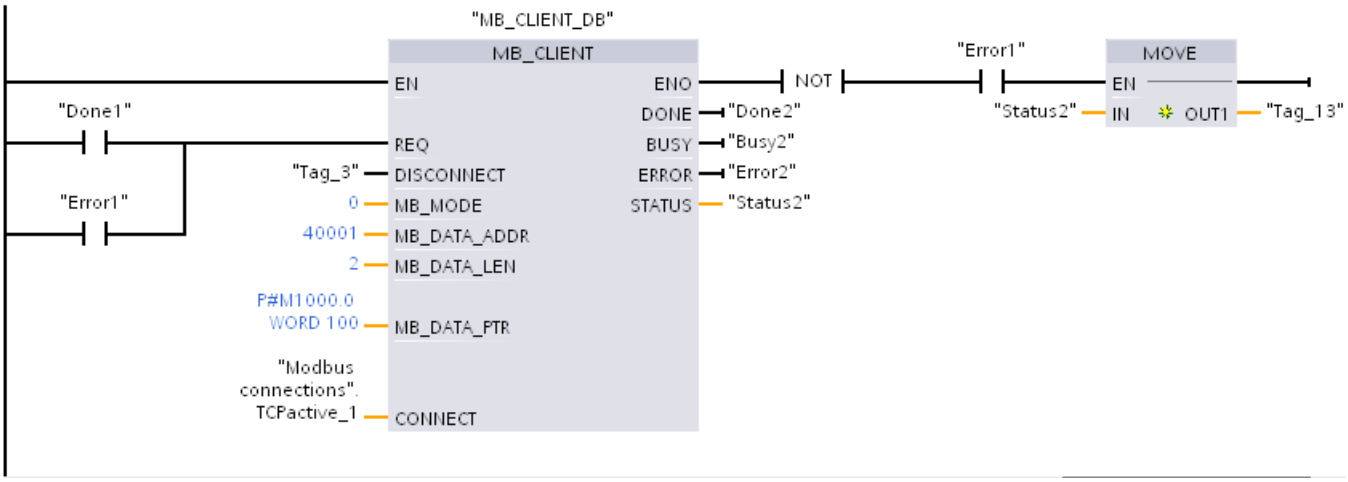
Se debe asegurar de que cada petición Modbus TCP acaba su ejecución. La secuencia de ejecución debe controlarla su lógica de programa. En el ejemplo siguiente se muestra cómo las salidas de las peticiones de cliente primera y segunda pueden controlar la secuencia de ejecución.

En el ejemplo se muestra cómo ambos clientes utilizan los mismos datos de conexión CONNECT (usado en diferentes momentos). Los clientes transfieren los datos de registro de retención del mismo servidor Modbus TCP remoto a la misma dirección de memoria M de la CPU local. Además, de forma opcional, se captura un error devuelto.

Segmento 1: Función Modbus 3: leer palabras de registro de retención del servidor Modbus TCP



Segmento 2: Función Modbus 3: leer palabras de registro de retención del servidor Modbus TCP



13.5.3 Modbus RTU

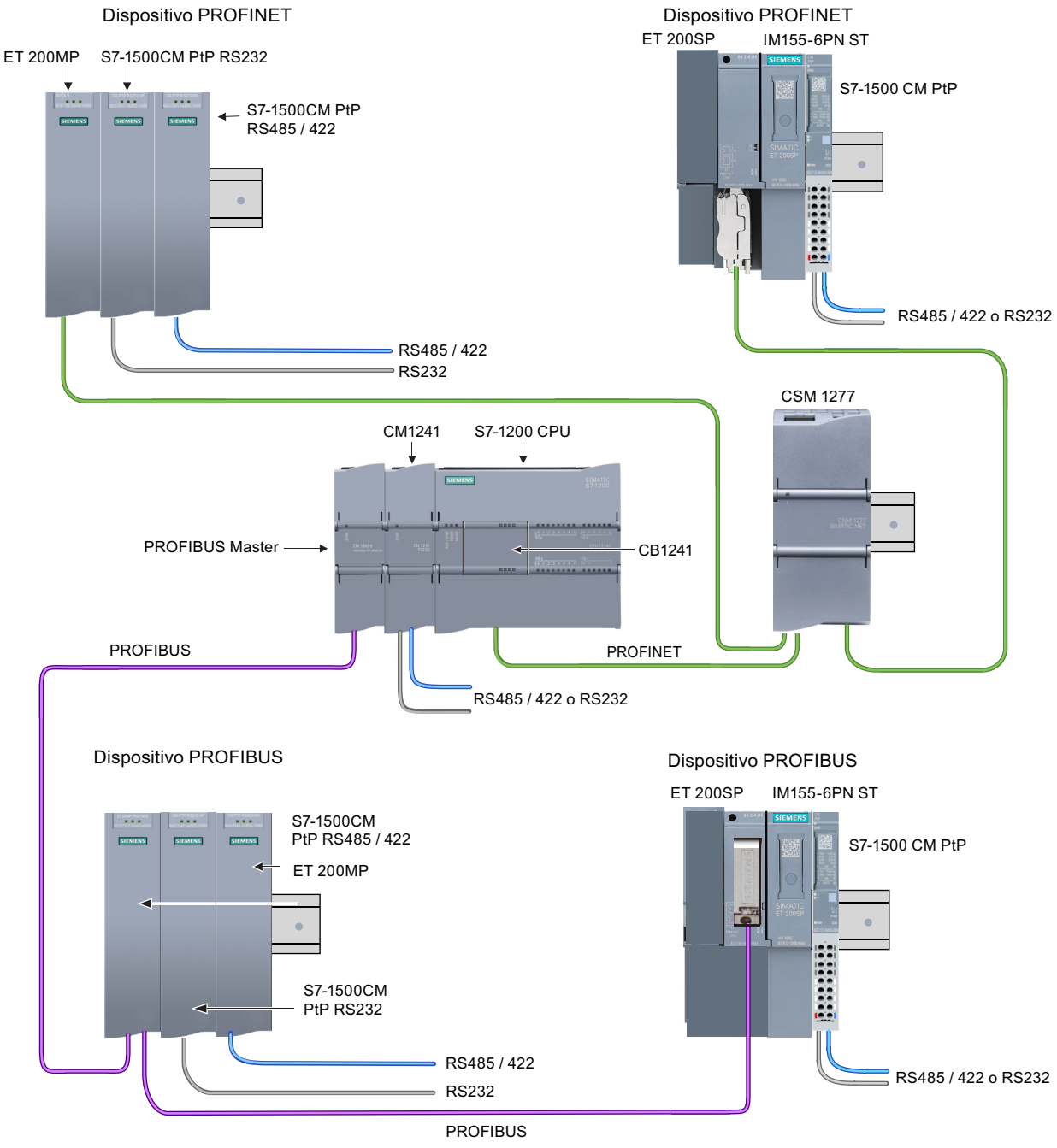
13.5.3.1 Vista general

A partir de la versión V4.1 de la CPU S7-1200 junto con STEP 7 V13 SP1, la CPU amplía la capacidad de Modbus RTU para usar un rack de periferia descentralizada PROFINET o PROFIBUS para comunicarse con varios dispositivos (lectores RFID, dispositivos GPS y otros):

- PROFINET (Página 594): la interfaz Ethernet de la CPU S7-1200 se conecta a un módulo de interfaz PROFINET. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.
- PROFIBUS (Página 777): un módulo de comunicación PROFIBUS se inserta en el lado izquierdo del rack con la CPU S7-1200. El módulo de comunicación PROFIBUS se conecta al rack que contiene un módulo de interfaz PROFIBUS. Los módulos de comunicación PtP en el rack con el módulo de interfaz pueden ofrecer comunicaciones serie a los dispositivos PtP.

Por este motivo, S7-1200 soporta dos juegos de instrucciones PtP:

- Instrucciones Legacy Modbus RTU (Página 1159): estas instrucciones Modbus RTU son anteriores a la versión V4.0 de S7-1200 y solo funcionan con la comunicación serie si se utiliza un módulo de comunicación CM 1241 o una Communication Board CB 1241.
- Instrucciones Modbus RTU (Página 1087): Estas instrucciones Modbus RTU ofrecen toda la funcionalidad de las instrucciones Legacy más la capacidad de conectarse a la periferia descentralizada de PROFINET y PROFIBUS. Las instrucciones Modbus RTU permiten configurar la comunicación entre los módulos de comunicación PtP en el rack de periferia descentralizada y los dispositivos PtP. Los módulos CM 1241 del S7-1200 deben tener como mínimo la versión de firmware V2.1 para poder utilizar estas instrucciones Modbus RTU.



Nota

Con la versión V4.1 de S7-1200 se pueden emplear las instrucciones punto a punto para todos los tipos de comunicación punto a punto: serie, serie mediante PROFINET y serie mediante PROFIBUS. STEP 7 ofrece las instrucciones Legacy punto a punto solo por su compatibilidad con programas existentes. Sin embargo, las instrucciones Legacy siguen funcionando con CPUs V4.1 así como con CPUs V4.0 o anteriores. No es necesario convertir programas anteriores de un juego de instrucciones a otro.

13.5.3.2 Seleccionar la versión de las instrucciones Modbus RTU

En STEP 7, hay disponibles las siguientes versiones de las instrucciones Modbus RTU:

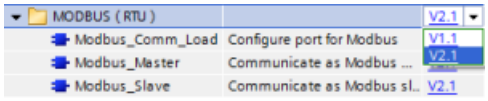
- Versión 3.4: compatible con CPU V4.0 y superiores y con CM V2.1 y superiores
- Versión 4.3 o superiores: compatible con CPU V4.2 y superiores y con CM V2.1 y superiores

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario. Además, la CPU S7-1200 sigue soportando las versiones anteriores de las instrucciones Modbus RTU (Página 1159).

No es posible utilizar las dos versiones de las instrucciones con el mismo módulo; en cambio, dos módulos distintos sí pueden utilizar diferentes versiones de las instrucciones.

En la Task Card Instrucciones, visualice las instrucciones MODBUS (RTU) en el grupo "Procesador de comunicaciones".

Para cambiar la versión de las instrucciones Modbus RTU, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.



Cuando se utiliza el árbol de instrucciones para insertar una instrucción Modbus RTU en el programa, se crea una nueva instancia del FB en el árbol del proyecto. El nuevo FB instancia se puede ver en el árbol de proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción Modbus RTU en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione una instancia del FB Modbus RTU del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción Modbus RTU.

13.5.3.3 Número máximo de esclavos Modbus soportados

El direccionamiento Modbus soporta un máximo de 247 esclavos (números de esclavo 1 a 247). Cada segmento de red Modbus puede tener un máximo de 32 dispositivos, en función de las capacidades de carga y accionamiento de la interfaz RS485. Cuando se llega al límite de 32 dispositivos hay que utilizar un repetidor para expandirse al próximo segmento. Se requieren siete repetidores para soportar los 247 esclavos conectados a un maestro de RS485.

Los repetidores de Siemens solo funcionan con PROFIBUS; su función consiste en supervisar el pase del testigo en PROFIBUS. No es posible utilizar repetidores de Siemens con otros protocolos. Así pues, para Modbus se necesitan repetidores de terceros.

Los tiempos de vigilancia o timeouts predeterminados de Modbus son largos; el uso de varios repetidores no genera problemas de retardo. El maestro Modbus no vigila si un esclavo es lento a la hora de responder o si varios repetidores retardan la respuesta.

13.5.3.4 Instrucciones Modbus RTU

Instrucción Modbus_Comm_Load (Configurar SIPLUS I/O o un puerto en el módulo PtP para Modbus RTU)

Tabla 13-76 Instrucción Modbus_Comm_Load

KOP / FUP	SCL	Descripción

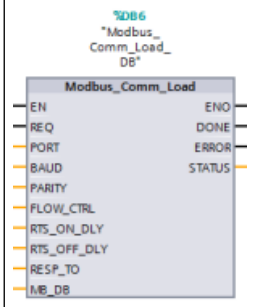
	`"Modbus_Comm_Load_DB"(`
	`REQ:=_bool_in,`
	`PORT:=_uint_in_,`
	`BAUD:=_udint_in_,`
	`PARITY:=_uint_in_,`
	`FLOW_CTRL:=_uint_in_,`
	`RTS_ON_DLY:=_uint_in_,`
	`RTS_OFF_DLY:=_uint_in_,`
	`RESP_TO:=_uint_in_,`
	`DONE=>_bool_out,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DB:=_fbtref_inout_);`
		La instrucción Modbus_Comm_Load configura SIPLUS I/O o un puerto PtP para la comunicación con el protocolo Modbus RTU.
		Opciones de hardware del puerto Modbus RTU: Instale un máximo de tres CMs (RS485 o RS232) más una CB (R4845).
		Opciones SIPLUS I/O para Modbus RTU: instale ET 200MP S7-1500CM PtP (RS485 / 422 o RS232) o ET 200SP S7-1500 CM PtP (RS485 / 422 o RS232)
		Un bloque de datos de instancia se asigna automáticamente cuando la instrucción Modbus_Comm_Load se inserta en el programa.

Tabla 13-77 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
EN	IN	Bool	Nota: La instrucción Modbus_Comm_Load de Modbus RTU utiliza las instrucciones RDREC y WRREC para inicializar el módulo PTP. Sin embargo, la instrucción RDREC/WRREC funciona de forma asíncrona, lo que significa que tarda varios ciclos en finalizar. Por consiguiente, el parámetro EN de la instrucción Modbus_Comm_Load debe mantenerse en TRUE hasta que finalice la instrucción RDREC/WRREC.
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Solo versión 2.0)
PORT	IN	Port	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PORT. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
BAUD	IN	UDInt	Selección de la velocidad de transferencia: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, no se permite ningún otro valor
PARITY	IN	UInt	Selección de paridad: • 0 – Ninguna • 1 – Impar • 2 – Par

Parámetro y tipo		Tipo de datos	Descripción
FLOW_CTRL [1]	IN	UInt	Selección del control de flujo: • 0 – (ajuste predeterminado) sin control de flujo • 1 – Control de flujo por hardware con RTS siempre ON (no es aplicable a los puertos RS485) • 2 – Control de flujo por hardware con RTS conmutado
RTS_ON_DLY [1]	IN	UInt	Selección de retardo RTS ON: • 0 – (ajuste predeterminado) Sin retardo desde RTS ON hasta que se transmite el primer carácter del mensaje • 1 a 65535 – Retardo en milisegundos desde RTS ON hasta que se transmite el primer carácter del mensaje (no es aplicable a los puertos RS485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RTS_OFF_DLY [1]	IN	UInt	Selección de retardo RTS OFF: • 0 – (ajuste predeterminado) Sin retardo desde el último carácter transmitido hasta que se desactiva RTS • 1 a 65535 – Retardo en milisegundos desde el último carácter transmitido hasta que se desactiva RTS (no es aplicable a los puertos RS485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RESP_TO [1]	IN	UInt	Plazo de respuesta: Tiempo en milisegundos permitido por el Modbus_Master para la respuesta del esclavo. Si el esclavo no responde en este tiempo, Modbus_Master repetirá la solicitud o la finalizará con un error cuando se haya enviado el número de reintentos indicado. 5 ms a 65535 ms (valor predeterminado = 1000 ms).
MB_DB	IN	Variant	Referencia al bloque de datos de instancia que utilizan las instrucciones Modbus_Master o Modbus_Slave. Una vez insertadas las instrucciones Modbus_Master o Modbus_Slave en el programa, el identificador de DB aparece en la lista desplegable de parámetros disponible en la conexión del cuadro MB_DB.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última orden sin errores. (Solo versión 2.0)
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última orden con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

[1] Parámetros opcionales para Modbus_Comm_Load (V 2.x o superior). Haga clic en la flecha situada en la parte inferior de una casilla KOP/FUP para expandir la casilla e incluir estos parámetros.

Modbus_Comm_Load se ejecuta para configurar un puerto para el protocolo Modbus RTU. Al configurar un puerto para el protocolo Modbus RTU, el mismo solo puede ser utilizado por la instrucción Modbus_Master o Modbus_Slave.

Es preciso utilizar una ejecución de Modbus_Comm_Load para configurar cada uno de los puertos de comunicación utilizado para la comunicación Modbus. Asígnele a cada puerto utilizado un DB de instancia Modbus_Comm_Load unívoco. En la CPU se pueden instalar hasta tres módulos de comunicación (RS232 o RS485) y una placa de comunicación (RS485). Llame Modbus_Comm_Load desde un OB de arranque y ejecute esta instrucción una vez, o bien utilice la marca de sistema del primer ciclo (Página 93) para iniciar la llamada y ejecutarla una vez.

Ejecute nuevamente Modbus_Comm_Load solo si es necesario modificar los parámetros de comunicación, p. ej. la velocidad de transferencia o la paridad.

Si se utiliza la librería Modbus con un módulo en un rack descentralizado, la instrucción Modbus_Comm_Load debe ejecutarse en una rutina de alarma cíclica (por ejemplo, una vez por segundo o una vez cada 10 segundos). Si falla la alimentación eléctrica del rack descentralizado o el módulo se extrae, hasta que se restablece el funcionamiento del módulo solo se envía el juego de parámetros HWConfig al módulo PtP. Todas las órdenes iniciadas por el Modbus_Master exceden el tiempo y Modbus_Slave pasa a silencio (no responde a ningún mensaje). La ejecución cíclica de la instrucción Modbus_Comm_Load soluciona estos problemas.

Un bloque de datos de instancia se asigna para Modbus_Master o Modbus_Slave cuando estas instrucciones se insertan en el programa. Este bloque de datos de instancia se referencia cuando se especifica el parámetro MB_DB para la instrucción Modbus_Comm_Load.

Variables del bloque de datos (DB) de instancia Modbus_Comm_Load

La tabla siguiente muestra las variables estáticas públicas almacenadas en el DB de instancia Modbus_Comm_Load que se pueden usar en el programa:

Tabla 13-78 Variables estáticas en el DB de instancia Modbus_Comm_Load

Variable	Tipo de datos	Valor predeterminado	Descripción
ICHAR_GAP	Word	0	Tiempo de retardo de caracteres máximo entre caracteres. Este parámetro está especificado en milisegundos y aumenta el período anticipado entre los caracteres recibidos. El número correspondiente de tiempos de bit para el parámetro se añade al valor predeterminado Modbus de 35 tiempos de bit (3,5 tiempos de carácter).
RETRIES	Word	2	Número de reintentos del maestro antes de devolver el código de error "0x80C8" para "sin respuesta".
EN_SUPPLY_VOLT	Bool	0	Habilitar el diagnóstico de falta de tensión de alimentación L+.
MODE	USInt	0	Estado operativo Los estados operativos válidos son los siguientes: • 0 = dúplex (RS232) • 1 = dúplex (RS422) a cuatro hilos (punto a punto) • 2 = dúplex (RS422) a cuatro hilos (maestro multipunto, CM PtP (ET 200SP)) • 3 = dúplex (RS422) a cuatro hilos (esclavo multipunto, CM PtP (ET 200SP)) • 4 = semidúplex (RS485) a dos hilos (consulte la nota inferior).

Variable	Tipo de datos	Valor predeterminado	Descripción
LINE_PRE	USInt	0	Estado inicial de la línea de recepción Los estados iniciales válidos son los siguientes: • 0 = "ningún" estado inicial (consulte la nota inferior) • 1 = señal R(A) = 5 V DC, señal R(B) = 0 V DC (detección de rotura de hilo): es posible detectar roturas de hilo con este estado inicial. Solo puede seleccionarse con: "dúplex (RS422) a cuatro hilos (conexión punto a punto)" y "dúplex (RS422) a cuatro hilos (esclavo multipunto)". • 2 = señal R(A) = 0 V DC, señal R(B) = 5 V DC: este ajuste predeterminado equivale al estado de reposo (no hay operaciones de transmisión activas). No es posible detectar roturas de hilo con este estado inicial.
BRK_DET	USInt	0	Detección de rotura de hilo Las selecciones siguientes son válidas: • 0 = detección desactivada • 1 = detección activada
EN_DIAG_ALARM	Bool	0	Activar alarma de diagnóstico: • 0 = no activada • 1 = activada
STOP_BITS	USInt	1	Número de bits de parada: • 1 = 1 bit de parada • 2 = 2 bits de parada • 0, 3 a 255 = reservado

Nota

Ajuste necesario para utilizar cables PROFIBUS con CM 1241 para RS485

Tabla 13-79 Códigos de condición de ejecución de Modbus_Comm_Load [1]

STATUS (W#16#)	Descripción
0000	No hay error
8180	ID de puerto no válido (identificador de puerto/hardware incorrecto para el módulo de comunicación)
8181	Velocidad de transferencia no válida
8182	Paridad no válida
8183	Valor de control de flujo no válido

STATUS (W#16#)	Descripción
8184	Valor de timeout de respuesta no válido (timeout de respuesta por debajo del valor mínimo de 5 ms)
8185	El parámetro MB_DB no es un bloque de datos de instancia de una instrucción Modbus_Master o Modbus_Slave.

[1] Además de los errores Modbus_Comm_Load indicados arriba, las instrucciones de comunicación PtP subyacentes pueden devolver errores.

Instrucción Modbus_Master (Comunicar vía SIPLUS I/O o el puerto PtP como maestro Modbus RTU)

Tabla 13-80 Instrucción Modbus_Master

KOP / FUP	SCL	Descripción

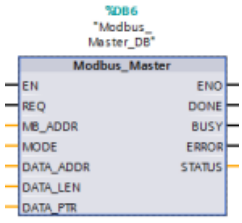
	`"Modbus_Master_DB"(`
	`REQ:=_bool_in_,`
	`MB_ADDR:=_uint_in_,`
	`MODE:=_usint_in_,`
	`DATA_ADDR:=_udint_in_,`
	`DATA_LEN:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DATA_PTR:=_variant_inout_);`
		La instrucción Modbus_Master se comunica como un maestro Modbus utilizando un puerto configurado por una ejecución anterior de la instrucción Modbus_Comm_Load. Un bloque de datos de instancia se asigna automáticamente cuando la instrucción Modbus_Master se inserta en el programa. El bloque de datos de instancia Modbus_Master se utiliza cuando se especifica el parámetro MB_DB para la instrucción Modbus_Comm_Load.

Tabla 13-81 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	0 = No hay ninguna petición 1 = Petición de transmitir datos a esclavo Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Dirección de estación Modbus RTU: Rango de direccionamiento estándar (de 1 a 247) Rango de direccionamiento avanzado (de 1 a 65535) El valor 0 está reservado para enviar un mensaje Broadcast a todos los esclavos Modbus. Los códigos de función Modbus 05, 06, 15 y 16 son los únicos que se soportan para el Broadcast.
MODE	IN	USInt	Selección de modo: Especifica el tipo de petición (lectura, escritura o diagnóstico). Consulte los detalles en la tabla de funciones Modbus que aparece más abajo.
DATA_ADDR	IN	UDInt	Dirección inicial en el esclavo: Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus. Las direcciones válidas se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_LEN	IN	UInt	Longitud de datos: Determina el número de bits o palabras a las que debe accederse en esta petición. Las longitudes válidas se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_PTR	IN_OUT	Variant	Puntero a los datos: Apunta a la dirección del M o del DB (tipo de DB no optimizado) de los datos que se están escribiendo o leyendo.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.

Parámetro y tipo		Tipo de datos	Descripción
BUSY	OUT	Bool	• 0 - Ninguna operación Modbus_Master en curso • 1- Operación Modbus_Master en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

Reglas de comunicación Modbus_Master

- Modbus_Comm_Load debe ejecutarse para configurar un puerto antes de que la instrucción Modbus_Master pueda comunicarse con ese puerto.
- Si un puerto debe utilizarse para iniciar peticiones del maestro Modbus, Modbus_Slave no debe utilizar este puerto. Una o más instancias de la ejecución de Modbus_Master pueden utilizarse en ese puerto, no obstante todas las ejecuciones de Modbus_Master deben utilizar el mismo DB de instancia Modbus_Master para ese puerto.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe consultar la instrucción Modbus_Master para transmitir y recibir condiciones completas.
- Llame todas las ejecuciones de Modbus_Master para un puerto determinado desde un OB de ciclo de programa. Las instrucciones Modbus_Master solo se pueden ejecutar en un nivel de ejecución, ya sea en el de retardo/cíclico o bien en el de ciclo del programa. Estas no se deben ejecutar en ambos niveles de prioridad. Si una instrucción de Modbus_Master de un nivel de prioridad de ejecución más alto inhibe la ejecución de otra instrucción de Modbus_Master, se puede obtener un funcionamiento incorrecto. Las instrucciones de Modbus_Master no deben ejecutarse en los niveles de prioridad para arranque, diagnóstico o error de tiempo.
- Cuando una instrucción Modbus_Master inicia una transmisión, esta instancia debe ejecutarse continuamente con la entrada EN habilitada hasta que se devuelva un estado DONE=1 o un estado ERROR=1. Una instancia Modbus_Master en particular se considera activa hasta que ocurra uno de estos dos eventos. Mientras la instancia original esté activa, cualquier llamada a otra instancia con la salida REQ habilitada resultará en un error. Si la ejecución continua de la instancia original se detiene, el estado de la petición se mantiene activo por un periodo de tiempo especificado por la variable estática "Blocked_Proc_Timeout". Una vez transcurre este periodo de tiempo, la siguiente instrucción Modbus_Master llamada con una entrada REQ habilitada se convertirá en la instancia activa. Esto impide que una sola instancia de Modbus_Master monopolice o cierre el acceso a un puerto. Si la instancia original que está activa no se habilita dentro del periodo de tiempo especificado por la variable estática "Blocked_Proc_Timeout", entonces la siguiente ejecución realizada por esta instancia (con REQ sin activar) borrará el estado activo. Si REQ está activada, entonces esta ejecución inicia una nueva petición Modbus_Master como si ninguna otra instancia estuviese activa.

Parámetro REQ

0 = No hay ninguna petición; 1 = Petición de transmitir datos a esclavo Modbus

Esta entrada se puede controlar con un contacto activado por nivel o por flanco. Siempre que esta entrada esté habilitada, se inicia un autómata finito para garantizar que ningún otro Modbus_Master que use el mismo DB de instancia pueda lanzar una petición mientras no se

complete la petición actual. Todos los demás estados de entrada se capturan y se retienen internamente para la petición actual hasta que se recibe la respuesta o se detecta un error.

Si la misma instancia de Modbus_Master se ejecuta nuevamente con la entrada REQ = 1 antes de que se finalice la petición actual, no se realizarán más transmisiones. No obstante, al finalizar la petición se lanza otra petición siempre que se vuelva a ejecutar Modbus_Master con entrada REQ = 1.

Los parámetros DATA_ADDR y MODE seleccionan el tipo de función Modbus

DATA_ADDR (dirección Modbus inicial en el esclavo): Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus.

La instrucción Modbus_Master utiliza una entrada MODE en vez de una entrada de código de función. La combinación de MODE y la dirección Modbus determinan el código de función utilizado en el mensaje Modbus real. La tabla siguiente muestra la correlación entre el parámetro MODE, el código de función Modbus y el rango de direcciones Modbus.

Tabla 13-82 Funciones Modbus

MODE	Función Modbus	Longitud de datos	Operación y datos	Dirección Modbus
0	01	De 1 a 2000 De 1 a 1992 [1]	Leer bits de salida: De 1 a (1992 o 2000) bits por petición	De 1 a 9999
0	02	De 1 a 2000 De 1 a 1992 [1]	Leer bits de entrada: De 1 a (1992 o 2000) bits por petición	De 10001 a 19999
0	03	De 1 a 125 De 1 a 124 [1]	Leer registros de retención: 1 a (124 o 125) palabras por petición	De 40001 a 49999 o De 400001 a 465535
0	04	De 1 a 125 De 1 a 124 [1]	Leer palabras de entrada: De 1 a (124 o 125) palabras por petición	De 30001 a 39999
104	04	De 1 a 125 De 1 a 124 1	Leer palabras de entrada: De 1 a (124 o 125) palabras por petición	De 00000 a 65535
1	05	1	Escribir un bit de salida: Un bit por petición	De 1 a 9999
1	06	1	Escribir un registro de retención: 1 palabra por petición	De 40001 a 49999 o De 400001 a 465535
1	15	De 2 a 1968 De 2 a 1960 [1]	Escribir varios bits de salida: 2 a (1960 o 1968) bits por petición	De 1 a 9999
1	16	De 2 a 123 De 2 a 122 [1]	Escribir varios registros de retención: De 2 a (122 o 123) palabras por petición	De 40001 a 49999 o De 400001 a 465535
2	15	De 1 a 1968 De 2 a 1960 [1]	Escribir uno o más bits de salida: De 1 a (1960 o 1968) bits por petición	De 1 a 9999
2	16	De 1 a 123 De 1 a 122 [1]	Escribir uno o más registros de retención: De 1 a (122 o 123) palabras por petición	De 40001 a 49999 o De 400001 a 465535
11	11	0	Leer la palabra de estado de la comunicación del esclavo y el contador de eventos. Esta palabra de estado indica ocupado (0 – no ocupado, 0xFFFF - ocupado). El contador de eventos se incrementa cada vez que se finaliza un mensaje correctamente. En esta función se ignoran los dos operandos DATA_ADDR y DATA_LEN de la instrucción Modbus_Master.	

MODE	Función Modbus	Longitud de datos	Operación y datos	Dirección Modbus
80	08	1	Compruebe el estado del esclavo utilizando el código de diagnóstico de datos 0x0000 (test de Loopback – el esclavo envía de regreso la petición) 1 palabra por petición	
81	08	1	Inicializar el contador de eventos del esclavo utilizando un código de diagnóstico de datos 0x000A 1 palabra por petición	
De 3 a 10, de 12 a 79, de 82 a 255			Reservado	

[1] Para el modo "Direccionamiento avanzado", las longitudes máximas de los datos se reduce en 1 byte o 1 palabra dependiendo del tipo de datos utilizados por la función.

Parámetro DATA_PTR

El parámetro DATA_PTR apunta a la dirección del M o del DB en que se escribe o bien del que se lee. Si se utiliza un bloque de datos, hay que crear un bloque de datos global que ofrezca un almacenamiento de datos para operaciones de lectura y escritura en esclavos Modbus.

Nota

El tipo de bloque de datos debe permitir un direccionamiento directo.

El bloque de datos debe permitir tanto el direccionamiento directo (absoluto) como el simbólico. Al crear el bloque de datos debe seleccionarse el atributo de acceso "estándar".

A partir de la versión V4.0 de la instrucción Modbus_Master se puede habilitar el atributo del bloque de datos "Acceso optimizado al bloque". Solo se puede utilizar un único elemento o una matriz de elementos en memoria optimizada con los siguientes tipos de datos: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDInt, SInt o WChar.

Estructuras del bloque de datos para el parámetro DATA_PTR

- Estos tipos de datos son válidos para la **lectura de palabras** de las direcciones Modbus de 30001 a 39999, de 40001 a 49999 y de 400001 a 465536, así como para la **escritura de palabras** en las direcciones Modbus 40001 a 49999 y 400001 a 465536.
  - Matriz estándar de tipos de datos WORD, UINT o INT
  - Estructura WORD, UINT o INT con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 bits.
  - Estructura compleja con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 o 32 bits.
- Para la **lectura** y escritura de bits de las direcciones Modbus 00001 a 09999 y lecturas de bits de 10001 a 19999.
  - Matriz estándar de tipos de datos booleanos.
  - Estructura booleana con nombres que incluye variables booleanas con nombres unívocos.
- Aunque no es imprescindible, se recomienda que cada instrucción Modbus_Master tenga su propia área de memoria. El motivo de esta recomendación es que la posibilidad de que se corrompan los datos aumenta si varias instrucciones Modbus_Master están leyendo y escribiendo en la misma área de memoria.
- No es necesario que las áreas de datos de DATA_PTR se encuentren en el mismo bloque de datos global. Es posible crear un bloque de datos con varias áreas para lecturas Modbus, uno para escrituras Modbus, o bien uno para cada estación esclava.

Variables del bloque de datos (DB) de la instrucción Modbus_Master

La tabla siguiente muestra las variables estáticas públicas almacenadas en el DB de instancia Modbus_Master que se pueden utilizar en el programa:

Tabla 13-83 Variables estáticas del DB de instancia Modbus_Master

Variable	Tipo de datos	Valor predeterminado	Descripción
Blocked_Proc_Timeout	Real	3.0	Tiempo (en segundos) que hay que esperar a una instancia Modbus_Master boqueada antes de eliminar esta instancia como ACTIVA. Esto puede ocurrir, por ejemplo, si se ha lanzado una peticiónModbus_Master y luego el programa deja de llamar la función Modbus_Master antes de que haya finalizado por completo la petición. El valor de tiempo debe ser mayor que 0 y menor que 55 segundos; de lo contrario se producirá un error.
Extended_Addressing	Bool	FALSE	Configura el direccionamiento de estaciones esclavas de uno o dos bytes: • FALSE = dirección de un byte; de 0 a 247 • TRUE = dirección de dos bytes (equivale al direccionamiento ampliado); de 0 a 65535
MB_DB	MB_BASE	-	El parámetro MB_DB de la instrucción Modbus_Comm_Load debe estar conectado al parámetro MB_DB de la instrucción Modbus_Master.

El programa puede escribir valores en las variables Blocked_Proc_Timeout y Extended_Addressing para controlar operaciones del Modbus_Master. Consulte la descripción del apartado Modbus_Slave sobre HR_Start_Offset (Página 1098) y Extended_Addressing (Página 1098) para ver un ejemplo de cómo utilizar estas variables en el editor de programas, así como detalles relacionados con el direccionamiento avanzado Modbus.

Códigos de condición

Tabla 13-84 Códigos de condición de ejecución Modbus_Master (errores de comunicación y configuración) [1]

STATUS (W#16#)	Descripción
0000	No hay error
80C8	Timeout del esclavo. El esclavo especificado no ha respondido en el tiempo especificado. Compruebe la velocidad de transferencia, la paridad y el cableado del dispositivo esclavo. Este error solo se notifica una vez que se han ejecutado los intentos configurados.
80C9	La instrucción Modbus_Master ha excedido el tiempo por uno de los motivos siguientes: • La instrucción espera una respuesta del módulo que se utiliza para la comunicación. • El valor Blocked_Proc_Timeout es demasiado pequeño. Este error se notifica si un dispositivo de periferia descentralizada PROFIBUS o PROFINET devuelve una de las siguientes respuestas: • interrupción de la alimentación o la comunicación • evento de extracción/inserción de módulo de comunicación En estos casos, la configuración hardware del PLC se carga de nuevo y Modbus_Comm_Load debe volver a ejecutarse para configurar debidamente el módulo de comunicación.
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE.
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno.
80E1	El mensaje se ha terminado debido a un error de paridad.
80E2	El mensaje se ha terminado debido a un error de trama.
80E3	El mensaje se ha terminado debido a un error de desbordamiento.
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total.
8180	ID de puerto no válido o error en la instrucción Modbus_Comm_Load
8186	Dirección de estación Modbus no válida
8188	Modo no válido especificado para petición Broadcast
8189	Valor de dirección de datos no válido
818A	Valor de longitud de datos no válido
818B	Puntero no válido al origen/destino de datos local: tamaño incorrecto
818C	Puntero no válido para DATA_PTR o Blocked_Proc_Timeout no válido. El área de datos debe ser una de las siguientes: • DB clásico • Matriz de tipos de datos simples en un DB simbólico o remanente • Memoria M

STATUS (W#16#)	Descripción
8200	El puerto está ocupado porque está procesando una petición de transmisión.
8280	Acuse negativo cuando se lee de un módulo. Comprobar la entrada en el parámetro PORT. Esto puede deberse a la pérdida de un módulo de periferia descentralizada PROFIBUS o PROFINET, ya sea debido a la pérdida de alimentación o comunicación o a la extracción de un módulo.
8281	Acuse negativo cuando se escribe en un módulo. Comprobar la entrada en el parámetro PORT. Esto puede deberse a la pérdida de un módulo de periferia descentralizada PROFIBUS o PROFINET, ya sea debido a la pérdida de alimentación o comunicación o a la extracción de un módulo.

Tabla 13-85 Códigos de condición de ejecución Modbus_Master (errores de protocolo Modbus) [1]

STATUS (W#16#)	Código de respuesta del esclavo	Errores de protocolo Modbus
8380	-	Error CRC
8381	01	Código de función no soportado
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o dirección fuera del rango válido del área DATA_PTR
8384	Más que 03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)
8386	-	El código de función en la respuesta no coincide con el código de la petición.
8387	-	Ha respondido el esclavo incorrecto
8388	-	La respuesta del esclavo a una petición de escritura es incorrecta. La petición de escritura devuelta por el esclavo no coincide con lo que el maestro ha enviado realmente.

[1] Además de los errores Modbus_Master indicados arriba, las instrucciones de comunicación PtP subyacentes pueden devolver errores.

Nota

Ajuste de la longitud máxima del registro para la comunicación Profibus

Al utilizar un módulo maestro Profibus CM1243-5 para controlar un dispositivo Profibus ET 200SP o ET 200MP que utilice un módulo punto a punto RS232, RS422 o RS485, es necesario ajustar explícitamente la variable del bloque de datos "max_record_len" a 240, como se describe a continuación:

Ajustar "max_record_len" en la sección Send_P2P del DB de instancia (por ejemplo, "Modbus_Master_DB".Send_P2P.max_record_len) a 240 tras ejecutar Modbus_Comm_Load.

Solo es necesario asignar max_record_len si se utiliza la comunicación Profibus; la comunicación Profinet ya utiliza un valor max_record_len válido.

Instrucción Modbus_Slave (Comunicar vía SIPLUS I/O o el puerto PtP como esclavo Modbus RTU)

Tabla 13-86 Instrucción Modbus_Slave

KOP / FUP	SCL	Descripción

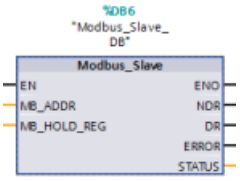
	`"Modbus_Slave_DB"(`
	`MB_ADDR:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		La instrucción Modbus_Slave permite al programa comunicarse de dos maneras:
		• como un esclavo Modbus RTU a través de un puerto PtP en el CM (RS485 o RS232) y la CB (RS485)
		• como un esclavo Modbus RTU mediante las opciones Modbus RTU SIPLUS I/O:
		– Instalar ET 200MP S7-1500CM PtP (RS485 / 422 o RS232).
		– Instalar ET 200SP S7-1500 CM PtP (RS485 / 422 o RS232).
		Cuando un maestro Modbus RTU remoto lanza una petición, el programa de usuario responde con la ejecución de Modbus_Slave. STEP 7 crea automáticamente un DB de instancia al introducir la instrucción. Utilice este nombre de Modbus_Slave_DB al especificar el parámetro MB_DB para la instrucción Modbus_Comm_Load.

Tabla 13-87 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
MB_ADDR	IN	V1.0: USInt V2.0: UInt	La dirección de estación del esclavo Modbus: Rango de direccionamiento estándar (de 1 a 247) Rango de direccionamiento avanzado (de 0 a 65535)
MB_HOLD_REG	IN_OUT	Variant	Puntero hacia el DB del registro de retención Modbus: El registro de retención Modbus puede ser un área de marcas o un bloque de datos.
NDR	OUT	Bool	Nuevos datos listos: • 0 – No hay datos nuevos • 1 – Indica que el maestro Modbus ha escrito datos nuevos
DR	OUT	Bool	Lectura de datos: • 0 – No se han leído datos • 1 – Indica que el maestro Modbus ha leído datos
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. Si la ejecución ha finalizado con un error, el valor del código de error en el parámetro STATUS solo es válido durante un ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de error de ejecución

Los códigos de las funciones de comunicación Modbus (1, 2, 4, 5 y 15) pueden leer y escribir bits y palabras directamente en la memoria imagen de proceso de las entradas y salidas de la CPU. Para estos códigos de función, el parámetro MB_HOLD_REG debe definirse como un tipo de

datos superior a un byte. La tabla siguiente muestra un ejemplo del mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

Tabla 13-88 Mapeo de las direcciones Modbus en la memoria imagen de proceso

Funciones Modbus						**S7-1200**	
Códigos	Función	Área de datos	Rango de direcciones			Área de datos	Dirección de la CPU
01	Leer bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
02	Leer bits	Entrada	10001	a	18192	Memoria imagen de proceso de las entradas	I0.0 a I1023.7
04	Leer palabras	Entrada	30001	a	30512	Memoria imagen de proceso de las entradas	IW0 a IW1022
05	Escribir bit	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
15	Escribir bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7

Los códigos de las funciones de comunicación Modbus (3, 6, 16) utilizan un registro de retención Modbus que puede ser un rango de dirección de memoria M o un bloque de datos. El tipo de registro de retención se especifica con el parámetro MB_HOLD_REG de la instrucción Modbus_Slave.

Nota

Tipo de bloque datos MB_HOLD_REG

Un bloque de datos del registro de retención Modbus debe permitir tanto el direccionamiento directo (absoluto) como el simbólico. Al crear el bloque de datos debe seleccionarse el atributo de acceso "estándar".

A partir de la versión V4.0 de la instrucción Modbus_Slave se puede habilitar el atributo del bloque de datos "Acceso optimizado al bloque". Solo se puede utilizar un único elemento o una matriz de elementos en memoria optimizada con los siguientes tipos de datos: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDInt, SInt o WChar.

La tabla siguiente muestra ejemplos del mapeo de las direcciones Modbus en el registro de retención para los códigos de función 03 (leer palabras), 06 (escribir palabras) y 16 (escribir palabras). El límite superior real de las direcciones de DB está determinado por los límites máximos de memoria de trabajo y de memoria M del respectivo modelo de CPU.

Tabla 13-89 Mapeo de las direcciones Modbus en la memoria de la CPU

Dirección del maestro Modbus	**Ejemplos de parámetros 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]

Tabla 13-90 Funciones de diagnóstico

Funciones de diagnóstico Modbus_Slave en el S7-1200		
Códigos	Subfunción	Descripción
08	0000H	Devolver datos de consulta del test de eco: • En las versiones de STEP 7 anteriores a V15.1 Modbus_Slave devuelve una palabra de datos recibida a un maestro Modbus. • A partir de STEP 7 V15.1, la instrucción Modbus_Slave V4.1 o posterior reenvía una o más palabras de datos recibidas.
08	000AH	Borrar contador de eventos de comunicación: La instrucción Modbus_Slave borra el contador de eventos de comunicación utilizado para la función Modbus 11.
11		Consultar contador de eventos de comunicación: La instrucción Modbus_Slave utiliza un contador de eventos de comunicación interno para registrar el número de peticiones de lectura y escritura Modbus correctas que se envían a Modbus_Slave. El contador no se incrementa con las funciones 8 ni 11, ni tampoco con peticiones Broadcast. Tampoco se incrementa con peticiones que resulten en un error de comunicación (p. ej. errores de paridad o CRC).

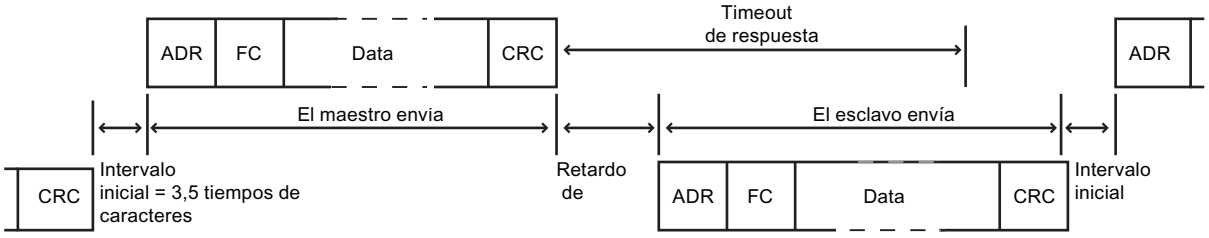
La instrucción Modbus_Slave soporta peticiones de escritura Broadcast de cualquier maestro Modbus, mientras la petición sea para acceder a direcciones válidas. Modbus_Slave generará el código de error "0x8188" para códigos de función no soportados en Broadcast.

Reglas de comunicación Modbus_Slave

- Modbus_Comm_Load debe ejecutarse para configurar un puerto antes de que la instrucción Modbus_Slave pueda comunicarse a través de ese puerto.
- Si un puerto debe responder como esclavo a un Modbus_Master, no programe este puerto con la instrucción Modbus_Master.
- Solo se puede utilizar una instancia Modbus_Slave en un determinado puerto, de lo contrario puede presentarse un comportamiento erróneo.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe controlar el proceso de comunicación consultando la instrucción Modbus_Slave para comprobar si se han finalizado las operaciones de transmisión y recepción.
- La instrucción Modbus_Slavedebe ejecutarse periódicamente a una frecuencia que permita responder sin demora a las peticiones entrantes de un Modbus_Master. Se recomienda ejecutar Modbus_Slave en cada ciclo desde un OB de ciclo de programa. Es posible ejecutar Modbus_Slave desde un OB de alarma cíclica, pero no es recomendable porque el potencial de retardos excesivos en el subprograma bloquea temporalmente la ejecución de otros subprogramas.

Temporización de señales Modbus

Modbus_Slave debe ejecutarse periódicamente para recibir todas las peticiones del Modbus_Master y responder según sea necesario. La frecuencia de ejecución de Modbus_Slave depende del periodo de timeout de respuesta del Modbus_Master. Esto se ilustra en el diagrama siguiente.



El periodo de timeout de respuesta RESP_TO es el tiempo que un Modbus_Master espera hasta el inicio de la respuesta de un Modbus_Slave. Este periodo no está definido en el protocolo Modbus, sino que es un parámetro de todo Modbus_Master. La frecuencia de ejecución (es decir, el tiempo que transcurre entre dos ejecuciones) de Modbus_Slave debe basarse en los parámetros particulares del Modbus_Master. Como mínimo, Modbus_Slave debería ejecutarse dos veces en el periodo de timeout de respuesta del Modbus_Master.

Variables del bloque de datos (DB) de la instrucción Modbus_Slave

La tabla siguiente muestra las variables estáticas públicas almacenadas en el DB de instancia Modbus_Slave que se pueden utilizar en el programa:

Tabla 13-91 Variables estáticas del DB de instancia Modbus_Slave

Variable	Tipo de datos	Valor predeterminado	Descripción
HR_Start_Offset	Word	0	Asigna la dirección de inicio del registro de retención Modbus (valor predeterminado = 0)
Extended_Addressing	Bool	FALSE	Configura el direccionamiento de esclavo de uno o dos bytes: • FALSE = dirección de un byte • TRUE = dirección de dos bytes
Request_Count	Word	0	Número total de peticiones recibidas por este esclavo
Slave_Message_Count	Word	0	Número de peticiones recibidas para este esclavo específico
Bad_CRC_Count	Word	0	Número de peticiones recibidas que tienen un error CRC
Broadcast_Count	Word	0	Número de peticiones Broadcast recibidas
Exception_Count	Word	0	Errores específicos de Modbus que requieren reconocimiento con una excepción devuelta al maestro
Success_Count	Word	0	Número de peticiones recibidas para este esclavo específico que no tienen errores de protocolo
MB_DB	MB_BASE	-	El parámetro MB_DB de la instrucción Modbus_Comm_Load debe estar conectado al parámetro MB_DB de la instrucción Modbus_Slave.
QB_Start	UInt	0	Dirección inicial de los bytes de salida en los que la CPU puede escribir (QB0 a QB65535)

Variable	Tipo de datos	Valor predeterminado	Descripción
QB_Count	UInt	65535	El número de bytes en los que puede escribir un dispositivo remoto. Si QB_Count = 0, un dispositivo remoto no podrá escribir en las salidas. Ejemplo: Para permitir la escritura solo en QB10 a QB17, QB_Start = 10 y QB_Count = 8.
QB_Read_Start	UInt	0	Dirección inicial de los bytes de salida en los que la CPU puede leer (QB0 a QB65535)
QB_Read_Count	UInt	65535	El número de bytes de salida en los que puede leer un dispositivo remoto. Si QB_Count = 0, un dispositivo remoto no podrá leer las salidas. Ejemplo: Para permitir la lectura solo en QB10 a QB17, QB_Start = 10 y QB_Count = 8.
IB_Read_Start	UInt	0	La dirección inicial de los bytes de entrada en los que la CPU puede leer (IB0 a IB65535)
IB_Read_Count	UInt	65535	El número de bytes de entrada en los que puede leer un dispositivo remoto. Si IB_Count = 0, un dispositivo remoto no podrá leer las entradas. Ejemplo: Para permitir la lectura solo en IB10 a IB17, IB_Start = 10 e IB_Count = 8.

El programa puede escribir datos en las operaciones de control del servidor Modbus y en las variables siguientes:

- HR_Start_Offset
- Extended_Addressing
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- IB_Read_Start
- IB_Read_Count

Los requisitos de la versión para la disponibilidad de variables de bloque de datos (DB) de la instrucción Modbus_Slave son los siguientes:

Tabla 13-92 Requisitos para la disponibilidad de variables del bloque de datos (DB) de la instrucción Modbus_Slave: instrucción, TIA Portal y CPU S7-1200

Versión de la instrucción Modbus_Slave	Versión del TIA Portal	Versión del firmware (FW) de la CPU S7-1200	Variables de bloque de datos
3.0	V14 SP1	FW de la CPU V4.0 o posterior	QB_Start
			QB_Count
4.0 o posterior	V15 o posterior	FW de la CPU V4.2 o posterior	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count

HR_Start_Offset

Las direcciones del registro de retención Modbus empiezan en 40001 o 400001. Estas direcciones corresponden a la dirección de memoria del PLC inicial del registro de retención. No obstante, es posible configurar la variable "HR_Start_Offset" para que defina la dirección inicial del registro de retención Modbus en otro valor diferente a 40001 o 400001.

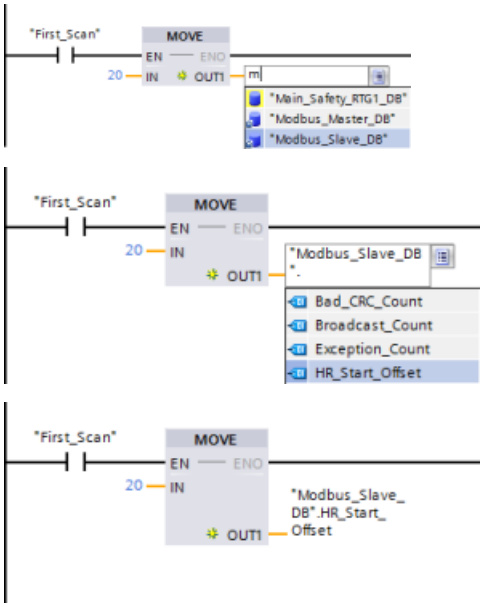
Por ejemplo, si ha configurado el registro de retención para que empiece en MW100 y su longitud es de 100 palabras. Un offset de 20 determina que la dirección inicial del registro de retención sea 40021 en vez de 40001. Cualquier dirección por debajo de 40021 y por encima de 400119 tendrá como resultado un error de direccionamiento.

Tabla 13-93 Ejemplo de direccionamiento del registro de retención Modbus

HR_Start_Offset	Dirección	Mínimo	Máximo
0	Dirección Modbus (palabra)	40001	40099
	Dirección S7-1200	MW100	MW298
20	Dirección Modbus (palabra)	40021	40119
	Dirección S7-1200	MW100	MW298

HR_Start_Offset es un valor de palabra que especifica la dirección inicial del registro de retención Modbus y que se almacena en el bloque de datos de instancia Modbus_Slave. Puede ajustar el valor de esta variable estática pública utilizando la lista desplegable de asistencia de parámetros, una vez que haya insertado Modbus_Slave en el programa.

Por ejemplo, una vez que haya insertado Modbus_Slave en un segmento KOP, puede ir a un segmento anterior y asignar el valor HR_Start_Offset. El valor debe asignarse antes de ejecutar Modbus_Slave.



Introducir una variable de esclavo Modbus utilizando el nombre de DB predeterminado:

1. Posicione el cursor en el campo del parámetro y escriba un carácter m.
2. Seleccione "Modbus_Slave_DB" en la lista desplegable.
3. Posicione el cursor en el lado derecho del nombre del DB (detrás de las comillas) e introduzca un punto.
4. Seleccione "Modbus_Slave_DB.HR_Start_Offset" en la lista desplegable.

Extended_Addressing

A la variable Extended_Addressing se accede de una manera similar que a la variable HR_Start_Offset mencionada más arriba, excepto que la variable Extended_Addressing es un valor booleano. El valor booleano se debe escribir a través de una bobina de salida y no de un cuadro de desplazamiento.

El direccionamiento del esclavo Modbus se puede configurar de modo que tenga un solo byte (ajuste predeterminado) o bien dos bytes. El direccionamiento avanzado se utiliza para direccionar más de 247 dispositivos dentro de una sola red. El direccionamiento avanzado permite direccionar como máximo 64000 direcciones. A continuación aparece una trama de la función 1 Modbus a modo de ejemplo.

Tabla 13-94 Dirección de esclavo de un byte (byte 0)

Función 1	**Byte 0**	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	
Petición	Direc. esclava	Código F	Dirección inicial		Long. de bobinas		
Respuesta válida	Direc. esclava	Código F	Longitud	Datos bobina			
Respuesta errónea	Direc. esclava	0x81	Código E				

Tabla 13-95 Dirección de esclavo de dos bytes (byte 0 y byte 1)

	Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**
Petición	Dirección de esclavo		Código F	Dirección inicial		Long. de bobinas	
Respuesta válida	Dirección de esclavo		Código F	Longitud	Datos bobina		
Respuesta errónea	Dirección de esclavo		0x81	Código E			

Acceso a áreas de datos de bloques de datos (DB) en lugar de acceso directo a direcciones Modbus

A partir de la versión V4.0 de la instrucción Modbus_Slave y la versión de firmware (FW) V4.2 de la CPU S7-1200 se puede acceder a áreas de datos de DB en lugar de acceder directamente a las memorias imagen de proceso y los registros de retención. Para hacerlo hay que deseleccionar las casillas de verificación "Depositar solo en la memoria de carga" y "Acceso optimizado al bloque" en la página de propiedades "Atributos" del DB global.

Si llega una petición Modbus y no ha definido un área de datos para el tipo de datos Modbus del código de función correspondiente, la instrucción Modbus_Slave tratará la petición como en las versiones anteriores de la instrucción: se accede directamente a memorias imagen de proceso y registros de retención.

Si ha definido un área de datos para el tipo de datos Modbus del código de función, la instrucción Modbus_Slave leerá o escribirá en esa área de datos. Si lee o escribe dependerá del tipo de orden.

Nota

Si se configura un área de datos, la instrucción Modbus_Slave ignora los offsets o rangos configurados por las variables estáticas en el bloque de datos de instancia que corresponde al tipo de datos del área de datos. Dichos offsets y rangos solo son aplicables a la memoria imagen de proceso o a la memoria referenciada por MB_HOLD_REG. Los parámetros de inicio y longitud del área de datos proporcionan su propia forma de definir offsets y rangos.

Para una petición Modbus individual, solo se puede leer de o escribir en un área de datos. Si se desea, por ejemplo, leer registros de retención que se extienden a lo largo de varias áreas de datos, se requieren varias peticiones Modbus.

Estas son las normas para definir áreas de datos:

- Se pueden definir hasta ocho áreas de datos en distintos DB; cada DB solo puede contener un área de datos. Una petición MODBUS única solo puede leer en exactamente una única área de datos o escribir en exactamente un área de datos. Cada área de datos corresponde a un área de direcciones MODBUS. Las áreas de datos se definen en la variable estática "Data_Area_Array" del DB de instancia.
- Si desea utilizar menos de ocho áreas de datos, posicione las áreas de datos necesarias una detrás de otra sin dejar huecos. La primera entrada vacía de las áreas de datos finaliza la búsqueda del área de datos durante el procesamiento. Si define, por ejemplo, los elementos 1, 2, 4 y 5, "Data_Area_Array" solo reconocerá los elementos 1 y 2, mientras que el elemento 3 está vacío.

- El campo Data_Area_Array consta de ocho elementos: Data_Area_Array[1] a Data_Area_Array[8]
- Cada elemento de campo Data_Area_Array[x], 1 <= x <= 8, es un UDT del tipo MB_DataArea y está estructurado del siguiente modo:

Parámetro	Tipo de datos	Significado
data_ type	UInt	Identificador del tipo de datos MODBUS mapeado a esta área de datos: • 0: Identificador de un elemento de campo vacío o un área de datos no utilizada. En este caso, los valores de db, start y length son irrelevantes. • 1: Salida de la memoria imagen de proceso (utilizada con los códigos de función 1, 5 y 15) • 2: Entrada de la memoria imagen de proceso (utilizada con el código de función 2) • 3: Registro de retención (utilizado con los códigos de función 3, 6 y 16) • 4: Registro de entrada (utilizado con el código de función 4) Nota: Si ha definido un área de datos para un tipo de datos MODBUS, la instrucción Modbus_Slave no podrá seguir accediendo directamente a este tipo de datos MODBUS. Si la dirección de una petición MODBUS para un tipo de datos de ese tipo no se corresponde con un área de datos definida, se devolverá un valor W#16#8383 en STATUS.
db	UInt	Número del bloque de datos al que se mapean el registro MODBUS o los bits definidos El número de DB debe ser único en las áreas de datos. El mismo número de DB no puede estar definido en varias áreas de datos. Hay que deseleccionar las casillas de verificación "Depositar solo en la memoria de carga" y "Acceso optimizado al bloque" en la página de propiedades "Atributos" del DB global. Las áreas de datos también comienzan por la dirección de byte 0 del DB. Valores permitidos: de 1 a 60999
start	UInt	Primera dirección MODBUS mapeada al bloque de datos comenzando por la dirección 0.0 Valores permitidos: de 0 a 65535
length	UInt	Número de bits (para los valores 1 y 2 de data_type) o el número de registros (para los valores 3 y 4 de data_type) Las áreas de direcciones MODBUS de un mismo tipo de datos MODBUS no pueden solaparse. Valores permitidos: de 1 a 65535

Ejemplos de definición de áreas de datos:

- Primer ejemplo: data_type = 3, db = 1, start = 10, length = 6
  La CPU mapea los registros de retención (data_type = 3) en el bloque de datos 1 (db = 1) y coloca la dirección Modbus 10 (start = 10) en la palabra de datos 0 y la última dirección Modbus válida 15 (length = 6) en la palabra de datos 5.
- Segundo ejemplo: data_type = 2, db = 15, start = 1700, length = 112
  La CPU mapea las entradas (data_type = 2) en el bloque de datos 15 (db = 15) y coloca la dirección Modbus 1700 (start = 1700) en la palabra de datos 0 y la última dirección Modbus válida 1811 (length = 112) en la palabra de datos 111.

Códigos de condición

Tabla 13-96 Códigos de condición de ejecución Modbus_Slave (errores de comunicación y configuración) [1]

STATUS (W#16#)	Descripción
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE.
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno.
80E1	El mensaje se ha terminado debido a un error de paridad.
80E2	El mensaje se ha terminado debido a un error de trama.
80E3	El mensaje se ha terminado debido a un error de desbordamiento.
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total.
8180	ID de puerto no válido o error en la instrucción Modbus_Comm_Load
8186	Dirección de estación Modbus no válida
8187	Puntero no válido al DB MB_HOLD_REG: Área demasiado pequeña
818C	Puntero MB_HOLD_REG no válido. El área de datos debe ser una de las siguientes: • DB clásico • Matriz de tipos de datos simples en un DB simbólico o remanente • Memoria M

Tabla 13-97 Códigos de condición de ejecución Modbus_Slave (errores de protocolo Modbus) [1]

STATUS (W#16#)	Código de respuesta del esclavo	Errores de protocolo Modbus
8380	Sin respuesta	Error CRC
8381	01	Código de función no soportado o no soportado en Broadcast
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o dirección fuera del rango válido del área DATA_PTR
8384	03	Error de valor de datos

STATUS (W#16#)	Código de respuesta del esclavo	Errores de protocolo Modbus
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)
8389		Definición no válida del área de datos: • Valor data_type no válido • Número de DB no válido o no existe: – Valor de DB no válido – El número de DB no existe – El número de DB ya se utiliza en otra área de datos – DB con acceso optimizado – El DB no está en la memoria de trabajo • Valor de longitud no válido • Solapamiento de rangos de direcciones MODBUS que pertenecen al mismo tipo de datos MODBUS

[1] Además de los errores Modbus_Slave indicados arriba, las instrucciones de comunicación PtP subyacentes pueden devolver errores.

Nota

Ajuste de la longitud máxima del registro para la comunicación PROFIBUS

Al utilizar un módulo maestro PROFIBUS CM1243-5 para controlar un dispositivo PROFIBUS ET 200SP o ET 200MP que utilice un módulo punto a punto RS232, RS422 o RS485, es necesario ajustar explícitamente la variable del bloque de datos "max_record_len" a 240, como se describe a continuación:

Ajustar "max_record_len" en la sección Send_P2P del DB de instancia (por ejemplo, "Modbus_Slave_DB".Send_P2P.max_record_len) a 240 tras ejecutar Modbus_Comm_Load.

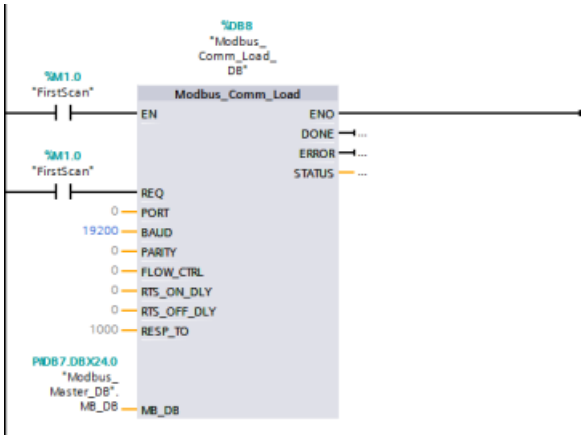
Solo es necesario asignar max_record_len si se utiliza la comunicación PROFIBUS; la comunicación PROFINET ya utiliza un valor max_record_len válido.

13.5.3.5 Ejemplos de Modbus RTU

Ejemplo: programa de maestro Modbus RTU

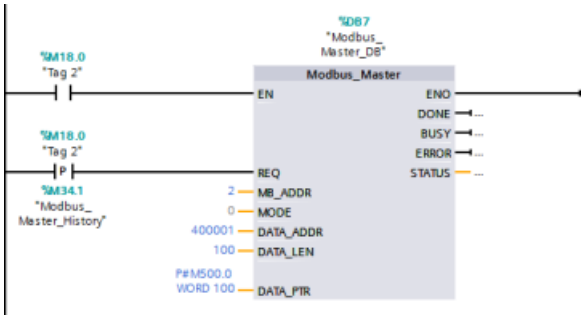
Modbus_Comm_Load se inicializa durante el arranque mediante la primera marca de exploración. La ejecución de Modbus_Comm_Load de esa forma solo debe hacerse cuando la configuración de puerto serie no vaya a cambiar en tiempo de ejecución.

Segmento 1: Configurar/inicializar el puerto de comunicación del módulo RS485 una sola vez durante el primer ciclo.

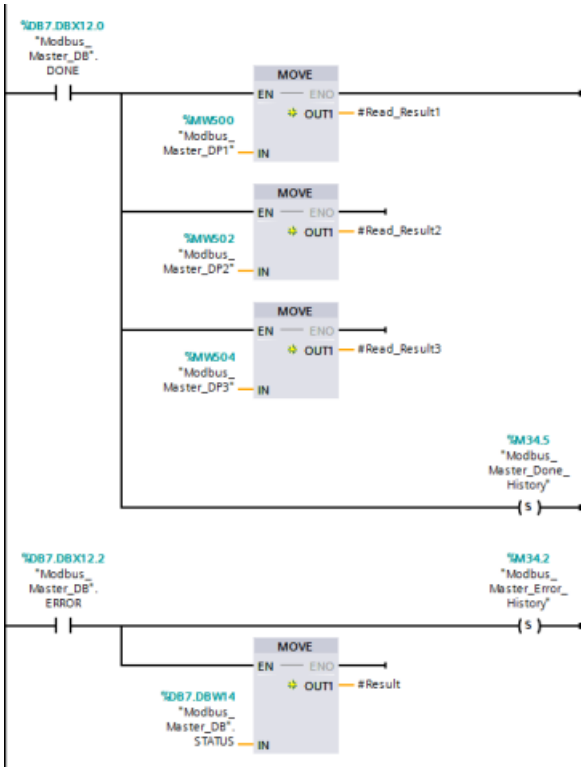


Una instrucción Modbus_Master se usa en el OB del ciclo de programa para comunicar con un solo esclavo. Se pueden usar más instrucciones Modbus_Master en el OB del ciclo de programa para comunicar con otros esclavos o se puede reutilizar un FB Modbus_Master para comunicar con más esclavos.

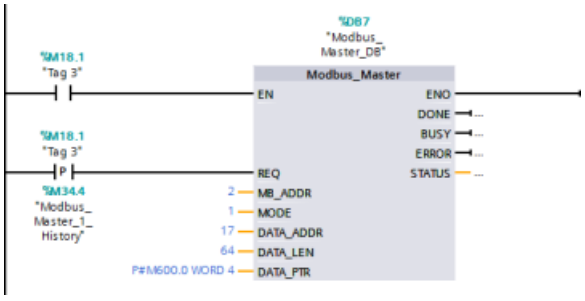
Segmento 2: Leer 100 palabras de los datos del registro de retención desde la ubicación 400001 en el esclavo #2 hasta la ubicación de memoria MW500-MW698.



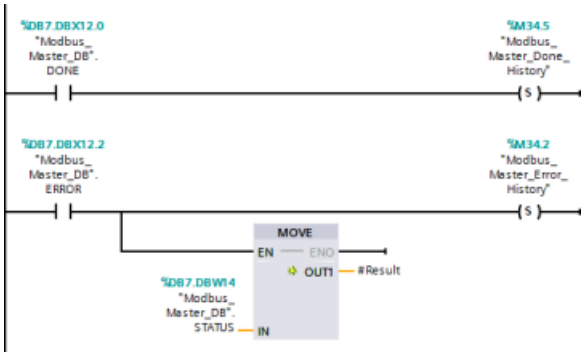
Segmento 3: Mover las 3 primeras palabras de los datos del registro de retención que se han leído a otra ubicación y activar un bit de historial DONE. Este segmento también activa un bit de historial ERROR y guarda la palabra STATUS en otra ubicación en caso de error.



Segmento 4: Escribir 64 bits de datos desde MW600-MW607 a las ubicaciones del bit de salida 00017 a 00081 en el esclavo #2.



Segmento 5: Activar un bit de historial DONE cuando haya finalizado la escritura. Si se produce un error, el programa activa un bit de historial ERROR y guarda el código STATUS.

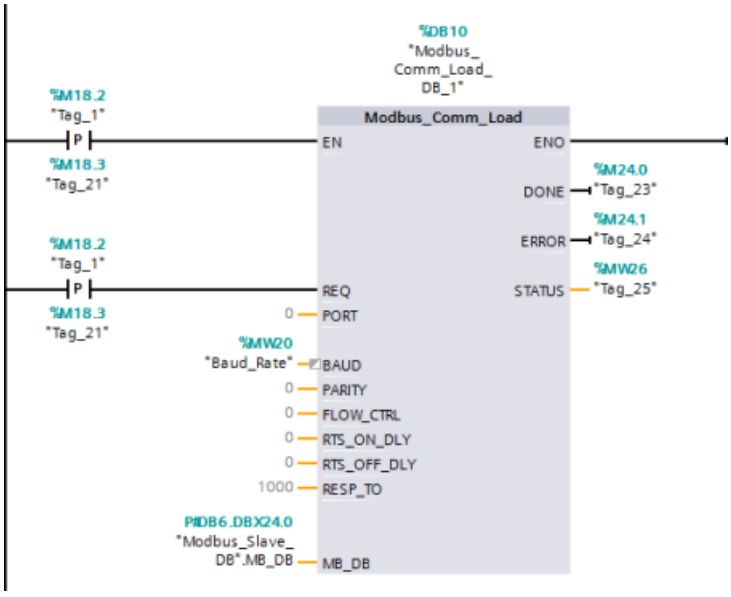


Ejemplo: programa de esclavo Modbus RTU

El MB_COMM_LOAD mostrado a continuación se inicializa cada vez que se habilita "Tag_1".

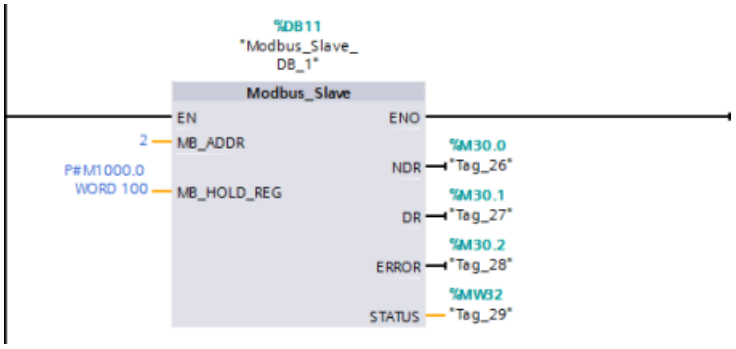
La ejecución de MB_COMM_LOAD de esa forma solo debe hacerse cuando la configuración de puerto serie cambie en tiempo de ejecución, a consecuencia de la configuración de HMI.

Segmento 1: inicializar los parámetros del módulo RS485 cada vez que un dispositivo HMI los cambia.



El MB_SLAVE mostrado a continuación se coloca en un OB cíclico que se ejecuta cada 10 ms. Con esto no se consigue que el esclavo proporcione la respuesta más rápida posible, pero proporciona buen rendimiento a 9600 baudios para mensajes cortos (20 bytes o menos en la petición).

Segmento 2: comprobar las peticiones del maestro Modbus en cada ciclo. El registro de retención Modbus está configurado para 100 palabras comenzando en MW1000.



13.6 Comunicación Legacy PtP (solo CM/CB 1241)

Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones de comunicación punto a punto existían con nombres diferentes y, en algunos casos, con interfaces ligeramente diferentes. Los conceptos generales sobre la comunicación punto a punto (Página 939), así como el puerto (Página 943) y la configuración de parámetros (Página 959) se aplican para ambos juegos de instrucciones. Consulte las instrucciones Legacy individuales punto a punto para obtener información sobre la programación.

Tabla 13-98 Clases de error comunes

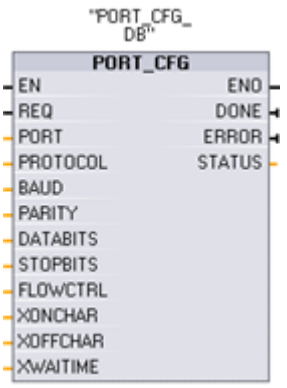
Descripción de la clase	Clases de error	Descripción
Configuración del puerto	80Ax	Permite definir errores comunes de configuración del puerto
Configuración de la transmisión	80Bx	Permite definir errores comunes de configuración de la transmisión
Configuración de la recepción	80Cx	Permite definir errores comunes de configuración de la recepción
Tiempo de ejecución de la transmisión	80Dx	Permite definir errores comunes en tiempo de ejecución de la transmisión
Tiempo de ejecución de la recepción	80Ex	Permite definir errores comunes en tiempo de ejecución de la recepción
Procesamiento de señales	80Fx	Permite definir errores comunes en relación con el procesamiento de señales

13.6.1 Instrucciones Legacy punto a punto

13.6.1.1 PORT_CFG (configurar los parámetros de comunicación dinámicamente)

Tabla 13-99 Instrucción PORT_CFG (configuración de puerto)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción PORT_CFG permite cambiar los parámetros de puerto (p. ej. la velocidad de transferencia) desde el programa.
		La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción PORT_CFG puede ejecutarse en el programa con el fin de modificar la configuración.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Los cambios de la configuración de PORT_CFG no se almacenan de forma permanente en la CPU. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU

cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Encontrará más información en Configurar los puertos de comunicación (Página 943) y Gestionar el control de flujo (Página 945).

Tabla 13-100 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de la configuración cuando se detecta un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
PROTOCOL	IN	UInt	0 - Protocolo de comunicación punto a punto (valor predeterminado) 1..n - Definición futura para protocolos específicos
BAUD	IN	UInt	Velocidad de transferencia del puerto (valor predeterminado: 6): 1 = 300 baudios, 2 = 600 baudios, 3 = 1200 baudios, 4 = 2400 baudios, 5 = 4800 baudios, 6 = 9600 baudios, 7 = 19200 baudios, 8 = 38400 baudios, 9 = 57600 baudios, 10 = 76800 baudios, 11 = 115200 baudios
PARITY	IN	UInt	Paridad del puerto (Valor predeterminado: 1): 1 = Sin paridad, 2 = Paridad par, 3 = Paridad impar, 4 = Paridad marca, 5 = Paridad espacio
DATABITS	IN	UInt	Bits por carácter (valor predeterminado:1): 1 = 8 bits de datos, 2 = 7 bits de datos
STOPBITS	IN	UInt	Bits de parada (valor predeterminado: 1): 1 = 1 bit de parada, 2 = 2 bits de parada
FLOWCTRL	IN	UInt	Control de flujo (valor predeterminado: 1): 1 = Sin control de flujo, 2 = XON/XOFF, 3 = RTS de hardware siempre ON, 4 = RTS de hardware conmutado
XONCHAR	IN	Char	Determina el carácter que se utiliza como carácter XON. Generalmente, es un carácter DC1 (16#11). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 16#11)
XOFFCHAR	IN	Char	Determina el carácter que se utiliza como carácter XOFF. Generalmente, es un carácter DC3 (116#3). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 16#13)
XWAITIME	IN	UInt	Determina cuánto tiempo se debe esperar un carácter XON tras recibir un carácter XOFF o cuánto tiempo se debe esperar la señal CTS tras habilitar RTS (0 a 65535 ms). Este parámetro se evalúa solo si está habilitado el control de flujo. (Valor predeterminado: 2000)
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

Tabla 13-101 Códigos de condición

STATUS (W#16#....)	Descripción
80A0	El protocolo indicado no existe.
80A1	La velocidad de transferencia indicada no existe.
80A2	La opción de paridad indicada no existe.
80A3	El número de bits de datos indicado no existe.
80A4	El número de bits de parada indicado no existe.
80A5	El tipo de control de flujo indicado no existe.
80A6	El tiempo de espera es 0 y el control de flujo está habilitado
80A7	XON y XOFF son valores no permitidos (p. ej. el mismo valor)

13.6.1.2 SEND_CFG (configurar los parámetros de transmisión en serie dinámicamente)

Tabla 13-102 Instrucción SEND_CFG (configuración de transmisión)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción SEND_CFG permite configurar dinámicamente los parámetros de transmisión serie de un puerto de comunicación PtP. Todos los mensajes en cola de espera en un CM o una CB se rechazarán cuando se ejecute SEND_CFG.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La configuración estática inicial del puerto puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción SEND_CFG puede ejecutarse en el programa con el fin de modificar la configuración.

Los cambios de la configuración de SEND_CFG no se almacenan de forma permanente en la CPU. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Véase Configurar los parámetros de transmisión.

Tabla 13-103 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)

Parámetro y tipo		Tipo de datos	Descripción
RTSONDLY	IN	UInt	Número de milisegundos que debe esperarse tras la habilitación de RTS antes de cualquier transmisión de datos Tx. Este parámetro solo es válido si está habilitado el control de flujo por hardware. El rango válido es de 0 a 65535 ms. Un valor 0 desactiva la función. (Valor predeterminado: 0)
RTSOFFDLY	IN	UInt	Número de milisegundos que se debe esperar después de la transmisión de datos Tx antes de que se inhiba RTS: Este parámetro solo es válido si está habilitado el control de flujo por hardware. El rango válido es de 0 a 65535 ms. Un valor 0 desactiva la función. (Valor predeterminado: 0)
BREAK	IN	UInt	Este parámetro indica que se enviará una pausa tras el inicio de cada mensaje durante el número de tiempos de bit indicado. El máximo es 65535 tiempos de bit hasta ocho segundos como máximo. Un valor 0 desactiva la función. (Valor predeterminado: 12)
IDLELINE	IN	UInt	Este parámetro indica que la línea permanecerá inactiva durante el número de tiempos de bit indicado hasta el inicio de cada mensaje. El máximo es 65535 tiempos de bit hasta ocho segundos como máximo. Un valor 0 desactiva la función. (Valor predeterminado: 12)
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

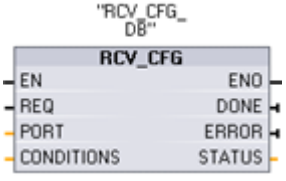
Tabla 13-104 Códigos de condición

STATUS (W#16#....)	Descripción
80B0	No se permite configurar una alarma de transmisión.
80B1	El tiempo de pausa excede el valor máximo permitido.
80B2	El tiempo de inactividad excede el valor máximo permitido.

13.6.1.3 RCV_CFG (configurar parámetros de recepción serie dinámicamente)

Tabla 13-105 Instrucción RCV_CFG (configuración de recepción)

KOP / FUP	SCL	Descripción

	`"RCV_CFG_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`CONDITIONS:=_struct_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		La instrucción RCV_CFG permite configurar dinámicamente los parámetros de recepción serie de un puerto de comunicación PtP. Esta instrucción configura las condiciones que indican el inicio y fin de un mensaje recibido. Todos los mensajes en cola de espera en un CM o una CB se rechazarán cuando se ejecute RCV_CFG.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

La configuración estática inicial del puerto de comunicación puede ajustarse en las propiedades de la configuración de dispositivos. Como alternativa, es posible utilizar los valores predeterminados. La instrucción RCV_CFG puede ejecutarse en el programa con el fin de modificar la configuración.

Los cambios de la configuración de RCV_CFG no se almacenan de forma permanente en la CPU. Los parámetros ajustados en la configuración de dispositivos se restablecen cuando la CPU cambia de RUN a STOP y tras desconectar y volver a conectar la alimentación. Encontrará más información en Configurar los parámetros de recepción (Página 947).

Tabla 13-106 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa el cambio de configuración cuando se produce un flanco ascendente en esta entrada. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
CONDITIONS	IN	CONDITIONS	La estructura de datos CONDITIONS indica las condiciones de inicio y fin del mensaje, como se describe a continuación.
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

Condiciones de inicio de la instrucción RCV_PTP

La instrucción RCV_PTP utiliza la configuración indicada por la instrucción RCV_CFG para determinar el inicio y fin de los mensajes de comunicación punto a punto. Las condiciones de inicio determinan el inicio de un mensaje. Una o más condiciones de inicio pueden determinar el inicio de un mensaje. Si se especifica más de una condición de inicio, todas las condiciones deberán cumplirse antes de iniciar el mensaje.

En el apartado "Configurar los parámetros de recepción (Página 947)" encontrará una descripción de las condiciones de inicio del mensaje.

Estructura de los tipos de datos del parámetro CONDITIONS, 1ª parte (condiciones de inicio)

Tabla 13-107 Estructura de CONDITIONS para condiciones de START

Parámetro y tipo		Tipo de datos	Descripción
STARTCOND	IN	UInt	Indica la condición de inicio (valor predeterminado: 1) • 01H - Carácter inicial • 02H - Cualquier carácter • 04H - Line Break • 08H - Idle Line • 10H - Secuencia 1 • 20H - Secuencia 2 • 40H - Secuencia 3 • 80H - Secuencia 4
IDLETIME	IN	UInt	Número de tiempos de bit necesario para el timeout por "idle line". (Valor predeterminado: 40). Se utiliza únicamente para una condición de "idle line". 0 a 65535
STARTCHAR	IN	Byte	Carácter de inicio utilizado con la condición de carácter de inicio. (Valor predeterminado: B#16#2)
SEQ[1].CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 1: (Valor predeterminado: B#16#0) Estos son los bits de habilitación de cada carácter de la secuencia de inicio. • 01H - Carácter 1 • 02H - Carácter 2 • 04H - Carácter 3 • 08H - Carácter 4 • 10H - Carácter 5 Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
SEQ[1].STR	IN	Char[5]	Caracteres de inicio de la secuencia 1 (5 caracteres). Valor predeterminado: 0
SEQ[2].CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 2. Valor predeterminado: B#16#0)
SEQ[2].STR	IN	Char[5]	Caracteres de inicio de la secuencia 2 (5 caracteres). Valor predeterminado: 0
SEQ[3].CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 3. Valor predeterminado: B#16#0
SEQ[3].STR	IN	Char[5]	Caracteres de inicio de la secuencia 3 (5 caracteres). Valor predeterminado: 0
SEQ[4].CTL	IN	Byte	Ignorar/comparar el control de cada carácter de la secuencia 4. Valor predeterminado: B#16#0
SEQ[4].STR	IN	Char[5]	Caracteres de inicio de la secuencia 4 (5 caracteres), valor predeterminado: 0

Ejemplo

Considere el siguiente mensaje recibido en código hexadecimal: "**68** 10 aa **68** bb 10 aa 16" y las secuencias de inicio configuradas que muestra la tabla siguiente. Las secuencias de inicio se comienzan a evaluar al recibirse correctamente el primer carácter 68H. Tras recibirse correctamente el cuarto carácter (el segundo 68H), se cumple la condición de inicio 1. Una vez cumplidas las condiciones de inicio, se iniciará la evaluación de las condiciones de fin.

El procesamiento de la secuencia de inicio puede cancelarse debido a distintos errores de paridad, trama o tiempo excedido entre caracteres. Debido a estos errores, no se recibe el mensaje puesto que no se ha cumplido la condición de inicio.

Tabla 13-108 Condiciones de inicio

Condición de inicio	Primer carácter	Primer carácter +1	Primer carácter +2	Primer carácter +3	Primer carácter +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

Condiciones de fin de la instrucción RCV_PTP

Las condiciones de fin configuradas determinan el fin de un mensaje. La primera aparición de una o más condiciones de fin configuradas determina el fin de un mensaje. La sección "Condiciones de fin del mensaje" del apartado "Configurar los parámetros de recepción (Página 947)" describe las condiciones de fin que se pueden configurar en la instrucción RCV_CFG.

Las condiciones de fin se pueden configurar tanto en las propiedades de la interfaz de comunicación en la configuración de dispositivos como desde la instrucción RCV_CFG. Cada vez que la CPU realiza una transición de STOP a RUN, los parámetros de recepción (condiciones de inicio y fin) vuelven a los ajustes de la configuración de dispositivos. Si el programa de usuario de STEP 7 ejecuta la instrucción RCV_CFG, los ajustes cambiarán a las condiciones de RCV_CFG.

Estructura de los tipos de datos del parámetro CONDITIONS, 2ª parte (condiciones de fin)

Tabla 13-109 Estructura de CONDITIONS para condiciones de END

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
ENDCOND	IN	UInt 0	Este parámetro define la condición de fin del mensaje: • 01H -Tiempo de respuesta • 02H -Tiempo del mensaje • 04H - Tiempo excedido entre caracteres • 08H - Longitud máxima • 10H - N + LEN + M • 20H - Secuencia
MAXLEN	IN	UInt 1	Longitud máxima del mensaje: Se utiliza solo si está seleccionada la condición de fin "Longitud máxima". 1 a 1024 bytes

Parámetro	Tipo de parámetro	Tipo de datos	Descripción
N	IN	UInt 0	Posición de byte dentro del campo de longitud en el mensaje. Se utiliza solo con la condición de fin N + LEN + M. 1 a 1022 bytes
LENGTHSIZE	IN	UInt 0	Tamaño del campo de longitud (1, 2 o 4 bytes). Se utiliza solo con la condición de fin N + LEN + M.
LENGTHM	IN	UInt 0	Indica el número de caracteres posteriores al campo de longitud que no se incluyen en el valor del campo de longitud. Se utiliza solo con la condición de fin N + LEN + M. 0 a 255 bytes
RCVTIME	IN	UInt 200	Indica cuánto tiempo debe esperarse hasta que se reciba el primer carácter. La recepción se terminará con un error si no se recibe correctamente ningún carácter dentro del tiempo indicado. Se utiliza solo con la condición de fin "Tiempo de respuesta". (de 0 a 65535 tiempos de bit con un máximo de 8 segundos) Este parámetro no es una condición de fin del mensaje dado que la evaluación finaliza cuando se recibe el primer carácter de una respuesta. Es una condición de fin solo en el sentido de que finaliza una operación de recepción debido a que no se recibe ninguna respuesta cuando se espera una. Se debe seleccionar una condición de fin distinta.
MSGTIME	IN	UInt 200	Indica cuánto tiempo debe esperarse hasta la recepción completa del mensaje una vez recibido el primer carácter. Este parámetro se utiliza solo si está seleccionada la condición de fin "Tiempo de mensaje excedido". (de 0 a 65535 milisegundos)
CHARGAP	IN	UInt 12	Indica el número de tiempos de bit entre caracteres. Si el número de tiempos de bit entre caracteres excede el valor especificado, se cumplirá la condición de fin. Se utiliza solo con la condición de fin "Tiempo excedido entre caracteres". (de 0 a 65535 tiempos de bit hasta 8 segundos máx.)
SEQ.CTL	IN	Byte B#16#0	Ignorar/comparar el control de cada carácter de la secuencia 1: Estos son los bits de habilitación de cada carácter de la secuencia de fin. El carácter 1 es el bit 0, el carácter 2 es el bit 1, ..., el carácter 5 es el bit 4. Si se inhibe el bit asociado a un carácter, cualquier carácter será válido en esta posición de la secuencia.
SEQ.STR	IN	Char[5] 0	Caracteres de inicio de la secuencia 1 (5 caracteres)

Tabla 13-110 Códigos de condición

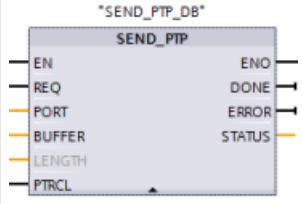
STATUS (W#16#....)	Descripción
80C0	Se ha seleccionado una condición de inicio no permitida
80C1	Se ha seleccionado una condición de fin no permitida o no se ha seleccionado ninguna condición de fin
80C2	Hay una alarma de recepción habilitada, pero esto no es posible.
80C3	La condición de fin "Longitud máxima" está habilitada y la longitud máxima es 0 o > 1024.

STATUS (W#16#....)	Descripción
80C4	La longitud calculada está habilitada y N es >= 1023.
80C5	La longitud calculada está habilitada y la longitud no es 1, 2 o 4.
80C6	La longitud calculada está habilitada y el valor M es > 255.
80C7	La longitud calculada está habilitada y es > 1024.
80C8	El timeout de respuesta está habilitado y es igual a cero.
80C9	El tiempo excedido entre caracteres está habilitado y es igual a cero.
80CA	El timeout por "idle line" está habilitado y es igual a cero.
80CB	La secuencia de fin está habilitada, pero todos los caracteres son irrelevantes.
80CC	La secuencia de inicio (una cualquiera de 4) está habilitada, pero todos los caracteres son irrelevantes.

13.6.1.4 SEND_PTP (transmitir datos del búfer de transmisión)

Tabla 13-111 Instrucción SEND_PTP (enviar datos punto a punto)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción SEND_PTP inicia la transferencia de datos y transmite el búfer asignado a la interfaz de comunicación. El programa de la CPU se sigue ejecutando hasta que el CM o la CB envía los datos a la velocidad de transferencia asignada. Solo una operación de transmisión puede estar pendiente en un momento dado. El CM o la CB devuelve un error si una segunda instrucción SEND_PTP se ejecuta mientras el CM o la CB está transmitiendo un mensaje.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-112 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa la transmisión solicitada cuando se produce un flanco ascendente en esta entrada de habilitación de la transmisión. Esto inicia la transferencia del contenido del búfer a la interfaz de comunicación punto a punto. (Valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de transmisión. (Valor predeterminado: 0) **Nota:** no se soportan los datos booleanos ni las matrices booleanas.
LENGTH [1]	IN	UInt	Longitud de la trama transmitida en bytes (valor predeterminado: 0) Al transmitir una estructura compleja, utilice siempre la longitud 0.
PTRCL	IN	Bool	Reservado para usos futuros

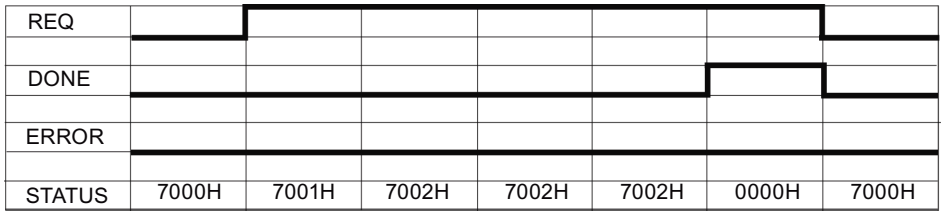
Parámetro y tipo		Tipo de datos	Descripción
DONE	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

[1] Parámetro opcional: haga clic en la flecha situada en la parte inferior de una casilla KOP/FUP para expandir la casilla e incluir este parámetro.

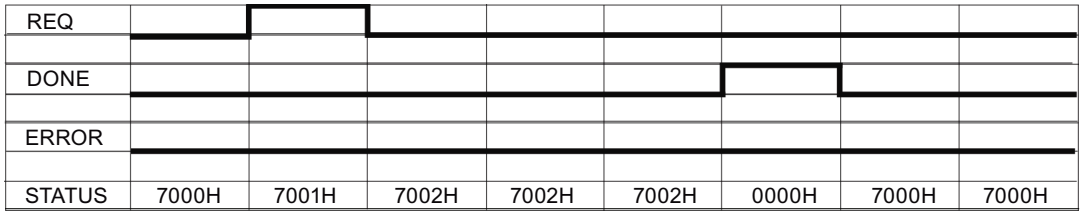
El estado lógico de las salidas DONE y ERROR es FALSE durante la operación de transmisión. Una vez finalizada la operación de transmisión, la salida DONE o ERROR se pone a TRUE para mostrar el estado de la transmisión. Mientras que DONE o ERROR sea TRUE, la salida STATUS será válida.

La instrucción devuelve el estado 16#7001 si la interfaz de comunicación acepta los datos transmitidos. Las ejecuciones posteriores de SEND_PTP devuelven 16#7002 si el CM o la CB aún está ocupada transmitiendo datos. Una vez finalizada la operación de transmisión, el CM o la CB devuelve el estado de transmisión como 16#0000 (si no han ocurrido errores). Las ejecuciones posteriores de SEND_PTP con REQ puesto a "low" devuelven el estado 16#7000 (no ocupado).

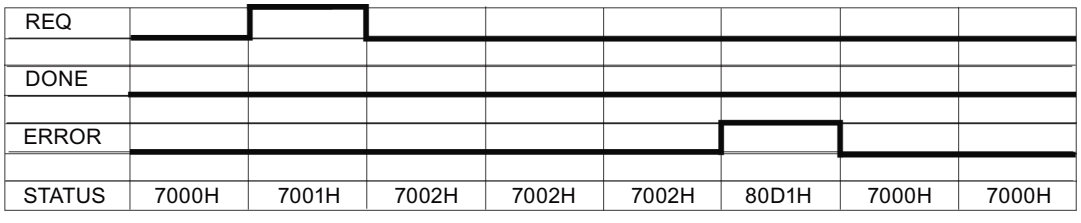
Los diagramas siguientes muestran la relación entre los valores de salida y REQ. Para esto se supone que la instrucción se llama periódicamente para comprobar el estado del proceso de transmisión. En el diagrama siguiente se supone que la instrucción se llama en cada ciclo (representado por los valores de STATUS).



El diagrama siguiente muestra cómo los parámetros DONE y STATUS son válidos solo durante un ciclo si la línea REQ se impulsa (durante un ciclo) para iniciar la operación de transmisión.



El diagrama siguiente muestra la relación de los parámetros DONE, ERROR y STATUS cuando se presenta un error.



Los valores DONE, ERROR y STATUS solo son válidos hasta que se ejecute nuevamente SEND_PTP con el mismo DB de instancia.

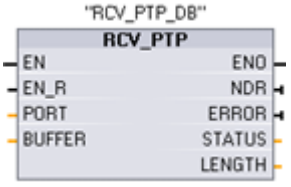
Tabla 13-113 Códigos de condición

STATUS (W#16#....)	Descripción
80D0	Nueva petición estando activo el transmisor
80D1	Transmisión cancelada debido a que no se ha detectado ningún CTS durante el tiempo de espera
80D2	Transmisión cancelada debido a que no se ha recibido DSR del dispositivo DCE
80D3	Transmisión cancelada debido a desbordamiento de la cola de espera (transmisión de más de 1024 bytes)
80D5	Señal de bias inverso (condición de rotura de hilo)
833A	El DB para el parámetro BUFFER no existe.

13.6.1.5 RCV_PTP (habilitar recepción de mensajes)

Tabla 13-114 Instrucción RCV_PTP (recepción punto a punto)

KOP / FUP	SCL	Descripción

	`"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 comprueba si se han recibido mensajes en el CM o la CB. Si hay un mensaje disponible, se transfiere del CM o de la CB a la CPU. Un error devuelve el valor de STATUS apropiado.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-115 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
EN_R	IN	Bool	Si esta entrada es TRUE y un mensaje está disponible, el mensaje se transferirá del CM o de la CB a BUFFER. Si EN_R es FALSE, se comprobará si el CM o la CB ha recibido mensajes y se actualizarán las salidas NDR, ERROR y STATUS, pero el mensaje no se transferirá a BUFFER. (Valor predeterminado: 0)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
BUFFER	IN	Variante	Este parámetro apunta a la dirección inicial del búfer de recepción. Este búfer debería ser lo suficientemente grande para recibir el mensaje de longitud máxima. No se soportan los datos booleanos ni las matrices booleanas. (Valor predeterminado: 0)

Parámetro y tipo		Tipo de datos	Descripción
NDR	OUT	Bool	TRUE durante una ejecución cuando se dispone de datos nuevos y la operación ha finalizado sin errores.
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la operación con un error.
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)
LENGTH	OUT	UInt	Longitud del mensaje devuelto (en bytes) (valor predeterminado: 0)

Observe la siguiente correlación entre la entrada EN_R y el búfer de avisos de la instrucción RCV_PTP:

La entrada EN_R controla la copia de un aviso recibido en el BUFFER.

Si la entrada EN_R es TRUE y el aviso está disponible, la CPU transfiere el aviso desde el CM o la CB al BUFFER y actualiza las salidas NDR, ERROR, STATUS, y LENGTH.

Si EN_R es FALSE, la CPU comprueba si hay avisos en el CM o la CB y actualiza las salidas NDR, ERROR, y STATUS, pero no transfiere el aviso al BUFFER. (Recuerde que el valor predeterminado de EN_R es FALSE.)

La práctica recomendada consiste en poner EN_R a TRUE y controlar la ejecución de la instrucción RCV_PTP con la entrada EN.

El valor de STATUS es válido si NDR o ERROR es TRUE. El valor STATUS proporciona el motivo para finalizar la recepción en el CM o la CB. Por lo general, este valor es positivo, lo cual indica que la recepción se realizó correctamente y que el proceso de recepción finalizó normalmente. Si el valor de STATUS es negativo (es decir, si está activado el bit más significativo del valor hexadecimal), esto indica que la recepción se ha finalizado debido a un error p. ej. de paridad, trama o desbordamiento.

Cada interfaz de comunicación PtP puede almacenar como máximo 1024 bytes en un búfer. Esto podría ser un mensaje grande o varios mensajes pequeños. Si más de un mensaje está disponible en el CM o en la CB, la instrucción RCV_PTP devolverá el mensaje más antiguo disponible. La siguiente ejecución de la instrucción RCV_PTP devolverá el siguiente mensaje más antiguo disponible.

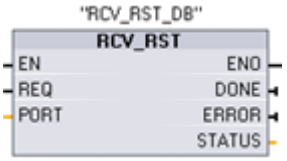
Tabla 13-116 Códigos de condición

STATUS (W#16#...)	Descripción
0000	No hay ningún búfer disponible
0094	Mensaje terminado porque se ha recibido la longitud máxima de caracteres
0095	Mensaje terminado debido a timeout del mensaje
0096	Mensaje terminado porque se ha excedido el tiempo entre caracteres
0097	Mensaje terminado debido a timeout de respuesta
0098	Mensaje terminado porque se ha cumplido la condición de longitud "N+LEN+M"
0099	Mensaje terminado porque se ha cumplido la condición de secuencia de fin
80E0	Mensaje terminado debido a que el búfer de recepción está lleno
80E1	Mensaje terminado debido a un error de paridad
80E2	Mensaje terminado debido a un error de trama
80E3	Mensaje terminado debido a un error de desbordamiento
80E4	Mensaje terminado porque la longitud calculada excede el tamaño del búfer
80E5	Señal de polarización inversa (condición de rotura de hilo)
833A	El DB para el parámetro BUFFER no existe.

13.6.1.6 RCV_RST (borrar búfer de recepción)

Tabla 13-117 Instrucción RCV_RST (inicializar receptor)

KOP / FUP	SCL	Descripción

	`"RCV_RST_DB"(`
	`REQ:=_bool_in_,`
	`PORT:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_);`
		La instrucción RCV_RST borra los búferes de recepción en el CM o la CB.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-118 Tipos de datos para parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Activa la inicialización del receptor cuando se produce un flanco ascendente en esta entrada de habilitación (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
DONE	OUT	Bool	Si es TRUE durante un ciclo, indica que la última petición se ha finalizado sin errores.
ERROR	OUT	Bool	Si es TRUE, indica que la última petición se ha finalizado con errores. Además, si esta salida es TRUE, la salida STATUS contendrá los códigos de error respectivos.
STATUS	OUT	Word	Código de error (valor predeterminado: 0) Consulte los códigos de estado de comunicación en Parámetros comunes de las instrucciones de comunicación punto a punto (Página 959).

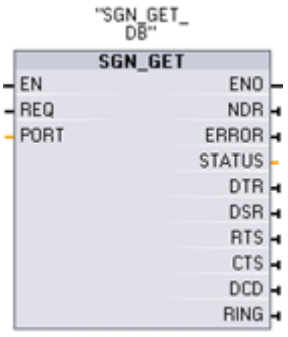
Nota

Puede que desee utilizar la instrucción RCV_RST para asegurarse de que se hayan borrado los búferes de mensajes después de un error de comunicación o después de cambiar un parámetro de comunicación, como por ejemplo la velocidad de transferencia. Al ejecutar RCV_RST el módulo borra todos los búferes de mensajes internos. Después de borrar todos los búferes, puede estar seguro de que cuando un programa ejecute una instrucción de recepción posterior, los mensajes devueltos serán nuevos, en lugar de mensajes antiguos correspondientes a algún momento anterior a la llamada de RCV_RST.

13.6.1.7 SGN_GET (consultar señales RS-232)

Tabla 13-119 Instrucción SGN_GET (consultar señales RS232)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción SGN_GET lee los estados actuales de las señales de comunicación RS232.
		Esta función es válida solo para el CM RS232.

[1] STEP 7 crea automáticamente el DB al introducir la instrucción.

Tabla 13-120 Tipos de datos para parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Los valores de las señales RS232 se consultan cuando se produce un flanco ascendente en esta entrada (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
NDR	OUT	Bool	TRUE durante un ciclo cuando se dispone de datos nuevos y la operación ha finalizado sin errores
ERROR	OUT	Bool	TRUE durante un ciclo tras haberse finalizado la operación con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)
DTR	OUT	Bool	Terminal de datos disponible, módulo listo (salida). Valor predeterminado: False
DSR	OUT	Bool	Equipo de datos listo, interlocutor listo (entrada). Valor predeterminado: False
RTS	OUT	Bool	Petición de transmitir, módulo listo para transmitir (salida). Valor predeterminado: False
CTS	OUT	Bool	Listo para transmitir; el interlocutor puede recibir datos (entrada). Valor predeterminado: False
DCD	OUT	Bool	Detección de portadora de datos, nivel de señal de recepción (siempre False, no se soporta)
RING	OUT	Bool	Indicación de timbre, señalización de una llamada entrante (siempre False, no se soporta)

Tabla 13-121 Códigos de condición

STATUS (W#16#....)	Descripción
80F0	El CM o la CB es un RS485; no hay señales disponibles

13.6.1.8 SGN_SET (activar señales RS-232)

Tabla 13-122 Instrucción SGN_SET (activar señales RS232)

KOP / FUP	SCL	Descripción

	`"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_);`
		La instrucción SGN_SET activa los estados de las señales de comunicación RS232.
		Esta función es válida solo para el CM RS232.

[1] STEP 7 crea el DB automáticamente al insertar la instrucción.

Tabla 13-123 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Inicia la activación de las señales RS232 cuando se produce un flanco ascendente en esta entrada (valor predeterminado: False)
PORT	IN	PORT	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC. (Valor predeterminado: 0)
SIGNAL	IN	Byte	Selecciona la señal que debe activarse: (es posible seleccionar varias). Valor predeterminado: 0 • 01H = Activar RTS • 02H = Activar DTR • 04H = Activar DSR
RTS	IN	Bool	Petición de transmitir, módulo listo para transmitir el valor que debe ajustarse (True o False), valor predeterminado: False
DTR	IN	Bool	Terminal de datos disponible, módulo listo para transmitir el valor que debe ajustarse (True o False). Valor predeterminado: False
DSR	IN	Bool	Equipo de datos listo (aplicable solo a las interfaces DCE) (no se utiliza).
DONE	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición sin error
ERROR	OUT	Bool	TRUE durante una ejecución tras haberse finalizado la última petición con un error
STATUS	OUT	Word	Código de condición de ejecución (valor predeterminado: 0)

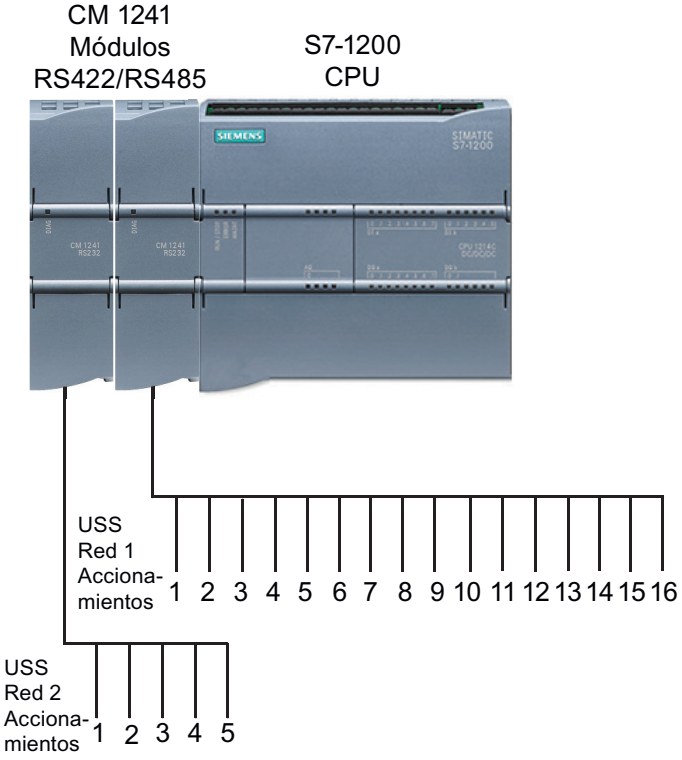
Tabla 13-124 Códigos de condición

STATUS (W#16#....)	Descripción
80F0	El CM o la CB es un RS485; no pueden activarse señales
80F1	Las señales no pueden activarse debido al control de flujo por hardware
80F2	Imposible activar DSR porque el módulo es DTE
80F3	Imposible activar DTR porque el módulo es DCE

13.7 Comunicación Legacy USS (solo CM/CB 1241)

Las instrucciones USS controlan el funcionamiento de accionamientos motorizados que soportan el protocolo interfaz serie universal (USS). Las instrucciones USS se pueden utilizar para comunicarse con varios accionamientos a través de conexiones RS485 con módulos de comunicación CM 1241 RS485 o una placa de comunicación CB 1241 RS485. Pueden instalarse hasta tres módulos CM 1241 RS422/RS485 y una placa CB 1241 RS485 en una CPU S7-1200. Cada puerto RS485 puede operar hasta 16 accionamientos.

El protocolo USS utiliza una red maestro-esclavo para una comunicación vía bus serie. El maestro utiliza un parámetro de dirección para enviar un mensaje a un esclavo seleccionado. Un esclavo jamás puede transmitir sin primero haber recibido una petición correspondiente. No es posible una transferencia directa de mensajes entre los distintos esclavos. La comunicación USS funciona en modo semidúplex. La siguiente figura de USS muestra un diagrama de red para una aplicación con accionamientos a modo de ejemplo.



Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones de comunicación USS existían con nombres diferentes y, en algunos casos, con interfaces ligeramente diferentes. Los conceptos generales son válidos para ambos juegos de instrucciones. Consulte las instrucciones Legacy USS individuales para obtener información sobre la programación.

13.7.1 Seleccionar la versión de las instrucciones USS

En STEP 7 están disponibles dos versiones de las instrucciones USS:

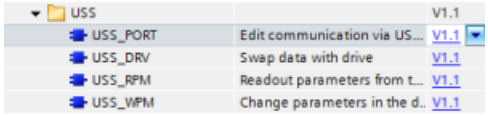
- La versión 2.0 estaba disponible inicialmente en STEP 7 Basic/Professional V13.
- La versión 2.1 y superiores está disponible en STEP 7 Basic/Professional V13 SP1 y versiones posteriores.

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

No es posible utilizar las dos versiones de las instrucciones con el mismo módulo; en cambio, dos módulos distintos sí pueden utilizar diferentes versiones de las instrucciones.



Haga clic en el icono de la Task Card del árbol de instrucciones para activar los encabezados y columnas del árbol de instrucciones.



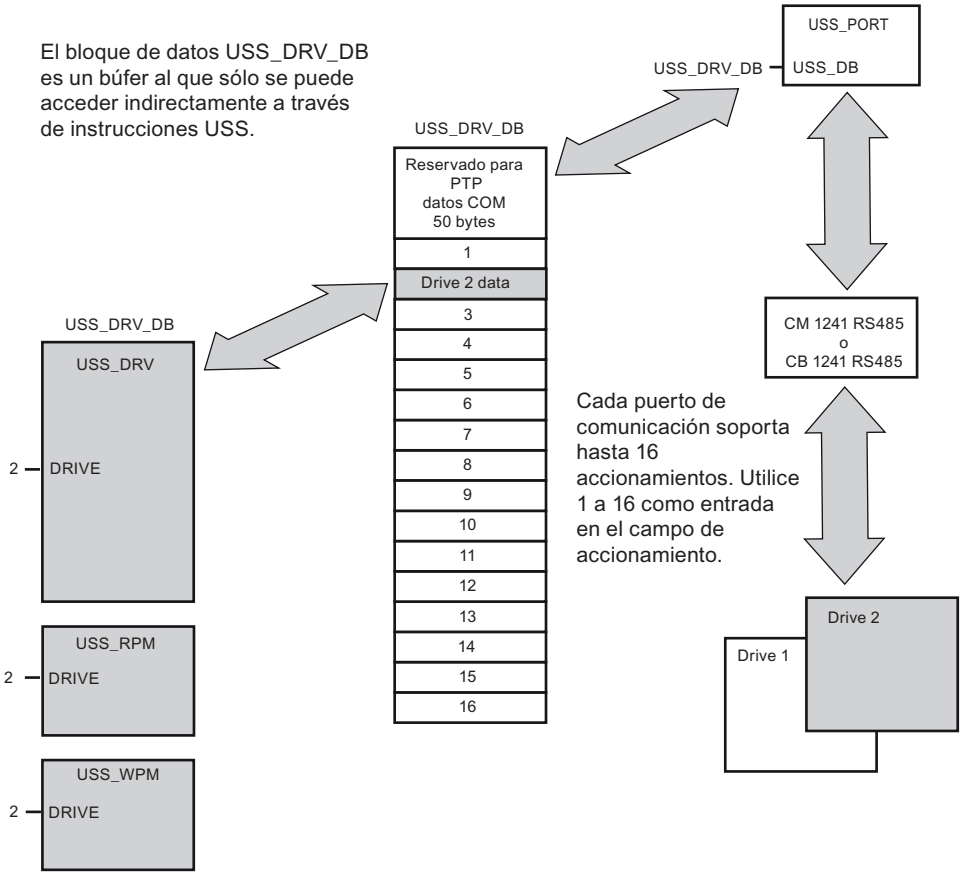
Para cambiar la versión de las instrucciones USS, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción USS en el programa, se crea en el árbol del proyecto una nueva instancia del FB o la FC dependiendo de la instrucción USS seleccionada. La nueva instancia del FB o la FC se puede ver en el árbol del proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción USS en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione una instancia del FB o la FC USS del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción USS.

13.7.2 Requisitos para utilizar el protocolo USS

Las cuatro instrucciones USS utilizan 1 FB y 3 FCs que soportan el protocolo USS. Un bloque de datos instancia (DB) USS_PORT se utiliza para cada red USS. El bloque de datos instancia USS_PORT contiene un almacenamiento temporal y búfers para todos los accionamientos en la red USS. Las instrucciones USS comparten la información en este bloque de datos.



Todos los accionamientos (máx. 16) conectados a un solo puerto RS485 pertenecen a una misma red USS. Todos los accionamientos conectados a otro puerto RS485 pertenecen a diferentes redes USS. Cada red USS se gestiona utilizando un bloque de datos unívoco. Todas las instrucciones asociadas a una red USS deben compartir dicho bloque de datos. Esto incluye todas las instrucciones USS_DRV, USS_PORT, USS_RPM y USS_WPM utilizadas para controlar todos los accionamientos de una red USS.

La instrucción USS_DRV es un bloque de función (FB). Cuando la instrucción USS_DRV se coloca en el editor de programas, es preciso indicar qué DB debe asignarse a ese FB en el diálogo "Opciones de llamada". Si se trata de la primera instrucción USS_DRV de este programa para esta red USS, es posible aceptar la asignación de DB predeterminada (o cambiar el nombre a discreción). Entonces se crea el nuevo DB. No obstante, si no es la primera instrucción USS_DRV de este canal, es preciso utilizar la lista desplegable del diálogo "Opciones de llamada" para seleccionar el nombre de DB que fue asignado previamente a esta red USS.

Las instrucciones USS_PORT, USS_RPM y USS_WPM son funciones (FCs). A estas FCs no se les asigna ningún DB cuando se colocan en el editor. En cambio, hay que asignar la referencia de DB apropiada a la entrada "USS_DB" de estas instrucciones. Haga doble clic en el campo de

parámetro. A continuación, haga clic en el símbolo de ayuda de parámetros para ver los nombres de DB disponibles.

La función USS_PORT gestiona la comunicación real entre la CPU y los accionamientos vía el puerto de comunicación punto a punto (PtP) RS485. Cada llamada a esta función gestiona una comunicación con un accionamiento. El programa debe llamar esta función lo suficientemente rápido para impedir un timeout de comunicación por parte de los módulos. Esta función puede llamarse desde un OB de ciclo de programa principal o desde cualquier OB de alarma.

Por lo general, la función USS_PORT debe llamarse desde un OB de alarma cíclica. El tiempo de ciclo del OB de alarma cíclica debe ajustarse a aprox. la mitad del intervalo mínimo de llamada (p. ej., una comunicación de 1200 baudios debe utilizar un tiempo de ciclo de 350 ms o menos).

El bloque de función USS_DRV permite al programa acceder a un accionamiento determinado en la red USS. Sus entradas y salidas representan el estado y los controles del accionamiento. Si la red comprende 16 accionamientos, el programa debe tener como mínimo 16 llamadas de USS_DRV, es decir, una para cada accionamiento. Estos bloques deben llamarse a la velocidad necesaria para controlar el funcionamiento del accionamiento.

El bloque de función USS_DRV sólo se puede llamar desde un OB de ciclo de programa principal.

⚠ PRECAUCIÓN

Factores que se deben tener en cuenta al llamar las instrucciones USS de OB

USS_DRV, USS_RPM y USS_WPM deben llamarse únicamente desde un OB de ciclo de programa principal. USS_PORT puede llamarse desde un OB cualquiera (generalmente desde un OB de alarma cíclica).

No utilice las instrucciones USS_DRV, USS_RPM o USS_WPM en un OB de prioridad más alta que la instrucción USS_PORT correspondiente. Por ejemplo, no inserte USS_PORT en el OB principal ni USS_RPM en un OB de alarma cíclica. Si no se impide la interrupción de la ejecución de USS_PORT podrían producirse errores inesperados, que podrían provocar lesiones corporales.

Las funciones USS_RPM y USS_WPM leen y escriben los parámetros operativos del accionamiento remoto. Estos parámetros controlan el funcionamiento interno del accionamiento. Estos parámetros se definen en el manual del accionamiento. El programa puede contener un número cualquiera de estas funciones. No obstante, sólo una petición de lectura o escritura puede estar activa en un accionamiento en un momento determinado. Las funciones USS_RPM y USS_WPM sólo pueden llamarse desde un OB de ciclo de programa principal.

Calcular el tiempo necesario para la comunicación con el accionamiento

La comunicación con el accionamiento es asíncrona al ciclo del S7-1200. Por lo general, pueden transcurrir varios ciclos del S7-1200 antes de que finalice una transacción de comunicación con un accionamiento.

El intervalo de USS_PORT es el tiempo necesario para una transacción con un accionamiento. La tabla siguiente muestra el intervalo de USS_PORT mínimo para cada velocidad de transferencia de comunicación. Si la función USS_PORT se llama más frecuentemente que el intervalo de USS_PORT, no se incrementará el número de transacciones. El intervalo de timeout del accionamiento es el tiempo disponible para una transacción si, debido a errores de

comunicación, se requieren 3 intentos para finalizar la transacción. De forma predeterminada, la librería del protocolo USS realiza automáticamente 2 reintentos por transacción.

Tabla 13-125 Calcular los requisitos de tiempo

Velocidad de transferencia	Intervalo mínimo calculado para la llamada de USS_PORT (milisegundos)	Intervalo de timeout por accionamiento (milisegundos)
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 Instrucciones Legacy USS

13.7.3.1 Instrucción USS_PORT (Procesar comunicación vía red USS)

Tabla 13-126 Instrucción USS_PORT

KOP / FUP	SCL	Descripción

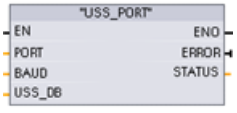
	`USS_PORT(`
	`  PORT:=_uint_in_,`
	`  BAUD:=_dint_in_,`
	`  ERROR=>_bool_out_,`
	`  STATUS=>_word_out_,`
	`  USS_DB:=_fbtref_inout_);`
		La instrucción USS_PORT gestiona la comunicación en una red USS.

Tabla 13-127 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
PORT	IN	Port	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
BAUD	IN	DInt	Velocidad de transferencia utilizada para la comunicación USS.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.

Parámetro y tipo		Tipo de datos	Descripción
ERROR	OUT	Bool	Si es TRUE (verdadero), esta salida indica que ha ocurrido un error y la salida STATUS es válida.
STATUS	OUT	Word	El valor de estado de la petición indica el resultado del ciclo o de la inicialización. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

Generalmente, el programa contiene solo una instrucción USS_PORT por cada puerto de comunicación PtP. Cada llamada de esta función gestiona una transferencia hacia o desde un accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo DB de instancia.

El programa debe ejecutar la instrucción USS_PORT con suficiente frecuencia para impedir timeouts del accionamiento. USS_PORT se llama generalmente desde un OB de alarma cíclica para impedir timeouts del accionamiento y para que las actualizaciones de datos USS más recientes estén disponibles para las llamadas de USS_DRV.

13.7.3.2 Instrucción USS_DRV (Intercambiar datos con el accionamiento)

Tabla 13-128 Instrucción USS_DRV

KOP / FUP	SCL	Descripción

Vista predeterminada

 	
Vista ampliada

 	
	`"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_);`
		La instrucción USS_DRV intercambia datos con el accionamiento creando peticiones e interpretando las respuestas del accionamiento. Para cada accionamiento debe utilizarse un bloque de función propio. No obstante, todas las funciones USS asociadas con una red USS y el puerto de comunicación PtP deben utilizar el mismo bloque de datos instancia. Es preciso crear el nombre del DB cuando se inserta la primera instrucción USS_DRV. Este DB creado al insertar la instrucción por primera vez se debe referenciar a continuación.
		STEP 7 crea el DB automáticamente al insertar la instrucción.

[1] KOP y FUP: Haga clic en el lado inferior del cuadro para ampliarlo y ver todos los parámetros. Los parámetros que aparecen atenuados son opcionales y no se requiere una asignación de parámetros.

Tabla 13-129 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
RUN	IN	Bool	Bit de arranque del accionamiento: Si es TRUE (verdadera), esta entrada habilita el accionamiento para que funcione a la velocidad predeterminada. Cuando RUN pasa a False mientras funciona un accionamiento, el motor se desacelerará hasta pararse. Este comportamiento difiere del de una caída de alimentación (OFF2) y del de un frenado del motor (OFF3).
OFF2	IN	Bool	Bit de parada eléctrica: Si es FALSE (falso), este bit hace que el accionamiento marche en inercia hasta parar sin frenar.
OFF3	IN	Bool	Bit de parada rápida: Si es FALSE (falso), este bit origina una parada rápida frenando el accionamiento, en vez de permitir que marche en inercia hasta parar.

Parámetro y tipo		Tipo de datos	Descripción
F_ACK	IN	Bool	Bit de acuse de fallo: Este bit se activa para resetear el bit de fallo en un accionamiento. El bit se activa tras haberse solucionado el fallo para indicar al accionamiento que no tiene que seguir notificando el fallo anterior.
DIR	IN	Bool	Control de sentido del accionamiento: Este bit se activa para indicar que el sentido es hacia delante (SPEED_SP positiva).
DRIVE	IN	USInt	Dirección del accionamiento: Esta entrada es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PZD_LEN	IN	USInt	Longitud de palabra: Este es el número de palabras de los datos PZD. Los valores válidos son 2, 4, 6 u 8 palabras. El valor predeterminado es 2.
SPEED_SP	IN	Real	Consigna de velocidad: Esta es la velocidad del accionamiento expresada como porcentaje de la frecuencia configurada. Un valor positivo indica el sentido hacia delante (si DIR es TRUE). El rango válido está comprendido entre 200,00 y -200,00.
CTRL3	IN	Word	Palabra de control 3: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL4	IN	Word	Palabra de control 4: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL5	IN	Word	Palabra de control 5: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL6	IN	Word	Palabra de control 6: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL7	IN	Word	Palabra de control 7: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
CTRL8	IN	Word	Palabra de control 8: Valor escrito en un parámetro configurable por el usuario en el accionamiento. El usuario debe configurarlo en el accionamiento (parámetro opcional).
NDR	OUT	Bool	Nuevos datos listos: Si es TRUE (verdadero), el bit indica que las salidas contienen datos de una petición de comunicación nueva.
ERROR	OUT	Bool	Ha ocurrido un error: Si es TRUE (verdadero), indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	Word	El valor de estado de la petición indica el resultado del ciclo. Esta no es una palabra de estado devuelta del accionamiento.
RUN_EN	OUT	Bool	Funcionamiento habilitado: Este bit indica si está funcionando el accionamiento.
D_DIR	OUT	Bool	Sentido del accionamiento: Este bit indica si el accionamiento está funcionando hacia delante.
INHIBIT	OUT	Bool	Accionamiento inhibido: Este bit indica el estado del bit de inhibición del accionamiento.
FAULT	OUT	Bool	Fallo del accionamiento: Este bit indica que el accionamiento ha registrado un fallo. El usuario debe corregir el problema y activar el bit F_ACK para borrar este bit si está activado.

Parámetro y tipo		Tipo de datos	Descripción
SPEED	OUT	Real	Velocidad actual del accionamiento (valor escalado de la palabra de estado 2 del accionamiento) Valor de velocidad del accionamiento expresado como porcentaje de la velocidad configurada.
STATUS1	OUT	Word	Palabra de estado 1 del accionamiento: Este valor contiene bits de estado fijos de un accionamiento.
STATUS3	OUT	Word	Palabra de estado 3 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS4	OUT	Word	Palabra de estado 4 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS5	OUT	Word	Palabra de estado 5 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS6	OUT	Word	Palabra de estado 6 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS7	OUT	Word	Palabra de estado 7 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.
STATUS8	OUT	Word	Palabra de estado 8 del accionamiento: Este valor contiene una palabra de estado configurable por el usuario en el accionamiento.

Cuando la instrucción USS_DRV se ejecuta por primera vez, el accionamiento que indica la dirección USS (parámetro DRIVE) se inicializa en el DB de instancia. Después de esta inicialización, las ejecuciones siguientes de USS_PORT pueden iniciar la comunicación con el accionamiento en este número de accionamiento.

Si se modifica el número del accionamiento, la CPU debe cambiar de STOP a RUN con objeto de inicializar el DB de instancia. Los parámetros de entrada se configuran en el búfer de mensajes USS TX y las salidas se leen de un búfer de respuesta válido "anterior" (si existe). Durante la ejecución de USS_DRV no se transmiten datos. Los accionamientos se comunican cuando USS_PORT se ha ejecutado. USS_DRV configura únicamente los mensajes que deben enviarse e interpreta los datos que puedan haberse recibido de una petición anterior.

El sentido de rotación del accionamiento se puede controlar utilizando la entrada DIR (Bool) o el signo (positivo o negativo) con la entrada SPEED_SP (Real). La tabla siguiente indica cómo interactúan estas entradas para determinar el sentido del accionamiento, suponiendo que el motor está cableado para la rotación adelante.

Tabla 13-130 Interacción de los parámetros SPEED_SP y DIR

SPEED_SP	DIR	Sentido de rotación del accionamiento
Valor > 0	0	Atrás
Valor > 0	1	Adelante
Valor < 0	0	Adelante
Valor < 0	1	Atrás

13.7.3.3 Instrucción USS_RPM (Leer parámetros del accionamiento)

Tabla 13-131 Instrucción USS_RPM

KOP / FUP	SCL	Descripción

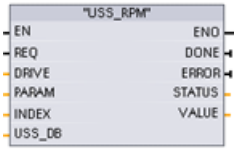
	`USS_RPM(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`VALUE=>_variant_out_,`
	`USS_DB:=_fbtref_inout_);`
		La instrucción USS_RPM lee un parámetro del accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo bloque de datos. USS_RPM debe llamarse desde un OB de ciclo de programa principal.

Tabla 13-132 Tipos de datos para los parámetros

Tipo de parámetro		Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), REQ indica que se desea una nueva petición de lectura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: DRIVE es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.
PARAM	IN	UInt	Número de parámetro: PARAM designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. En algunos accionamientos, el byte más significativo puede acceder a valores PARAM superiores a 2047. Encontrará más información sobre cómo acceder a un rango ampliado en el manual del accionamiento.
INDEX	IN	UInt	Índice de parámetro: INDEX designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Valor del parámetro que se ha leído y que es válido solo si el bit DONE es TRUE (verdadero).
DONE[1]	OUT	Bool	Si es TRUE (verdadero), indica que la salida VALUE contiene el valor del parámetro de lectura solicitado anteriormente. Este bit se activa cuando USS_DRV detecta los datos de respuesta de lectura del accionamiento. Este bit se desactiva en los casos siguientes: los datos de respuesta se solicitan mediante otra consulta USS_RPM o bien en la segunda de las dos llamadas siguientes de USS_DRV.

Tipo de parámetro		Tipo de datos	Descripción
ERROR	OUT	Bool	Ha ocurrido un error: Si es TRUE (verdadero), ERROR indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	Word	STATUS indica el resultado de la petición de lectura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

[1] El bit DONE indica que se han leído datos válidos del accionamiento de motor referenciado y que se han transmitido a la CPU. No indica que la librería USS sea capaz de leer inmediatamente otros parámetros. Hay que enviar una petición PKW vacía al accionamiento del motor, la cual debe ser acusada por la instrucción antes de que el canal de parámetros para el accionamiento en cuestión vuelva a estar disponible. Si se llama inmediatamente una FC USS_RPM o USS_WPM para el accionamiento de motor especificado se producirá un error 0x818A.

13.7.3.4 Instrucción USS_WPM (Modificar parámetros en el accionamiento)

Nota

Operaciones de escritura EEPROM (para la EEPROM dentro de un accionamiento USS)

No utilice excesivamente la operación de escritura permanente en EEPROM. Minimice el número de operaciones de escritura en EEPROM para prolongar la vida útil de la EEPROM.

Tabla 13-133 Instrucción USS_WPM

KOP / FUP	SCL	Descripción

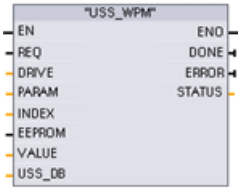
	`USS_WPM(REQ:=_bool_in_,`
	`DRIVE:=_usint_in_,`
	`PARAM:=_uint_in_,`
	`INDEX:=_uint_in_,`
	`EEPROM:=_bool_in_,`
	`VALUE:=_variant_in_,`
	`DONE=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`USS_DB:=_fbtref_inout_);`
		La instrucción USS_WPM modifica un parámetro en el accionamiento. Todas las funciones USS asociadas a una red USS y a un puerto de comunicación PtP deben utilizar el mismo bloque de datos.
		USS_WPM debe llamarse desde un OB de ciclo de programa principal.

Tabla 13-134 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Enviar petición: Si es TRUE (verdadero), REQ indica que se desea una nueva petición de escritura. Esto se ignora si la petición para este parámetro ya está pendiente.
DRIVE	IN	USInt	Dirección del accionamiento: DRIVE es la dirección del accionamiento USS. El rango válido está comprendido entre el accionamiento 1 y el 16.

Parámetro y tipo		Tipo de datos	Descripción
PARAM	IN	UInt	Número de parámetro: PARAM designa el parámetro del accionamiento que se escribe. El rango de este parámetro está comprendido entre 0 y 2047. En algunos accionamientos, el byte más significativo puede acceder a valores PARAM superiores a 2047. Encontrará más información sobre cómo acceder a un rango ampliado en el manual del accionamiento.
INDEX	IN	UInt	Índice de parámetro: INDEX designa el índice de parámetro del accionamiento que se escribirá. Valor de 16 bits en el que el byte menos significativo es el valor de índice real en un rango de 0 a 255. El accionamiento también puede utilizar el byte más significativo. Este byte es específico del accionamiento. Para más información, consulte el manual del accionamiento.
EEPROM	IN	Bool	Guardar en la EEPROM del accionamiento: Si es TRUE (verdadero), una operación de escritura en los parámetros del accionamiento se guardará en la EEPROM de este. Si es FALSE (falso), la operación de escritura será temporal por lo que no se conservará tras desconectar y volver a conectar la alimentación del accionamiento.
VALUE	IN	Word, Int, UInt, DWord, DInt, UDInt, Real	Valor del parámetro en el que se debe escribir. Debe ser válido en la transición de REQ.
USS_DB	INOUT	USS_BASE	El nombre del DB de instancia que se crea e inicializa cuando se inserta una instrucción USS_DRV en el programa.
DONE[1]	OUT	Bool	Si es TRUE (verdadero), DONE indica que la entrada VALUE se ha escrito en el accionamiento. Este bit se activa cuando USS_DRV detecta los datos de respuesta de escritura del accionamiento. El bit se desactiva cuando los datos de respuesta se solicitan mediante otra consulta USS_WPM o bien en la segunda de las dos llamadas siguientes de USS_DRV.
ERROR	OUT	Bool	Si es TRUE (verdadero), ERROR indica que ha ocurrido un error y la salida STATUS es válida. Todas las demás salidas se ponen a cero cuando ocurre un error. Los errores de comunicación se notifican solo en las salidas ERROR y STATUS de la instrucción USS_PORT.
STATUS	OUT	Word	STATUS indica el resultado de la petición de escritura. Encontrará más información acerca de algunos códigos de estado en la variable "USS_Extended_Error".

[1] El bit DONE indica que se han leído datos válidos del accionamiento de motor referenciado y que se han transmitido a la CPU. No indica que la librería USS sea capaz de leer inmediatamente otros parámetros. Hay que enviar una petición PKW vacía al accionamiento del motor, la cual debe ser acusada por la instrucción antes de que el canal de parámetros para el accionamiento en cuestión vuelva a estar disponible. Si se llama inmediatamente una FC USS_RPM o USS_WPM para el accionamiento de motor especificado se producirá un error 0x818A.

13.7.4 Códigos de estado Legacy USS

La salida STATUS de las funciones USS devuelve los códigos de estado de estas funciones.

Tabla 13-135 Códigos STATUS [1]

STATUS (W#16#....)	Descripción
0000	No hay error
8180	La longitud de la respuesta del accionamiento no concuerda con los caracteres recibidos del accionamiento. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8181	El parámetro VALUE no es un tipo de datos Word, Real o DWord.
8182	El usuario ha introducido un parámetro en formato Word y se ha recibido una respuesta en formato DWord o Real del accionamiento.
8183	El usuario ha introducido un parámetro en formato DWord o Real y se ha recibido una respuesta en formato Word del accionamiento.
8184	El telegrama de respuesta del accionamiento tiene una suma de verificación incorrecta. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8185	Dirección del accionamiento no válida (rango de direcciones válidas: de 1 a 16)
8186	La consigna de velocidad está fuera del rango válido (rango de SP de velocidad válido: -200% a 200%).
8187	Un número de accionamiento incorrecto ha respondido a la petición enviada. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
8188	Se ha indicado una longitud de palabra PZD no permitida (rango válido = 2, 4, 6 u 8 palabras)
8189	Se ha indicado una velocidad de transferencia no permitida.
818A	Otra petición para este accionamiento está utilizando el canal de petición de parámetros.
818B	El accionamiento no ha respondido a las peticiones ni reintentos. El número de accionamiento en el que ha ocurrido el error se devuelve en la variable "USS_Extended_Error". La descripción de los errores avanzados aparece a continuación de esta tabla.
818C	El accionamiento ha devuelto un error avanzado relativo a la petición de parámetros. La descripción de los errores avanzados aparece a continuación de esta tabla.
818D	El accionamiento ha devuelto un error de acceso no permitido relativo a la petición de parámetros. Para más información sobre la limitación de acceso a los parámetros, consulte el manual del accionamiento.
818E	El accionamiento no se ha inicializado. Este código de error se devuelve a USS_RPM o USS_WPM si USS_DRV no se ha llamado por lo menos una vez para este accionamiento. De esta manera se impide que la inicialización de USS_DRV en el primer ciclo sobrescriba una petición pendiente de lectura o escritura de los parámetros, puesto que inicializa el accionamiento como entrada nueva. Para corregir este error, llame USS_DRV para este número de accionamiento.
80Ax-80Fx	Errores específicos que devuelven los FBs de comunicación PtP llamados por la librería USS - La librería USS no modifica estos códigos de error que se definen en las descripciones de la instrucción PtP.

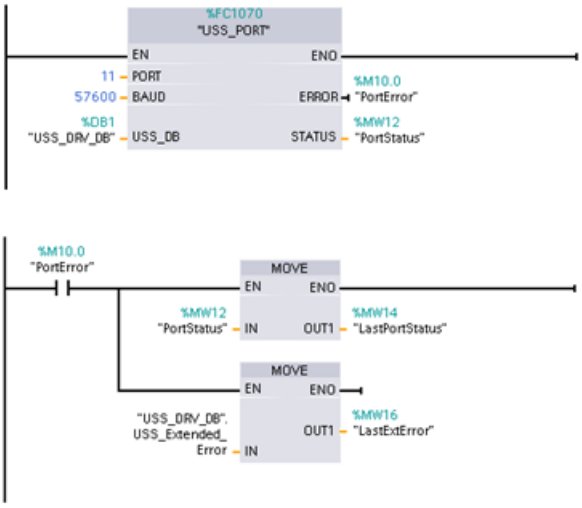
[1] Además de los errores de la instrucción USS indicados arriba, las instrucciones de comunicaciones PtP subyacentes pueden devolver errores.

Para muchos códigos STATUS se suministra información adicional en la variable "USS_Extended_Error" del DB de instancia USS_DRV. Para los códigos STATUS hexadecimales 8180, 8184, 8187 y 818B, USS_Extended_Error contiene el número del accionamiento en que se ha presentado el error de comunicación. Para el código STATUS hexadecimal 818C,

USS_Extended_Error contiene un código de error que devuelve el accionamiento cuando se utiliza una instrucción USS_RPM o USS_WPM.

Ejemplo: notificación de errores de comunicación

Los errores de comunicación (STATUS = 16#818B) se notifican solo en la instrucción USS_PORT y no en la instrucción USS_DRV. Si, por ejemplo, la red no ha sido finalizada correctamente, un accionamiento puede pasar a RUN, no obstante la instrucción USS_DRV mostrará el valor 0 para todos los parámetros de salida. En este caso, el error de comunicación solo se puede detectar en la instrucción USS_PORT. Debido a que este error solo es visible durante un ciclo, hay que agregar algunos elementos lógicos de detección, como se indica en el ejemplo a continuación. En este ejemplo, si el bit de error de la instrucción USS_PORT es TRUE, entonces los valores STATUS y USS_Extended_Error se guardan en el área de marcas. El número de accionamiento se deposita en la variable USS_Extended_Error si el código STATUS tiene el valor hexadecimal 8180, 8184, 8187 u 818B.



Segmento 1 El estado del puerto "PortStatus" y los valores de los códigos de error avanzados de "USS_DRV_DB".USS_Extended_Error solo son válidos durante un ciclo del programa. Los valores deben capturarse para un procesamiento posterior.

Segmento 2 El contacto "PortError" inicia el almacenamiento del valor "PortStatus" en "LastPortStatus" y del valor"USS_DRV_DB".USS_Extended_Error en "LastExtError".

Acceso de lectura y escritura a los parámetros internos de un accionamiento

Los accionamientos USS soportan accesos de lectura y escritura a los parámetros internos de los accionamientos. Esta función permite controlar y configurar el accionamiento de forma remota. Las operaciones de acceso a los parámetros del accionamiento pueden fallar debido a errores tales como valores fuera de rango o peticiones no permitidas del modo actual del accionamiento. El accionamiento genera un código de error cuyo valor se devuelve en la variable "USS_Extended_Error". Este valor del código de error es válido únicamente para la última ejecución de una instrucción USS_RPM o USS_WPM. El código de error del accionamiento se deposita en la variable USS_Extended_Error si el STATUS code tiene el valor hexadecimal 818C. El valor del código de error de "USS_Extended_Error" depende del modelo de accionamiento. Los códigos de error avanzados para las operaciones de lectura y escritura de parámetros se describen en el manual del accionamiento.

13.7.5 Requisitos generales para la configuración del accionamiento Legacy USS

Los requisitos generales para la configuración del accionamiento Legacy USS constan de los puntos siguientes:

- Los accionamientos deben configurarse para utilizar 4 palabras PKW.
- Los accionamientos pueden configurarse para 2, 4, 6 u 8 palabras PZD.
- El número de palabras PZD en el accionamiento debe concordar con la entrada PZD_LEN de la instrucción USS_DRV para ese accionamiento.
- La velocidad de transferencia en todos los accionamientos debe concordar con la entrada BAUD de la instrucción USS_PORT .
- El accionamiento debe configurarse para control remoto.
- Para el accionamiento debe configurarse una consigna de frecuencia a USS en el puerto COM.
- La dirección del accionamiento debe configurarse en un rango de 1 a 16 y debe concordar con la entrada DRIVE del bloque USS_DRV para ese accionamiento.
- El control de sentido del accionamiento debe configurarse para que utilice la polaridad de la consigna del accionamiento.
- La red RS485 se debe terminar correctamente.

La conexión e instalación general del accionamiento USS es la misma para instrucción USS (V4.1) e instrucciones Legacy USS (V4.0 y anteriores). Consulte el Ejemplo: conexión e instalación general del accionamiento USS (Página 1010) para más información.

13.8 Comunicación Legacy Modbus TCP

13.8.1 Vista general

Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones de comunicación Modbus TCP existían con nombres diferentes y, en algunos casos, con interfaces ligeramente diferentes. Los conceptos generales son válidos para ambos juegos de instrucciones. Consulte las instrucciones Legacy Modbus TCP individuales para obtener información sobre la programación.

13.8.2 Seleccionar la versión de las instrucciones Modbus TCP

En STEP 7 hay disponibles las versiones siguientes de las instrucciones Modbus TCP:

- Versión antigua 2.1: compatible con todas las versiones de CPU y CM
- Versión antigua 3.1: compatible con todas las versiones de CPU y CM
- Versión 4.2: compatible con CPU V4.0 y superiores y con CM V2.1 y superiores
- Versión 5.1: compatible con CPU V4.2 y superiores y con CM V2.1 y superiores
- Versión antigua 6.0: compatible con CPU V4.2 y superiores y con CM V2.1 y superiores

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

En la Task Card Instrucciones, visualice las instrucciones MODBUS TCP en "Otros" del grupo "Comunicación".

Para cambiar la versión de las instrucciones Modbus TCP, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.



Cuando se utiliza el árbol de instrucciones para insertar una instrucción Modbus TCP en el programa, se crea una nueva instancia del FB en el árbol del proyecto. El nuevo FB instancia se puede ver en el árbol de proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción Modbus TCP en un programa, es necesario examinar las propiedades del árbol del proyecto y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione una instancia del FB Modbus TCP del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción Modbus TCP.

13.8.3 Instrucciones Legacy Modbus TCP

13.8.3.1 MB_CLIENT (comunicar como cliente TCP Modbus vía PROFINET)

Tabla 13-136 Instrucción MB_CLIENT

KOP / FUP	SCL	Descripción

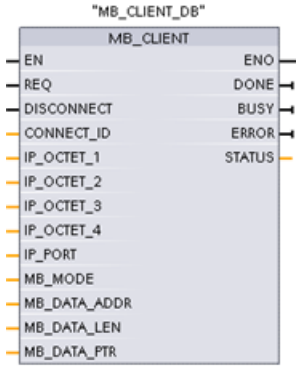
	`"MB_CLIENT_DB"(`
	`REQ:=_bool_in_,`
	`DISCONNECT:=_bool_in_,`
	`CONNECT_ID=_uint_in_,`
	`IP_OCTET_1:=_byte_in_,`
	`IP_OCTET_2:=_byte_in_,`
	`IP_OCTET_3:=_byte_in_,`
	`IP_OCTET_4:=_byte_in_,`
	`IP_PORT:=_uint_in_,`
	`MB_MODE:=_usint_in_,`
	`MB_DATA_ADDR:=_udint_in_,`
	`MB_DATA_LEN:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DATA_PTR:=_variant_inout_);`
		MB_CLIENT se comunica como cliente Modbus TCP a través del conector PROFINET de la CPU S7-1200. No se necesitan módulos hardware de comunicación adicionales.
		MB_CLIENT puede establecer una conexión cliente-servidor, enviar una petición de función Modbus, recibir una respuesta y controlar la desconexión de un servidor Modbus TCP.

Tabla 13-137 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	In	Bool	FALSE = Ninguna petición de comunicación Modbus TRUE = Petición de comunicación con un servidor Modbus TCP
DISCONNECT	IN	Bool	El parámetro DISCONNECT permite que su programa controle la conexión y desconexión con un dispositivo servidor Modbus. Si DISCONNECT = 0 y no hay ninguna conexión, entonces MB_CLIENT intenta establecer una conexión con la dirección IP y el número de puerto asignados. Si DISCONNECT = 1 y existe una conexión, entonces se intenta una operación de desconexión. Siempre que esta entrada esté habilitada, no se intentará ninguna otra operación.
CONNECT_ID	IN	UInt	El parámetro CONNECT_ID debe identificar unívocamente cada conexión dentro del PLC. Cada instancia única de la instrucción MB_CLIENT o MB_SERVER debe contener un parámetro CONNECT_ID único.
IP_OCTET_1	IN	USInt	Dirección IP del servidor Modbus TCP: octeto 1 Parte de 8 bits de la dirección IP IPv4 de 32 bits del servidor Modbus TCP al que se conectará el cliente y con el que se comunicará mediante el protocolo Modbus TCP.
IP_OCTET_2	IN	USInt	Dirección IP del servidor Modbus TCP: octeto 2
IP_OCTET_3	IN	USInt	Dirección IP del servidor Modbus TCP: octeto 3
IP_OCTET_4	IN	USInt	Dirección IP del servidor Modbus TCP: octeto 4
IP_PORT	IN	UInt	Valor predeterminado = 502: el número de puerto IP del servidor al que se intentará conectar el cliente y con el que se acabará comunicando mediante el protocolo TCP/IP.
MB_MODE	IN	USInt	Selección de modo: asigna el tipo de petición (lectura, escritura o diagnóstico). Consulte los detalles en la tabla de funciones Modbus que aparece más abajo.
MB_DATA_ADDR	IN	UDInt	Dirección inicial Modbus: asigna la dirección inicial de los datos a los que accederá MB_CLIENT. Las direcciones válidas se indican en la tabla de funciones Modbus que aparece más abajo.
MB_DATA_LEN	IN	UInt	Longitud de datos Modbus: asigna el número de bits o palabras a las que debe accederse en esta petición. Las longitudes válidas se indican en la tabla de funciones Modbus que aparece más abajo.
MB_DATA_PTR	IN_OUT	Variant	Puntero al registro de datos Modbus: El registro almacena temporalmente datos que van a un servidor Modbus o vienen de este. El puntero debe asignar una dirección de memoria M o un DB global no optimizado.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación MB_CLIENT en curso • 1 - Operación MB_CLIENT en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la ejecución de MB_CLIENT con un error. El valor del código de error en el parámetro STATUS es válido solo durante el ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

Parámetro REQ

FALSE = Ninguna petición de comunicación Modbus
TRUE = Petición de comunicación con un servidor Modbus TCP

Si ninguna instancia de MB_CLIENT está activa y el parámetro DISCONNECT=0, se iniciará una nueva petición Modbus cuando REQ=1. Si la conexión no está todavía establecida, se establecerá una nueva.

Si la misma instancia de MB_CLIENT se ejecuta nuevamente con DISCONNECT=0 y REQ=1, antes de que finalice la petición actual no se realizarán más transmisiones Modbus. No obstante, en cuanto finalice la petición actual, se podrá procesar otra petición si MB_CLIENT se ejecuta con REQ=1.

Cuando finalice la petición de comunicación MB_CLIENT actual, el bit DONE será TRUE durante un ciclo. El bit DONE se puede usar como puerta temporal para secuenciar varias peticiones MB_CLIENT.

Nota

Coherencia de los datos de entrada durante el procesamiento de MB_CLIENT

Una vez que un cliente Modbus inicia una operación Modbus, todos los estados de entrada se guardan internamente y se comparan en cada llamada sucesiva. La comparación se utiliza para determinar si una llamada en particular originó la petición de cliente activa. Se puede realizar más de una llamada de MB_CLIENT con un DB de instancia común.

Por lo tanto, es importante que las entradas no se cambien durante el tiempo en el que se procesa activamente una operación MB_CLIENT . Si no se observa esta regla, un MB_CLIENT no puede determinar que es la instancia activa.

Los parámetros MB_MODE y MB_DATA_ADDR seleccionan la función de comunicación Modbus

MB_DATA_ADDR asigna la dirección Modbus inicial de los datos a los que se accederá. La instrucción MB_CLIENT utiliza una entrada MB_MODE en vez de una entrada de código de función.

La combinación de los valores MB_MODE y MB_DATA_ADDR determina el código de función utilizado en el mensaje Modbus real. La tabla siguiente muestra la correspondencia entre el parámetro MB_MODE, la función Modbus y el rango de direcciones Modbus.

Tabla 13-138 Funciones Modbus

MB_MODE	Función Modbus	Longitud de datos	Operación y datos	MB_DATA_ADDR
0	01	De 1 a 2000	Leer bits de salida: De 1 a 2000 bits por petición	De 1 a 9999
0	02	De 1 a 2000	Leer bits de entrada: De 1 a 2000 bits por petición	De 10001 a 19999
0	03	De 1 a 125	Leer registros de retención: De 1 a 125 palabras por petición	De 40001 a 49999 o De 400001 a 465535
0	04	De 1 a 125	Leer palabras de entrada: De 1 a 125 palabras por petición	De 30001 a 39999
1	05	1	Escribir un bit de salida: Un bit por petición	De 1 a 9999
1	06	1	Escribir un registro de retención: 1 palabra por petición	De 40001 a 49999 o De 400001 a 465535

MB_MODE	Función Modbus	Longitud de datos	Operación y datos	MB_DATA_ADDR
1	15	De 2 a 1968	Escribir varios bits de salida: De 2 a 1968 bits por petición	De 1 a 9999
1	16	De 2 a 123	Escribir varios registros de retención: de 2 a 123 palabras por petición	De 40001 a 49999 o De 400001 a 465535
2	15	De 1 a 1968	Escribir uno o más bits de salida: de 1 a 1968 bits por petición	De 1 a 9999
2	16	De 1 a 123	Escribir uno o más registros de retención: de 1 a 123 palabras por petición	De 40001 a 49999 o De 400001 a 465535
11	11	0	Leer la palabra de estado de la comunicación del servidor y el contador de eventos. Esta palabra de estado indica ocupado (0 – no ocupado, 0xFFFF - ocupado). El contador de eventos se incrementa cada vez que se finaliza un mensaje correctamente. En esta función se ignoran los parámetros MB_DATA_ADDR y MB_DATA_LEN de MB_CLIENT.	
80	08	1	Comprobar el estado del servidor utilizando el código de diagnóstico de datos 0x0000 (test de Loopback – el servidor envía de regreso la petición) 1 palabra por petición	
81	08	1	Inicializar el contador de eventos del servidor utilizando el código de diagnóstico de datos 0x000A 1 palabra por petición	
De 3 a 10, de 12 a 79, de 82 a 255			Reservado	

Nota

MB_DATA_PTR asigna un búfer para guardar datos leídos/escritos hasta/desde un servidor Modbus TCP

El búfer de datos puede ubicarse en una dirección de memoria M o un DB global no optimizado.

Para un búfer en la memoria M utilice el formato de puntero Any estándar. Este está en el formato P#"dirección de bit" "tipo de datos" "longitud"; por ejemplo: P#M1000.0 WORD 500.

MB_DATA_PTR asigna un búfer de comunicación

- Funciones de comunicación de MB_CLIENT:
  - Leer y escribir datos de 1 bit de las direcciones del servidor Modbus (00001 a 09999)
  - Leer datos de 1 bit de las direcciones del servidor Modbus (de 10001 a 19999)
  - Leer datos de palabras de 16 bits de las direcciones del servidor Modbus (de 30001 a 39999) y (de 40001 a 49999)
  - Escribir datos de palabras de 16 bit de las direcciones del servidor Modbus (40001 a 49999)
- Los datos de tamaño de palabra o de bit se transfieren a/desde el búfer de memoria M o el DB asignado por MB_DATA_PTR.
- Si MB_DATA_PTR asigna un DB como búfer, el usuario debe asignar tipos de datos a todos los elementos de datos del DB.
  - El tipo de datos de 1 bit Bool representa una dirección de bit Modbus
  - Los tipos de datos de una palabra de 16 bits como WORD, UInt e Int representan una dirección de palabra Modbus
  - Los tipos de datos de palabra doble de 32 bits como DWORD, DInt y Real representan dos direcciones de palabra Modbus
- MB_DATA_PTR puede asignar elementos de DB complejos como:
  - Matrices estándar
  - Estructuras con nombre en las que cada elemento es único
  - Estructuras complejas con nombre, en las que cada elemento tiene un nombre unívoco y un tipo de datos de 16 o 32 bits
- No es necesario que las áreas de datos de MB_DATA_PTR se encuentren en el mismo bloque de datos global (o área de memoria M). Se puede asignar un bloque de datos para lecturas Modbus, otro para escrituras Modbus o un bloque de datos para cada estación MB_CLIENT.

Varias conexiones de cliente

Un cliente Modbus TCP puede admitir conexiones concurrentes hasta el número máximo de conexiones para Open User Communications permitidas por el PLC. El número total de conexiones en un PLC dado, clientes y servidores Modbus TCP incluidos, no debe sobrepasar el número máximo de conexiones para Open User Communications admitidas (Página 591). Las conexiones Modbus TCP pueden compartirse entre conexiones de cliente o de servidor.

Cada conexión de cliente debe cumplir estas reglas:

- Cada conexión MB_CLIENT debe usar un DB de instancia distinto
- Cada conexión MB_CLIENT debe especificar una dirección IP de servidor única
- Cada conexión MB_CLIENT debe especificar una ID de conexión única
- Según la configuración del servidor pueden ser necesarios, o no, números únicos de puerto IP

La ID de conexión debe ser unívoca para cada conexión individual. Es decir, solo debe usarse una ID de conexión única con cada DB de instancia individual. Resumiendo, el DB de instancia y la ID de conexión están emparejadas y deben ser únicas para cada conexión.

Tabla 13-139 Variables estáticas del bloque de datos de instancia MB_CLIENT accesibles por el usuario

Variable	Tipo de datos	Valor predeterminado	Descripción
Blocked_Proc_Timeout	Real	3,0	Tiempo (en segundos) que hay que esperar a una instancia cliente Modbus bloqueada antes de eliminar esta instancia como ACTIVA. Esto puede ocurrir, por ejemplo, si se ha lanzado una petición de cliente y la aplicación deja de ejecutar la función de cliente antes de haber finalizado por completo la petición. El límite máximo de S7-1200 es 55 segundos.
MB_Unit_ID	Word	255	Identificador de unidad Modbus: Un servidor Modbus TCP se direcciona a través de su dirección IP. Por lo tanto, el parámetro MB_UNIT_ID no se usa para el direccionamiento Modbus TCP. El parámetro MB_UNIT_ID corresponde a la dirección de esclavo en el protocolo Modbus RTU. Si un servidor Modbus TCP se usa como pasarela a un protocolo Modbus RTU, MB_UNIT_ID puede usarse para identificar el dispositivo esclavo conectado en la red serie. MB_UNIT_ID se usaría para reenviar la solicitud a la dirección del esclavo Modbus RTU correcta. Algunos dispositivos Modbus TCP pueden necesitar que el parámetro MB_UNIT_ID se inicialice dentro de un rango restringido de valores.
RCV_TIMEOUT	Real	2,0	Tiempo en segundos que el MB_CLIENT espera a que el servidor responda a una petición.
Conectado	Bool	0	Indica si la conexión al servidor asignado está conectada o desconectada: 1=conectada, 0=desconectada

Tabla 13-140 Errores de protocolo MB_CLIENT

STATUS (W#16#)	Código de respuesta al cliente Modbus (B#16#)	Errores de protocolo Modbus
8381	01	Código de función no soportado
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o acceso fuera de los límites del área de direcciones de MB_HOLD_REG
8384	03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)

Tabla 13-141 Códigos de condición de ejecución de MB_CLIENT [1]

STATUS (W#16#)	Errores de parámetros de MB_CLIENT
7001	MB_CLIENT está esperando a que un servidor Modbus responda a una petición de conexión o de desconexión en el puerto TCP asignado. Esto solo se notifica en la primera ejecución de una operación de conexión o de desconexión.
7002	MB_CLIENT está esperando a que un servidor Modbus responda a una petición de conexión o de desconexión en el puerto TCP asignado. Esto se notificará en las ejecuciones sucesivas, mientras se espera a que finalice una operación de conexión o de desconexión.
7003	Una operación de desconexión ha finalizado correctamente (válido solo para un ciclo del PLC).
80C8	El servidor no ha respondido en el tiempo asignado. MB_CLIENT debe recibir una respuesta mediante la ID de transacción que se transmitió originalmente dentro del tiempo asignado o se devuelve este error. Compruebe la conexión al dispositivo servidor Modbus. Este error solo se notifica una vez se han ejecutado los intentos configurados (si procede).
8188	El valor de modo no es válido
8189	Valor de dirección de datos no válido
818A	Valor de longitud de datos no válido
818B	Puntero al área DATA_PTR no válido. Puede ser la combinación de MB_DATA_ADDRESS + MB_DATA_LEN.
818C	Puntero a un área DATA_PTR optimizada (debe ser un área de memoria M o DB no optimizada)
8200	El puerto está ocupado porque está procesando una petición Modbus existente.
8380	La trama Modbus recibida está corrupta o no se han recibido suficientes bytes.
8387	El parámetro de ID de conexión asignado es diferente de la ID utilizada para peticiones anteriores. Solo se puede usar una ID de conexión en cada DB de instancia de MB_CLIENT. Esto también se usa como error interno si la ID de protocolo de Modbus TCP recibida de un servidor no es 0.
8388	Un servidor Modbus ha devuelto una cantidad de datos diferente de la solicitada. Solo se aplica a las funciones Modbus 15 y 16.

[1] Además de los errores de MB_CLIENT indicados, las instrucciones de comunicación del bloque T subyacente (TCON, TDISCON, TSEND y TRCV (Página 652)) pueden devolver errores.

13.8.3.2 MB_SERVER (comunicar como servidor TCP Modbus vía PROFINET)

La instrucción "MB_SERVER" se comunica como servidor Modbus TCP a través del conector PROFINET de la CPU S7-1200. La instrucción "MB_SERVER" procesa peticiones de conexión de un cliente Modbus TCP, recibe y procesa peticiones Modbus y envía respuestas.

Para usar la instrucción no se requiere un módulo de hardware adicional.

ATENCIÓN
Indicaciones de seguridad Tenga en cuenta que cada cliente de la red recibe acceso de lectura y escritura a las entradas y salidas de la memoria imagen de proceso y al bloque de datos o área de memoria de bits definidos en el registro de retención Modbus. La opción está disponible para restringir el acceso a una dirección IP y para prevenir operaciones de lectura y escritura no autorizadas. No obstante, tenga en cuenta que la dirección compartida también puede emplearse para un acceso no autorizado.

Tabla 13-142 Instrucción MB_SERVER

KOP / FUP	SCL	Descripción

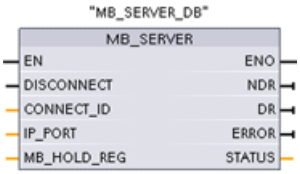
	`"MB_SERVER_DB"(`
	`DISCONNECT:=_bool_in_,`
	`CONNECT_ID:=_uint_in_,`
	`IP_PORT:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		MB_SERVER se comunica como servidor Modbus TCP a través del conector PROFINET de la CPU S7-1200. No se necesitan módulos hardware de comunicación adicionales.
		MB_SERVER puede aceptar una petición para conectarse a un cliente Modbus TCP, recibir una petición de función Modbus y enviar un mensaje de respuesta

Tabla 13-143 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
DISCONNECT	IN	Bool	MB_SERVER intenta establecer una conexión "pasiva" con un dispositivo interlocutor. Es decir, el servidor escucha pasivamente una petición de conexión TCP desde cualquier dirección IP que lo solicite. Si DISCONNECT = 0 y no existe una conexión, puede iniciarse una conexión pasiva. Si DISCONNECT = 1 y existe una conexión, se inicia una operación de desconexión. Esto permite que su programa controle cuándo se acepta una conexión. Siempre que esta entrada esté habilitada, no se intentará ninguna otra operación.
CONNECT_ID	IN	UInt	CONNECT_ID identifica unívocamente cada conexión dentro del PLC. Cada instancia única de la instrucción MB_CLIENT o MB_SERVER debe contener un parámetro CONNECT_ID único.
IP_PORT	IN	UInt	Valor predeterminado = 502: El número de puerto IP que identifica el puerto IP que se vigilará para peticiones de conexión de un cliente Modbus.
MB_HOLD_REG	IN_OUT	Variant	Puntero al registro de retención Modbus de MB_SERVER: El registro de retención debe ser una dirección de memoria M o bien un DB global no optimizado. Esta área de memoria se utiliza para retener los valores a los que un cliente Modbus puede acceder con las funciones de registro Modbus 3 (lectura), 6 (escritura) y 16 (escritura).
NDR	OUT	Bool	Nuevos datos listos: 0 = No hay datos nuevos, 1= Indica que el cliente Modbus ha escrito datos nuevos
DR	OUT	Bool	Lectura de datos: 0 = No se han leído datos, 1= Indica que el cliente Modbus ha leído datos.
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la ejecución de MB_SERVER con un error. El valor del código de error en el parámetro STATUS es válido solo durante el ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

MB_SERVER permite que los códigos de funciones Modbus entrantes (1, 2, 4, 5 y 15) puedan leer y escribir bits y palabras directamente en la memoria imagen de proceso de las entradas y salidas de la CPU S7-1200. Para los códigos de función de transferencia de datos (3, 6 y 16), el parámetro MB_HOLD_REG debe definirse como un tipo de datos superior a un byte. La tabla

siguiente muestra el mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

Tabla 13-144 Mapeo de las direcciones Modbus en la memoria imagen de proceso

Funciones Modbus						**S7-1200**	
Códigos	Función	Área de datos	Rango de direcciones			Área de datos	Dirección de la CPU
01	Leer bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
02	Leer bits	Entrada	10001	a	18192	Memoria imagen de proceso de las entradas	I0.0 a I1023.7
04	Leer palabras	Entrada	30001	a	30512	Memoria imagen de proceso de las entradas	IW0 a IW1022
05	Escribir bit	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
15	Escribir bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7

Los códigos de funciones de mensajes Modbus (3, 6 y 16) entrantes leen o escriben palabras en un registro de retención Modbus que puede ser un rango de direcciones de memoria M o un bloque de datos. El tipo de registro de retención se especifica con el parámetro MB_HOLD_REG.

Nota

Asignación del parámetro MB_HOLD_REG

El registro de retención Modbus puede estar en una dirección de memoria M o en un DB global no optimizado.

Para un registro de retención Modbus en memoria M, utilice el formato de puntero estándar ANY. Es decir, en el formato P#"Bit Address" "Data Type" "Length". Por ejemplo: P#M1000.0 WORD 500.

La tabla siguiente muestra ejemplos del mapeo de las direcciones Modbus en el registro de retención usado para los códigos de función 03 (leer palabras), 06 (escribir palabras) y 16 (escribir palabras). El límite superior real de las direcciones de DB está determinado por los límites máximos de memoria de trabajo y de memoria M del respectivo modelo de CPU.

Tabla 13-145 Ejemplos de mapeo de direcciones Modbus en direcciones de memoria de CPU

Dirección de ModBus	**Ejemplos del parámetro 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]

Varias conexiones de servidor

Se pueden crear varias conexiones de servidor. Con eso, un solo PLC puede establecer conexiones concurrentes con varios clientes Modbus TCP.

Un servidor Modbus TCP puede soportar conexiones simultáneas hasta el número máximo de conexiones Open User Communication permitida por el PLC. El número total de conexiones en un PLC dado, clientes y servidores Modbus TCP incluidos, no debe sobrepasar el número máximo de conexiones Open User Communication admitidas (Página 591). Las conexiones Modbus TCP pueden compartirse entre conexiones de cliente o de servidor.

Cada conexión de servidor debe cumplir estas reglas:

- Cada conexión MB_SERVER debe usar un DB de instancia distinto
- Cada conexión MB_SERVER debe establecerse con un número de puerto IP único. Solo se admite 1 conexión por puerto.
- Cada conexión MB_SERVER debe usar una ID de conexión única.
- El MB_SERVER debe ser llamado individualmente para cada conexión (con su DB de instancia respectivo).

La ID de conexión debe ser unívoca para cada conexión individual. Es decir, solo debe usarse una ID de conexión única con cada DB de instancia individual. Resumiendo, el DB de instancia y la ID de conexión están emparejadas y deben ser únicas para cada conexión.

Tabla 13-146 Códigos de función de diagnóstico Modbus

Funciones de diagnóstico Modbus MB_SERVER		
Códigos	Subfunción	Descripción
08	0x0000	Devolver datos de consulta del test de eco: MB_SERVER reenvía a un cliente Modbus una palabra de datos que se ha recibido.
08	0x000A	Borrar contador de eventos de comunicación: La instrucción MB_SERVER borra el contador de eventos de comunicación utilizado para la función Modbus 11.
11		Consultar contador de eventos de comunicación: La instrucción MB_SERVER utiliza un contador de eventos de comunicación interno para registrar el número de peticiones de lectura y escritura Modbus correctas que se envían al servidor Modbus. El contador no incrementa con ninguna orden de Función 8 o Función 11. Tampoco se incrementa con las peticiones que producen un error de comunicación. La función Broadcast no está disponible para Modbus TCP porque solo puede haber una conexión cliente-servidor a la vez.

Variables de MB_SERVER

Esta tabla muestra las variables estáticas públicas almacenadas en el bloque de datos de instancia MB_SERVER que se pueden utilizar en el programa.

Tabla 13-147 Variables estáticas públicas de MB_SERVER

Variable	Tipo de datos	Valor predeterminado	Descripción
HR_Start_Offset	Word	0	Asigna la dirección inicial del registro de retención Modbus.
Request_Count	Word	0	Número de todas las peticiones recibidas por este servidor.
Server_Message_Count	Word	0	Número de peticiones recibidas para este servidor específico.
Xmt_Rcv_Count	Word	0	Número de transmisiones o recepciones con error detectado. También se incrementa con cada mensaje recibido que sea un mensaje Modbus no válido.
Exception_Count	Word	0	Errores específicos Modbus que requieren una excepción devuelta
Success_Count	Word	0	Número de peticiones recibidas para este servidor específico que no tienen errores de protocolo.
Conectado	Bool	0	Indica si la conexión al cliente asignado está conectada o desconectada: 1=conectada, 0=desconectada

El programa puede escribir valores en las variables HR_Start_Offset y controlar operaciones del servidor Modbus. Las demás variables se pueden leer para vigilar el estado Modbus.

HR_Start_Offset

Las direcciones del registro de retención Modbus empiezan en 40001. Estas direcciones corresponden a la dirección de memoria del PLC inicial del registro de retención. No obstante, se puede configurar la variable "HR_Start_Offset" para que defina la dirección inicial del registro de retención Modbus en un valor diferente de 40001.

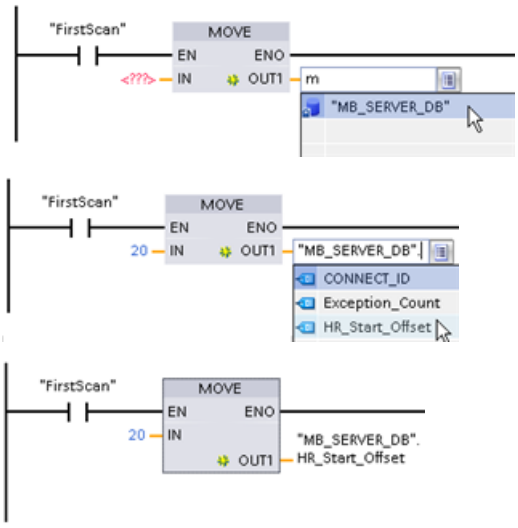
Por ejemplo, si ha configurado el registro de retención para que empiece en MW100 y su longitud es de 100 palabras. Un offset de 20 determina que la dirección inicial del registro de retención sea 40021 en vez de 40001. Cualquier dirección por debajo de 40021 y por encima de 40119 tendrá como resultado un error de direccionamiento.

Tabla 13-148 Ejemplo de direccionamiento del registro de retención Modbus

HR_Start_Offset	Dirección	Mínimo	Máximo
0	Dirección Modbus (palabra)	40001	40099
	Dirección S7-1200	MW100	MW298
20	Dirección Modbus (palabra)	40021	40119
	Dirección S7-1200	MW100	MW298

HR_Start_Offset es un valor de palabra que especifica la dirección inicial del registro de retención Modbus y que se almacena en el bloque de datos de instancia MB_SERVER. Puede ajustar el valor de esta variable estática pública utilizando la lista desplegable de asistencia de parámetros, una vez que haya insertado MB_SERVER en el programa.

Por ejemplo, una vez que haya insertado MB_SERVER en un segmento KOP, puede ir a un segmento anterior y asignar el valor HR_Start_Offset. El valor debe asignarse antes de ejecutar MB_SERVER.



Introducir una variable de servidor Modbus utilizando el nombre de DB predeterminado:

1. Posicione el cursor en el campo del parámetro y escriba un carácter m.
2. Seleccione "MB_SERVER_DB" en la lista desplegable de nombres de DB.
3. Seleccione "MB_SERVER_DB.HR_Start_Offset" en la lista desplegable de variables de DB.

Tabla 13-149 Códigos de condición de ejecución de MB_SERVER [1]

STATUS (W#16#)	Código de respuesta al servidor Modbus (B#16#)	Errores de protocolo Modbus
7001		MB_SERVER está esperando a que un cliente Modbus se conecte al puerto TCP asignado. Se informa de este código en la primera ejecución de una operación de conexión o de desconexión.
7002		MB_SERVER está esperando a que un cliente Modbus se conecte al puerto TCP asignado. Se informará de este código en las ejecuciones sucesivas, mientras se espera la finalización de una operación de conexión o de desconexión.
7003		Una operación de desconexión ha finalizado correctamente (válido solo para un ciclo del PLC).
8187		Puntero no válido a MB_HOLD_REG: Área demasiado pequeña
818C		Puntero a un área MB_HOLD_REG optimizada (debe ser un área de memoria M o DB no optimizada) o el temporizador de procesos bloqueados supera el límite de 55 segundos. (específico de S7-1200)
8381	01	Código de función no soportado
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o acceso fuera de los límites del área de direcciones de MB_HOLD_REG
8384	03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)

[1] Además de los errores de MB_SERVER indicados, las instrucciones de comunicación del bloque T subyacente (TCON, TDISCON, TSEND y TRCV (Página 652)) pueden devolver errores.

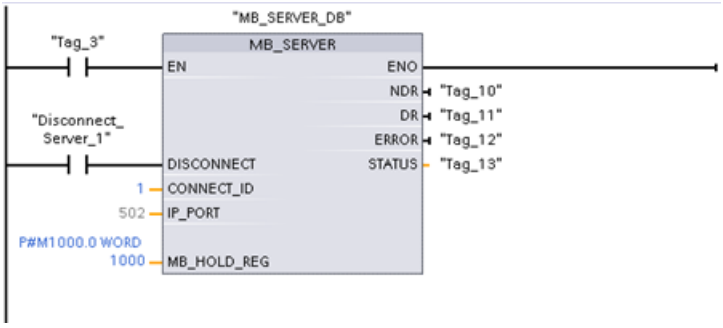
13.8.4 Ejemplos de Legacy Modbus TCP

13.8.4.1 Ejemplo: Legacy MB_SERVER conexiones TCP múltiples

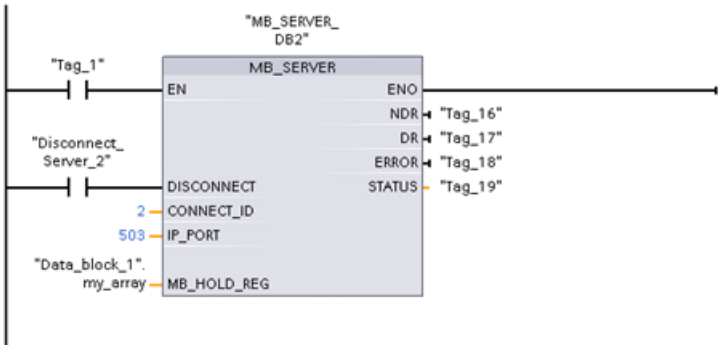
Puede haber múltiples conexiones de servidor Modbus TCP. Para conseguirlo, se debe ejecutar MB_SERVER independientemente para cada conexión. Cada conexión debe usar un DB de instancia, una ID de conexión y un puerto IP independientes. El S7-1200 solo permite una conexión por puerto IP.

Para un rendimiento óptimo, se debe ejecutar MB_SERVER cada ciclo de programa para cada conexión.

Segmento 1: Conexión n.º 1 con IP_PORT, ID de conexión y DB de instancia independientes



Segmento 2: Conexión n.º 2 con IP_PORT, ID de conexión y DB de instancia independientes



13.8.4.2 Ejemplo: Legacy MB_CLIENT 1: Peticiones múltiples con conexión TCP común

Se pueden enviar varias peticiones de cliente Modbus por la misma conexión. Para ello, se deben utilizar los mismos DB de instancia, ID de conexión y número de puerto.

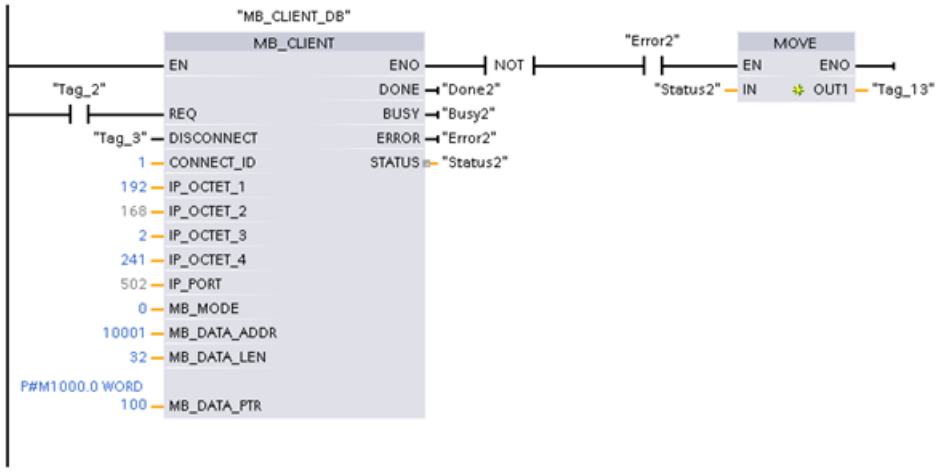
Solo puede estar activo 1 cliente a la vez. Una vez que se termina la ejecución de un cliente, empieza la ejecución del siguiente. Su programa es responsable del orden de ejecución.

El ejemplo muestra ambos clientes escribiendo en la misma área de memoria. Además, de forma opcional, se captura un error devuelto.

Segmento 1: Función Modbus 1: leer 16 bits de imagen de salida



Segmento 2: Función Modbus 2: leer 32 bits de imagen de entrada



13.8.4.3 Ejemplo: Legacy MB_CLIENT 2: Peticiones múltiples con varias conexiones TCP

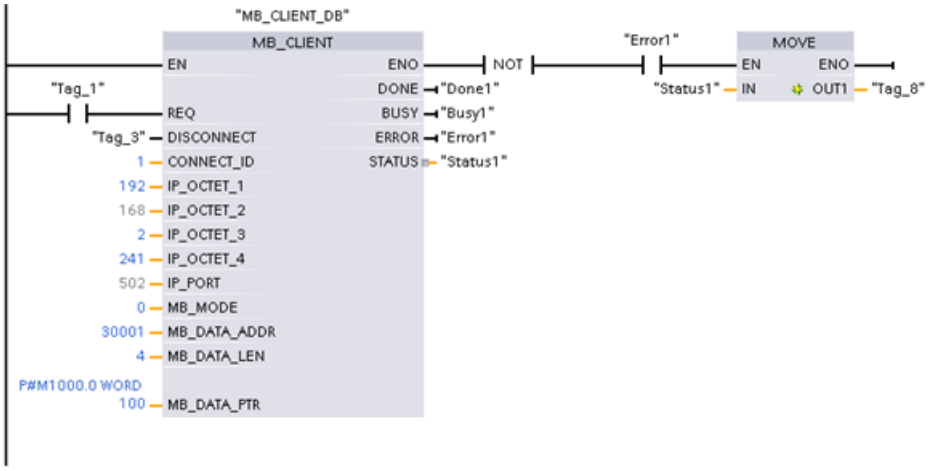
Las peticiones de cliente Modbus se pueden enviar por varias conexiones. Para ello, se deben utilizar diferentes DB de instancia, direcciones IP e ID de conexión.

El número de puerto debe ser diferente si las conexiones se establecen con el mismo servidor Modbus. Si las conexiones se realizan con servidores diferentes, no hay restricciones en número de puertos.

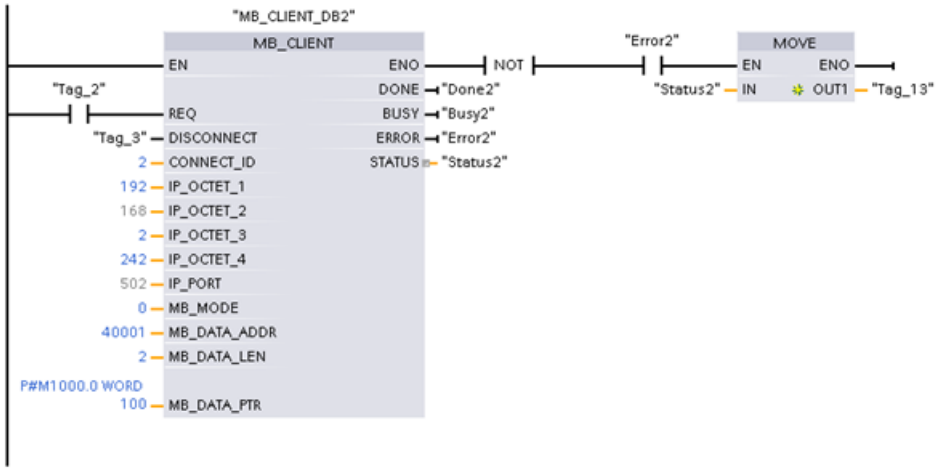
El ejemplo muestra ambos clientes escribiendo en la misma área de memoria. Además, de forma opcional, se captura un error devuelto.

Segmento 1:

Función Modbus 4: leer palabras de entrada (en memoria de S7-1200)



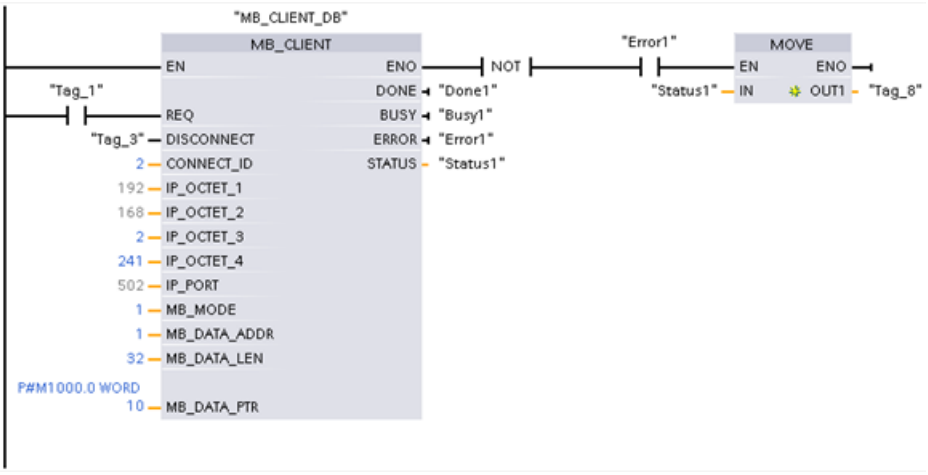
Segmento 2: Función Modbus 3: leer palabras de registro de retención de un servidor Modbus TCP



13.8.4.4 Ejemplo: Legacy MB_CLIENT 3: Petición de escritura de imagen de salida

Este ejemplo muestra la petición de un cliente Modbus para escribir en la memoria imagen de las salidas del S7-1200.

Segmento 1: Función Modbus 15: escribir bits en la memoria imagen de las salidas del S7-1200

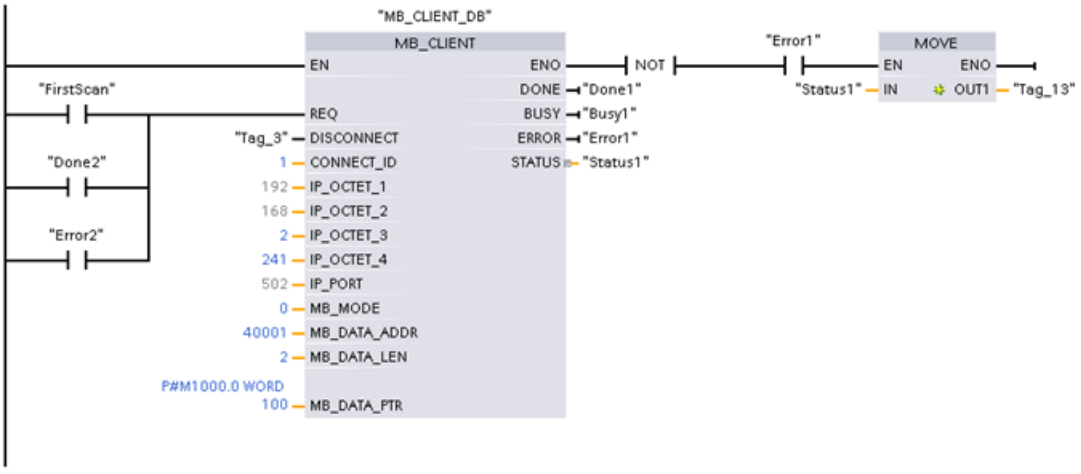


13.8.4.5 Ejemplo: Legacy MB_CLIENT 4: Coordinación de varias peticiones

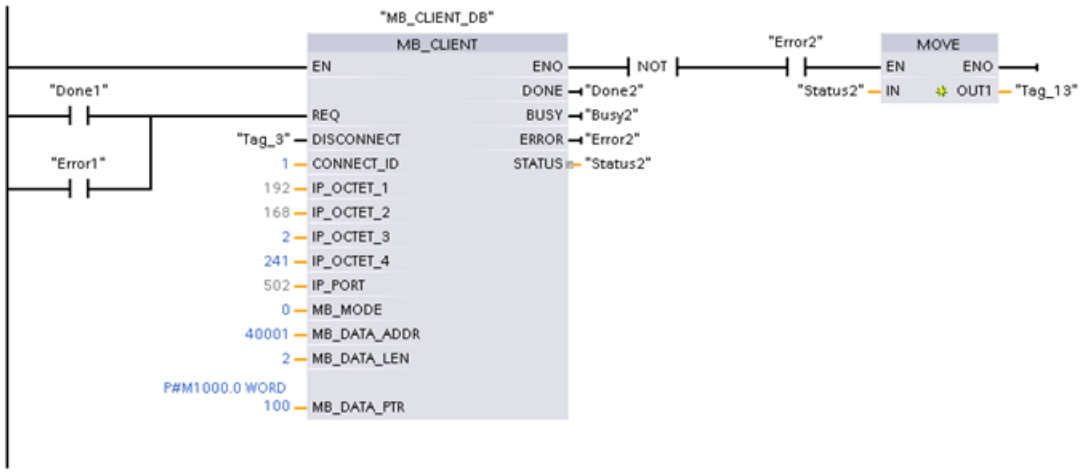
Se debe asegurar de que cada petición Modbus TCP acaba su ejecución. Su programa debe proporcionar esta coordinación. El ejemplo muestra cómo las salidas de las peticiones de cliente primera y segunda pueden usarse para coordinar la ejecución.

El ejemplo muestra ambos clientes escribiendo en la misma área de memoria. Además, de forma opcional, se captura un error devuelto.

Segmento 1: Función Modbus 3: leer palabras de registro de retención



Segmento 2: Función Modbus 3: leer palabras de registro de retención



13.9 Comunicación Legacy Modbus RTU (solo CM/CB 1241)

13.9.1 Vista general

Antes de la versión de STEP 7 V13 SP1 y las CPUs S7-1200 V4.1, las instrucciones de comunicación Modbus RTU existían con nombres diferentes y, en algunos casos, con interfaces ligeramente diferentes. Los conceptos generales son válidos para ambos juegos de instrucciones. Consulte las instrucciones Legacy Modbus RTU individuales para obtener información sobre la programación.

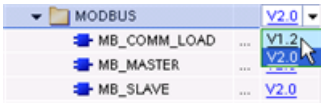
13.9.2 Seleccionar la versión de las instrucciones Modbus RTU

En STEP 7, hay disponibles dos versiones de las instrucciones Modbus RTU antiguas:

- Versión antigua 1.3: compatible con todas las versiones de CPU y CP
- Versión antigua 2.2: compatible con todas las versiones de CPU y CP
  (Nota: La configuración de la versión 2.2 agrega los parámetros REQ y DONE a MB_COMM_LOAD. En V2.2, el parámetro MB_ADDR para MB_MASTER y MB_SLAVE admite un valor UInt para el direccionamiento avanzado.)

Para garantizar la compatibilidad y facilitar la migración, se puede elegir qué versión de la instrucción se debe insertar en el programa de usuario.

En la Task Card Instrucciones, visualice las instrucciones MODBUS en el grupo Procesador de comunicaciones.



Para cambiar la versión de las instrucciones Modbus, seleccione la versión en la lista desplegable. Es posible seleccionar el grupo o bien instrucciones individuales.

Cuando se utiliza el árbol de instrucciones para insertar una instrucción Modbus en el programa, se crea una nueva instancia FB en el árbol de proyectos. El nuevo FB instancia se puede ver en el árbol de proyecto en PLC_x > Bloques de programa > Bloques de sistema > Recursos de programa.

Para comprobar la versión de la instrucción Modbus en un programa, es necesario examinar las propiedades del árbol de proyectos y no las propiedades de un cuadro que se visualiza en el editor de programas. Seleccione un FB instancia Modbus del árbol del proyecto, haga clic con el botón derecho del ratón y seleccione "Propiedades", luego seleccione la página de "información" para ver el número de versión de la instrucción Modbus.

13.9.3 Instrucciones Legacy Modbus RTU

13.9.3.1 MB_COMM_LOAD (configurar puerto en módulo PtP para Modbus RTU)

Tabla 13-150 Instrucción MB_COMM_LOAD

KOP / FUP	SCL	Descripción

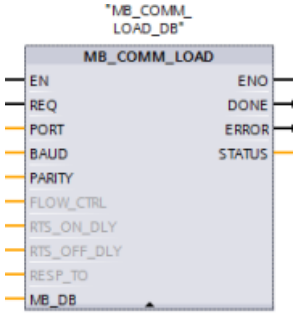
	`"MB_COMM_LOAD_DB"(`
	`REQ:=_bool_in,`
	`PORT:=_uint_in_,`
	`BAUD:=_udint_in_,`
	`PARITY:=_uint_in_,`
	`FLOW_CTRL:=_uint_in_,`
	`RTS_ON_DLY:=_uint_in_,`
	`RTS_OFF_DLY:=_uint_in_,`
	`RESP_TO:=_uint_in_,`
	`DONE=>_bool_out,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_DB:=_fbtref_inout_);`
		La instrucción MB_COMM_LOAD configura un puerto PtP para la comunicación con el protocolo Modbus RTU. Opciones de hardware del puerto Modbus: Instale un máximo de tres CMs (RS485 o RS232) más una CB (R4845). Un bloque de datos instancia se asigna automáticamente cuando la instrucción MB_COMM_LOAD se inserta en el programa.

Tabla 13-151 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación. (Solo versión 2.0)
PORT	IN	Port	Tras haber instalado y configurado un dispositivo de comunicación CM o CB, el identificador de puerto aparece en la lista desplegable de parámetros disponible en la conexión del cuadro PUERTO. El valor de puerto CM o CB asignado es la propiedad de configuración del dispositivo "identificador de hardware". El nombre simbólico del puerto se asigna en la ficha "Constantes del sistema" de la tabla de variables PLC.
BAUD	IN	UDInt	Selección de la velocidad de transferencia: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, no se permite ningún otro valor

Parámetro y tipo		Tipo de datos	Descripción
PARITY	IN	UInt	Selección de paridad: • 0 – Ninguna • 1 – Impar • 2 – Par
FLOW_CTRL [1]	IN	UInt	Selección del control de flujo: • 0 – (ajuste predeterminado) sin control de flujo • 1 – Control de flujo por hardware con RTS siempre ON (no es aplicable a los puertos RS485) • 2 – Control de flujo por hardware con RTS conmutado
RTS_ON_DLY [1]	IN	UInt	Selección de retardo RTS ON: • 0 – (ajuste predeterminado) Sin retardo desde RTS ON hasta que se transmite el primer carácter del mensaje • 1 a 65535 – Retardo en milisegundos desde RTS ON hasta que se transmite el primer carácter del mensaje (no es aplicable a los puertos RS485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RTS_OFF_DLY [1]	IN	UInt	Selección de retardo RTS OFF: • 0 – (ajuste predeterminado) Sin retardo desde el último carácter transmitido hasta que se desactiva RTS • 1 a 65535 – Retardo en milisegundos desde el último carácter transmitido hasta que se desactiva RTS (no es aplicable a los puertos RS485). Los retardos RTS se aplican siempre independientemente de la selección de FLOW_CTRL.
RESP_TO [1]	IN	UInt	Plazo de respuesta: Tiempo en milisegundos permitido por el MB_MASTER para la respuesta del esclavo. Si el esclavo no responde en este tiempo, MB_MASTER repetirá la petición o la finalizará con un error cuando se haya enviado el número de reintentos indicado. 5 ms a 65535 ms (valor predeterminado = 1000 ms).
MB_DB	IN	Variant	Referencia al bloque de datos instancia que utilizan las instrucciones MB_MASTER o MB_SLAVE. Una vez insertadas las instrucciones MB_SLAVE o MB_MASTER en el programa, el identificador de DB aparece en la lista desplegable de parámetros disponible en la conexión MB_DB del cuadro.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores. (Solo versión 2.0)
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

[1] Parámetros opcionales para MB_COMM_LOAD (V 2.x o superior). Haga clic en la flecha situada en la parte inferior de una casilla KOP/FUP para expandir la casilla e incluir estos parámetros.

MB_COMM_LOAD se ejecuta para configurar un puerto para el protocolo Modbus RTU. Al configurar un puerto para el protocolo Modbus RTU, el mismo solo puede ser utilizado por la instrucción MB_MASTER o MB_SLAVE.

Es preciso utilizar una ejecución de MB_COMM_LOAD para configurar cada uno de los puertos de comunicación utilizado para la comunicación Modbus. Asígnele a cada puerto utilizado un

DB de instancia MB_COMM_LOAD unívoco. En la CPU se pueden instalar hasta tres módulos de comunicación (RS232 o RS485) y una placa de comunicación (RS485). Llame MB_COMM_LOAD desde un OB de arranque y ejecute esta instrucción una vez, o bien utilice la marca de sistema del primer ciclo (Página 93) para iniciar la llamada y ejecutarla una vez. Ejecute nuevamente MB_COMM_LOAD solo si es necesario modificar los parámetros de comunicación, p. ej. la velocidad de transferencia o la paridad.

Un bloque de datos instancia se asigna para MB_MASTER o MB_SLAVE cuando estas instrucciones se insertan en el programa. Este bloque de datos instancia se referencia cuando se especifica el parámetro MB_DB para la instrucción MB_COMM_LOAD.

Variables del bloque de datos MB_COMM_LOAD

La tabla siguiente muestra las variables estáticas públicas almacenadas en el DB de instancia MB_COMM_LOAD que se pueden utilizar en el programa.

Tabla 13-152 Variables estáticas en el DB de instancia

Variable	Tipo de datos	Descripción
ICHAR_GAP	UInt	Retardo del tiempo excedido entre caracteres. Este parámetro está especificado en milisegundos y se utiliza para aumentar el tiempo previsto entre caracteres recibidos. El número correspondiente de tiempos de bit para el parámetro se agrega al valor predeterminado Modbus de 35 tiempos de bit (3,5 tiempos de carácter).
RETRIES	UInt	Número de reintentos del maestro antes de devolver el código de error "sin respuesta" 0x80C8.
STOP_BITS	USInt	Número de bits de parada utilizados para ajustar cada carácter. Los valores válidos son 1 y 2.

Tabla 13-153 Códigos de condición de ejecución de MB_COMM_LOAD [1]

STATUS (W#16#)	Descripción
0000	No hay error
8180	ID de puerto no válida (identificador de puerto/hardware incorrecto para el módulo de comunicación)
8181	Velocidad de transferencia no válida
8182	Paridad no válida
8183	Valor de control de flujo no válido
8184	Valor de timeout de respuesta no válido (timeout de respuesta por debajo del valor mínimo de 5 ms)
8185	El parámetro MB_DB no es un bloque de datos instancia de una instrucción MB_MASTER o MB_SLAVE.

[1] Además de los errores MB_COMM_LOAD indicados arriba, las instrucciones de comunicaciones PtP subyacentes pueden devolver errores.

13.9.3.2 MB_MASTER (comunicar como maestro Modbus vía puerto PtP)

Tabla 13-154 Instrucción MB_MASTER

KOP / FUP	SCL	Descripción

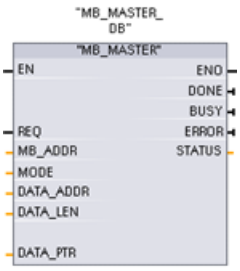
	`"MB_MASTER_DB"(`
	`REQ:=_bool_in_,`
	`MB_ADDR:=_uint_in_,`
	`MODE:=_usint_in_,`
	`DATA_ADDR:=_udint_in_,`
	`DATA_LEN:=_uint_in_,`
	`DONE=>_bool_out_,`
	`BUSY=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`DATA_PTR:=_variant_inout_);`
		La instrucción MB_MASTER se comunica como un maestro Modbus utilizando un puerto configurado por una ejecución anterior de la instrucción MB_COMM_LOAD. Un bloque de datos de instancia se asigna automáticamente cuando la instrucción MB_MASTER se inserta en el programa. El bloque de datos de instancia MB_MASTER se utiliza cuando se especifica el parámetro MB_DB para la instrucción MB_COMM_LOAD.

Tabla 13-155 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
REQ	IN	Bool	0 = No hay ninguna petición 1 = Petición de transmitir datos a esclavo Modbus
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Dirección de estación Modbus RTU: Rango de direccionamiento estándar (de 1 a 247) Rango de direccionamiento avanzado (de 1 a 65535) El valor 0 está reservado para enviar un mensaje Broadcast a todos los esclavos Modbus. Los códigos de función Modbus 05, 06, 15 y 16 son los únicos que se soportan para el Broadcast.
MODE	IN	USInt	Selección de modo: Especifica el tipo de petición (lectura, escritura o diagnóstico). Consulte los detalles en la tabla de funciones Modbus que aparece más abajo.
DATA_ADDR	IN	UDInt	Dirección inicial en el esclavo: Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus. Las direcciones válidas se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_LEN	IN	UInt	Longitud de datos: Determina el número de bits o palabras a las que debe accederse en esta petición. Las longitudes válidas se indican en la tabla de funciones Modbus que aparece más abajo.
DATA_PTR	IN	Variant	Puntero a los datos: Apunta a la dirección del M o del DB (tipo de DB no optimizado) de los datos que se están escribiendo o leyendo.
DONE	OUT	Bool	El bit DONE es TRUE durante un ciclo tras haberse finalizado la última petición sin errores.
BUSY	OUT	Bool	• 0 – No hay ninguna operación MB_MASTER en curso • 1 – Operación MB_MASTER en curso
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en el parámetro STATUS solo es válido durante un único ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de condición de ejecución

Reglas de comunicación del maestro Modbus

- MB_COMM_LOAD debe ejecutarse para configurar un puerto antes de que la instrucción MB_MASTER pueda comunicarse con ese puerto.
- Si un puerto debe utilizarse para iniciar peticiones de maestro Modbus, MB_SLAVE no debe utilizar este puerto. Una o más instancias de la ejecución de MB_MASTER pueden utilizarse en ese puerto, no obstante todas las ejecuciones de MB_MASTER deben utilizar el mismo DB de instancia MB_MASTER para ese puerto.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe consultar la instrucción MB_MASTER para transmitir y recibir condiciones completas.
- Llame todas las ejecuciones de MB_MASTER para un puerto determinado desde un OB de ciclo de programa. Las instrucciones para maestro Modbus solo se pueden ejecutar en un nivel de ejecución, ya sea en el de retardo/cíclico o bien en el de ciclo del programa. Estas no se deben ejecutar en ambos niveles de prioridad. Si una instrucción de maestro Modbus de un nivel de prioridad de ejecución más alto inhibe la ejecución de otra instrucción de maestro Modbus, se puede obtener un funcionamiento incorrecto. Las instrucciones de maestro Modbus no deben ejecutarse en los niveles de prioridad para arranque, diagnóstico o error de tiempo.
- Cuando una instrucción de maestro inicia una transmisión, esta instancia debe ejecutarse continuamente con la entrada EN habilitada hasta que se devuelva un estado DONE=1 o un estado ERROR=1. Una instancia MB_MASTER en particular se considera activa hasta que ocurra uno de estos dos eventos. Mientras la instancia original esté activa, cualquier llamada a otra instancia con la salida REQ habilitada resultará en un error. Si la ejecución continua de la instancia original se detiene, el estado de la petición se mantiene activo por un periodo de tiempo especificado por la variable estática Blocked_Proc_Timeout. Una vez transcurre este periodo de tiempo, la siguiente instrucción de maestro llamada con una entrada REQ habilitada se convertirá en la instancia activa. Esto impide que una sola instancia de maestro Modbus monopolice o cierre el acceso a un puerto. Si la instancia original que está activa no se habilita dentro del periodo de tiempo especificado por la variable estática "Blocked_Proc_Timeout", entonces la siguiente ejecución realizada por esta instancia (con REQ sin activar) borrará el estado activo. Si REQ está activada, entonces esta ejecución inicia una nueva petición del maestro como si ninguna otra instancia estuviese activa.

Parámetro REQ

0 = No hay ninguna petición; 1 = Petición de transmitir datos a esclavo Modbus

Esta entrada se puede controlar con un contacto activado por nivel o por flanco. Siempre que esta entrada esté habilitada, se inicia un autómata finito para garantizar que ningún otro MB_MASTER que use el mismo DB de instancia pueda emitir una petición mientras no se complete la petición actual. Todos los demás estados de entrada se capturan y se retienen internamente para la petición actual hasta que se recibe la respuesta o se detecta un error.

Si la misma instancia de MB_MASTER se ejecuta nuevamente con la entrada REQ = 1 antes de que se finalice la petición actual, no se realizarán más transmisiones. No obstante, al finalizar la petición se emite una nueva petición siempre que se vuelva a ejecutar MB_MASTER con entrada REQ = 1.

Los parámetros DATA_ADDR y MODE seleccionan el tipo de función Modbus

DATA_ADDR (dirección Modbus inicial en el esclavo): Determina la dirección inicial de los datos a los que debe accederse en el esclavo Modbus.

La instrucción MB_MASTER utiliza la entrada MODE en vez de una entrada de código de función. La combinación de MODE y la dirección Modbus determinan el código de función utilizado en el mensaje Modbus real. La tabla siguiente muestra la correlación entre el parámetro MODE, el código de función Modbus y el rango de direcciones Modbus.

Tabla 13-156 Funciones Modbus

MODE	Función Modbus	Longitud de datos	Operación y datos	Dirección de Modbus
0	01	De 1 a 2000 De 1 a 1992 [1]	Leer bits de salida: De 1 a (1992 o 2000) bits por petición	De 1 a 9999
0	02	De 1 a 2000 De 1 a 1992 [1]	Leer bits de entrada: De 1 a (1992 o 2000) bits por petición	De 10001 a 19999
0	03	De 1 a 125 De 1 a 124 [1]	Leer registros de retención: 1 a (124 o 125) palabras por petición	De 40001 a 49999 o De 400001 a 465535
0	04	De 1 a 125 De 1 a 124 [1]	Leer palabras de entrada: De 1 a (124 o 125) palabras por petición	De 30001 a 39999
1	05	1	Escribir un bit de salida: Un bit por petición	De 1 a 9999
1	06	1	Escribir un registro de retención: 1 palabra por petición	De 40001 a 49999 o De 400001 a 465535
1	15	De 2 a 1968 De 2 a 1960 [1]	Escribir varios bits de salida: 2 a (1960 o 1968) bits por petición	De 1 a 9999
1	16	De 2 a 123 De 2 a 122 [1]	Escribir varios registros de retención: De 2 a (122 o 123) palabras por petición	De 40001 a 49999 o De 400001 a 465535
2	15	De 1 a 1968 De 2 a 1960 [1]	Escribir uno o más bits de salida: De 1 a (1960 o 1968) bits por petición	De 1 a 9999
2	16	De 1 a 123 De 1 a 122 [1]	Escribir uno o más registros de retención: De 1 a (122 o 123) palabras por petición	De 40001 a 49999 o De 400001 a 465535
11	11	0	Leer la palabra de estado de la comunicación del esclavo y el contador de eventos. Esta palabra de estado indica ocupado (0 – no ocupado, 0xFFFF - ocupado). El contador de eventos se incrementa cada vez que se finaliza un mensaje correctamente. En esta función se ignoran ambos operandos de MB_MASTER: DATA_ADDR y DATA_LEN.	
80	08	1	Compruebe el estado del esclavo utilizando el código de diagnóstico de datos 0x0000 (test de Loopback – el esclavo envía de regreso la petición) 1 palabra por petición	

MODE	Función Modbus	Longitud de datos	Operación y datos	Dirección de Modbus
81	08	1	Inicializar el contador de eventos del esclavo utilizando un código de diagnóstico de datos 0x000A 1 palabra por petición	
De 3 a 10, de 12 a 79, de 82 a 255			Reservado	

[1] Para el modo "Direccionamiento avanzado", las longitudes máximas de los datos se reduce en 1 byte o 1 palabra dependiendo del tipo de datos utilizados por la función.

Parámetro DATA_PTR

El parámetro DATA_PTR apunta a la dirección del M o del DB en que se escribe o bien del que se lee. Si se utiliza un bloque de datos, hay que crear un bloque de datos global que ofrezca un almacenamiento de datos para operaciones de lectura y escritura en esclavos Modbus.

Nota

El tipo de bloque de datos debe permitir un direccionamiento directo.

El bloque de datos debe permitir tanto el direccionamiento directo (absoluto) como el simbólico. Al crear el bloque de datos debe seleccionarse el atributo de acceso "estándar".

Estructuras del bloque de datos para el parámetro DATA_PTR

- Estos tipos de datos son válidos para la **lectura de palabras** de las direcciones Modbus de 30001 a 39999, de 40001 a 49999 y de 400001 a 465536, así como para la **escritura de palabras** en las direcciones Modbus 40001 a 49999 y 400001 a 465536.
  - Matriz estándar de tipos de datos WORD, UINT o INT
  - Estructura WORD, UINT o INT con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 bits.
  - Estructura compleja con nombres, en la que todo elemento tiene un nombre unívoco y un tipo de datos de 16 o 32 bits.
- Para la **lectura** y escritura de bits de las direcciones Modbus 00001 a 09999 y lecturas de bits de 10001 a 19999.
  - Matriz estándar de tipos de datos booleanos.
  - Estructura booleana con nombres que incluye variables booleanas con nombres unívocos.

- Aunque no es imprescindible, se recomienda que cada instrucción MB_MASTER tenga su propia área de memoria. El motivo de esta recomendación es que la posibilidad de que se corrompan los datos aumenta si varias instrucciones MB_MASTER están leyendo y escribiendo en la misma área de memoria.
- No es necesario que las áreas de datos de DATA_PTR se encuentren en el mismo bloque de datos global. Es posible crear un bloque de datos con varias áreas para lecturas Modbus, uno para escrituras Modbus, o bien uno para cada estación esclava.

Variables del bloque de datos del maestro Modbus

La tabla siguiente muestra las variables estáticas públicas almacenadas en el DB de instancia MB_MASTER que se pueden utilizar en el programa.

Tabla 13-157 Variables estáticas en el DB de instancia

Variable	Tipo de datos	Valor inicial	Descripción
Blocked_Proc_Timeout	Real	3,0	Tiempo (en segundos) que hay que esperar a una instancia maestro Modbus boqueada antes de eliminar esta instancia por estar ACTIVA. Esto puede ocurrir, por ejemplo, si se ha lanzado una petición del maestro y luego el programa deja de llamar la función del maestro antes de que haya finalizado por completo la petición. El valor de tiempo debe ser mayor que 0 y menor que 55 segundos; de lo contrario se producirá un error. El valor predeterminado es 0,5 segundos.
Extended_Addressing	Bool	False	Configura el direccionamiento de esclavos de uno o dos bytes. El valor predeterminado es 0. (0=dirección de un byte, 1=dirección de dos bytes)

El programa puede escribir valores en las variables Blocked_Proc_Timeout y Extended_Addressing para controlar operaciones del maestro Modbus. Consulte en el apartado MB_SLAVE la descripción de HR_Start_Offset y Extended_Addressing para obtener un ejemplo de cómo utilizar estas variables en el editor de programas así como detalles relacionados con el direccionamiento avanzado Modbus. (Página 1168)

Códigos de condición

Tabla 13-158 Códigos de condición de ejecución MB_MASTER (errores de comunicación y configuración) [1]

STATUS (W#16#)	Descripción
0000	No hay error
80C8	Timeout del esclavo. Compruebe la velocidad de transferencia, la paridad y el cableado del esclavo.
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE.
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno.
80E1	El mensaje se ha terminado debido a un error de paridad.
80E2	El mensaje se ha terminado debido a un error de trama.
80E3	El mensaje se ha terminado debido a un error de desbordamiento.

STATUS (W#16#)	Descripción
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total.
8180	ID de puerto no válida o error en la instrucción MB_COMM_LOAD
8186	Dirección de estación Modbus no válida
8188	Modo no válido especificado para petición Broadcast
8189	Valor de dirección de datos no válido
818A	Valor de longitud de datos no válido
818B	Puntero no válido al origen/destino de datos local: tamaño incorrecto
818C	Puntero no válido para DATA_PTR o Blocked_Proc_Timeout no válido: El área de datos debe ser un DB (que permita acceso tanto simbólico como directo) o una memoria M.
8200	El puerto está ocupado porque está procesando una petición de transmisión.

Tabla 13-159 Códigos de condición de ejecución MB_MASTER (errores de protocolo Modbus) [1]

STATUS (W#16#)	Código de respuesta del esclavo	Errores de protocolo Modbus
8380	-	Error CRC
8381	01	Código de función no soportado
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o dirección fuera del rango válido del área DATA_PTR
8384	Más que 03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)
8386	-	El código de función en la respuesta no coincide con el código de la petición.
8387	-	Ha respondido el esclavo incorrecto
8388	-	La respuesta del esclavo a una petición de escritura es incorrecta. La petición de escritura devuelta por el esclavo no coincide con lo que el maestro ha enviado realmente.

[1] Además de los errores MB_MASTER indicados arriba, las instrucciones de comunicaciones PtP subyacentes pueden devolver errores.

13.9.3.3 MB_SLAVE (comunicar como esclavo Modbus vía puerto PtP)

Tabla 13-160 Instrucción MB_SLAVE

KOP / FUP	SCL	Descripción

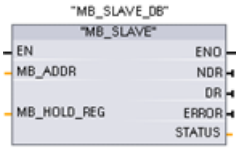
	`"MB_SLAVE_DB"(`
	`MB_ADDR:=_uint_in_,`
	`NDR=>_bool_out_,`
	`DR=>_bool_out_,`
	`ERROR=>_bool_out_,`
	`STATUS=>_word_out_,`
	`MB_HOLD_REG:=_variant_inout_);`
		La instrucción MB_SLAVE permite al programa comunicarse como un esclavo Modbus a través de un puerto PtP en el CM (RS485 o RS232) y la CB (RS485). Cuando un maestro Modbus RTU remoto lanza una petición, el programa de usuario responde con la ejecución de MB_SLAVE. STEP 7 crea automáticamente un DB de instancia al introducir la instrucción. Utilice este nombre de MB_SLAVE_DB al especificar el parámetro MB_DB para la instrucción MB_COMM_LOAD.

Tabla 13-161 Tipos de datos para los parámetros

Parámetro y tipo		Tipo de datos	Descripción
MB_ADDR	IN	V1.0: USInt V2.0: UInt	La dirección de estación del esclavo Modbus: Rango de direccionamiento estándar (de 1 a 247) Rango de direccionamiento avanzado (de 0 a 65535)
MB_HOLD_REG	IN	Variant	Puntero hacia el DB del registro de retención Modbus: El registro de retención Modbus puede ser un área de marcas o un bloque de datos.
NDR	OUT	Bool	Nuevos datos listos: • 0 – No hay datos nuevos • 1 – Indica que el maestro Modbus ha escrito datos nuevos
DR	OUT	Bool	Lectura de datos: • 0 – No se han leído datos • 1 – Indica que el maestro Modbus ha leído datos
ERROR	OUT	Bool	El bit ERROR es TRUE durante un ciclo tras haberse finalizado la última petición con un error. Si la ejecución ha finalizado con un error, el valor del código de error en el parámetro STATUS solo es válido durante un ciclo en que ERROR = TRUE.
STATUS	OUT	Word	Código de error de ejecución

Los códigos de las funciones de comunicación Modbus (1, 2, 4, 5 y 15) pueden leer y escribir bits y palabras directamente en la memoria imagen de proceso de las entradas y salidas de la CPU. Para estos códigos de función, el parámetro MB_HOLD_REG debe definirse como un tipo de datos superior a un byte. La tabla siguiente muestra un ejemplo del mapeo de las direcciones Modbus en la memoria imagen de proceso de la CPU.

Tabla 13-162 Mapeo de las direcciones Modbus en la memoria imagen de proceso

Funciones Modbus						S7-1200	
Códigos	Función	Área de datos	Rango de direcciones			Área de datos	Dirección de la CPU
01	Leer bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7

Funciones Modbus						S7-1200	
02	Leer bits	Entrada	10001	a	18192	Memoria imagen de proceso de las entradas	I0.0 a I1023.7
04	Leer palabras	Entrada	30001	a	30512	Memoria imagen de proceso de las entradas	IW0 a IW1022
05	Escribir bit	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7
15	Escribir bits	Salida	1	a	8192	Memoria imagen de proceso de las salidas	Q0.0 a Q1023.7

Los códigos de las funciones de comunicación Modbus (3, 6, 16) utilizan un registro de retención Modbus que puede ser un rango de dirección de memoria M o un bloque de datos. El tipo de registro de retención se especifica con el parámetro MB_HOLD_REG de la instrucción MB_SLAVE.

Nota

Tipo de bloque de datos MB_HOLD_REG

Un bloque de datos del registro de retención Modbus debe permitir tanto el direccionamiento directo (absoluto) como el simbólico. Al crear el bloque de datos debe seleccionarse el atributo de acceso "estándar".

La tabla siguiente muestra ejemplos del mapeo de las direcciones Modbus en el registro de retención para los códigos de función 03 (leer palabras), 06 (escribir palabras) y 16 (escribir palabras). El límite superior real de las direcciones de DB está determinado por los límites máximos de memoria de trabajo y de memoria M del respectivo modelo de CPU.

Tabla 13-163 Mapeo de las direcciones Modbus en la memoria de la CPU

Dirección del maestro Modbus	Ejemplos del parámetro 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]

Tabla 13-164 Funciones de diagnóstico

Funciones de diagnóstico Modbus de MB_SLAVE en el S7-1200		
Códigos	Subfunción	Descripción
08	0000H	Devolver datos de consulta del test de eco: La instrucción MB_SLAVE responde al maestro Modbus con una palabra de datos que se están recibiendo datos.

Funciones de diagnóstico Modbus de MB_SLAVE en el S7-1200		
08	000AH	Borrar contador de eventos de comunicación: La instrucción MB_SLAVE borra el contador de eventos de comunicación utilizado para la función Modbus 11.
11		Consultar contador de eventos de comunicación: La instrucción MB_SLAVE utiliza un contador de eventos de comunicación interno para registrar el número de peticiones de lectura y escritura Modbus correctas que se envían al esclavo Modbus. El contador no se incrementa con las funciones 8 ni 11, ni tampoco con peticiones Broadcast. Tampoco se incrementa con peticiones que resulten en un error de comunicación (p. ej. errores de paridad o CRC).

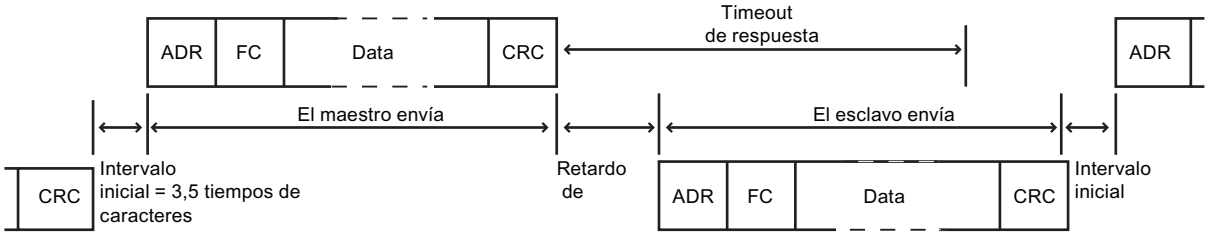
La instrucción MB_SLAVE soporta peticiones de escritura Broadcast de cualquier maestro Modbus, mientras que la petición sea para acceder a direcciones válidas. MB_SLAVE generará el código de error 0x8188 para códigos de función no soportados en Broadcast.

Reglas de comunicación del esclavo Modbus

- MB_COMM_LOAD debe ejecutarse para configurar un puerto antes de que la instrucción MB_SLAVE pueda comunicarse a través de ese puerto.
- Si un puerto debe responder como esclavo a un maestro Modbus, no programe este puerto con la instrucción MB_MASTER.
- Solo se puede utilizar una instancia MB_SLAVE en un determinado puerto, de lo contrario puede presentarse un comportamiento erróneo.
- Las instrucciones Modbus no utilizan eventos de alarma de comunicación para controlar el proceso de comunicación. El programa debe controlar el proceso de comunicación consultando la instrucción MB_SLAVE para comprobar si se han finalizado las operaciones de transmisión y recepción.
- La instrucción MB_SLAVE debe ejecutarse periódicamente a una frecuencia que permita responder sin demora a las peticiones entrantes de un maestro Modbus. Se recomienda ejecutar MB_SLAVE en cada ciclo desde un OB de ciclo de programa. Es posible ejecutar MB_SLAVE desde un OB de alarma cíclica, pero no es recomendable porque el potencial de retardos excesivos en el subprograma bloquea temporalmente la ejecución de otros subprogramas.

Temporización de señales Modbus

MB_SLAVE debe ejecutarse periódicamente para recibir todas las peticiones del maestro Modbus y responder según sea necesario. La frecuencia de ejecución de MB_SLAVE depende del periodo de timeout de respuesta del maestro Modbus. Esto se ilustra en el diagrama siguiente.



El periodo de timeout de respuesta RESP_TO es el tiempo que un maestro Modbus espera hasta el inicio de la respuesta de un esclavo Modbus. Este periodo no está definido en el protocolo

Modbus, sino que es un parámetro de todo maestro Modbus. La frecuencia de ejecución (es decir, el tiempo que transcurre entre dos ejecuciones) de MB_SLAVE debe basarse en los parámetros particulares del maestro Modbus. Como mínimo, MB_SLAVE debería ejecutarse dos veces en el periodo de timeout de respuesta del maestro Modbus.

Variables del esclavo Modbus

Esta tabla muestra las variables estáticas públicas almacenadas en el bloque de datos de instancia MB_SLAVE que se pueden utilizar en el programa.

Tabla 13-165 Variables del esclavo Modbus

Variable	Tipo de datos	Descripción
Request_Count	Word	Número de todas las peticiones recibidas por este esclavo
Slave_Message_Count	Word	Número de peticiones recibidas para este esclavo específico
Bad_CRC_Count	Word	Número de peticiones recibidas que tienen un error CRC
Broadcast_Count	Word	Número de peticiones Broadcast recibidas
Exception_Count	Word	Errores específicos Modbus que requieren una excepción devuelta
Success_Count	Word	Número de peticiones recibidas para este esclavo específico que no tienen errores de protocolo
HR_Start_Offset	Word	Especifica la dirección inicial del registro de retención Modbus (valor predeterminado = 0)
Extended_Addressing	Bool	Configura el direccionamiento de esclavos de uno o dos bytes (0=dirección de un byte, 1=dirección de dos bytes, valor predeterminado = 0)

El programa puede escribir valores en las variables HR_Start_Offset y Extended_Addressing y controlar operaciones del esclavo Modbus. Las demás variables se pueden leer para vigilar el estado Modbus.

HR_Start_Offset

Las direcciones del registro de retención Modbus empiezan en 40001 o 400001. Estas direcciones corresponden a la dirección de memoria del PLC inicial del registro de retención. No obstante, es posible configurar la variable "HR_Start_Offset" para que defina la dirección inicial del registro de retención Modbus en otro valor diferente a 40001 o 400001.

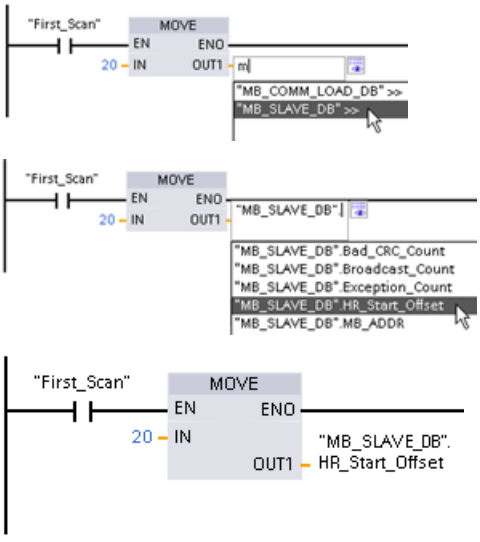
Por ejemplo, si ha configurado el registro de retención para que empiece en MW100 y su longitud es de 100 palabras. Un offset de 20 determina que la dirección inicial del registro de retención sea 40021 en vez de 40001. Cualquier dirección por debajo de 40021 y por encima de 400119 tendrá como resultado un error de direccionamiento.

Tabla 13-166 Ejemplo de direccionamiento del registro de retención Modbus

HR_Start_Offset	Dirección	Mínimo	Máximo
0	Dirección Modbus (palabra)	40001	40099
	Dirección S7-1200	MW100	MW298
20	Dirección Modbus (palabra)	40021	40119
	Dirección S7-1200	MW100	MW298

HR_Start_Offset es un valor de palabra que especifica la dirección inicial del registro de retención Modbus y que se almacena en el bloque de datos de instancia MB_SLAVE. Es posible ajustar el valor de esta variable estática pública utilizando la lista desplegable de parámetros, una vez haya insertado MB_SLAVE en el programa.

Por ejemplo, una vez haya insertado MB_SLAVE en un segmento KOP, es posible ir a un segmento anterior y asignar el valor HR_Start_Offset. El valor debe asignarse antes de ejecutar MB_SLAVE.



Introducir una variable de esclavo Modbus utilizando el nombre de DB predeterminado:

1. Posicione el cursor en el campo del parámetro y escriba un carácter m.
2. Seleccione "MB_SLAVE_DB" en la lista desplegable.
3. Posicione el cursor en el lado derecho del nombre del DB (detrás de las comillas) e introduzca un punto.
4. Seleccione "MB_SLAVE_DB.HR_Start_Offset" en la lista desplegable.

Extended_Addressing

A la variable Extended_Addressing se accede de una manera similar que a la variable HR_Start_Offset mencionada más arriba, excepto que la variable Extended_Addressing es un valor booleano. El valor booleano se debe escribir a través de una bobina de salida y no de un cuadro de desplazamiento.

El direccionamiento del esclavo Modbus se puede configurar de modo que tenga un solo byte (ajuste predeterminado) o bien dos bytes. El direccionamiento avanzado se utiliza para direccionar más de 247 dispositivos dentro de una sola red. El direccionamiento avanzado permite direccionar como máximo 64000 direcciones. A continuación aparece una trama de la función 1 Modbus a modo de ejemplo.

Tabla 13-167 Dirección de esclavo de un byte (byte 0)

Función 1	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	
Petición	Direc. esclava	Código F	Dirección inicial		Long. de bobinas		
Respuesta válida	Direc. esclava	Código F	Longitud	Datos bobina			
Respuesta errónea	Direc. esclava	0x81	Código E				

Tabla 13-168 Dirección de esclavo de dos bytes (byte 0 y byte 1)

	Byte 0	**Byte 1**	**Byte 2**	**Byte 3**	**Byte 4**	**Byte 5**	**Byte 6**
Petición	Dirección esclava		Código F	Dirección inicial		Long. de bobinas	
Respuesta válida	Dirección esclava		Código F	Longitud	Datos bobina		
Respuesta errónea	Dirección esclava		0x81	Código E			

Códigos de condición

Tabla 13-169 Códigos de condición de ejecución MB_SLAVE (errores de comunicación y configuración) [1]

STATUS (W#16#)	**Descripción**
80D1	El receptor ha lanzado una petición de control de flujo para suspender una transmisión activa y no ha habilitado nuevamente la transmisión en el tiempo de espera indicado. Este error también se genera durante el control de flujo por hardware cuando el receptor no confirma CTS en el tiempo de espera indicado.
80D2	La petición de transmisión se ha cancelado porque no se recibe ninguna señal DSR del DCE.
80E0	El mensaje se ha terminado porque el búfer de recepción está lleno.
80E1	El mensaje se ha terminado debido a un error de paridad.
80E2	El mensaje se ha terminado debido a un error de trama.
80E3	El mensaje se ha terminado debido a un error de desbordamiento.
80E4	El mensaje se ha terminado debido a que la longitud especificada excede el tamaño del búfer total.
8180	ID de puerto no válida o error en la instrucción MB_COMM_LOAD
8186	Dirección de estación Modbus no válida
8187	Puntero no válido a MB_HOLD_REG DB: Área demasiado pequeña
818C	Puntero MB_HOLD_REG no válido al área de marcas o al DB (el área DB debe permitir un direccionamiento tanto simbólico como directo)

Tabla 13-170 Códigos de condición de ejecución MB_SLAVE (errores de protocolo Modbus) [1]

STATUS (W#16#)	**Código de respuesta del esclavo**	**Errores de protocolo Modbus**
8380	Sin respuesta	Error CRC
8381	01	Código de función no soportado o no soportado en Broadcast
8382	03	Error de longitud de datos
8383	02	Error en la dirección de los datos o dirección fuera del rango válido del área DATA_PTR
8384	03	Error de valor de datos
8385	03	Valor de código de diagnóstico de datos no soportado (código de función 08)

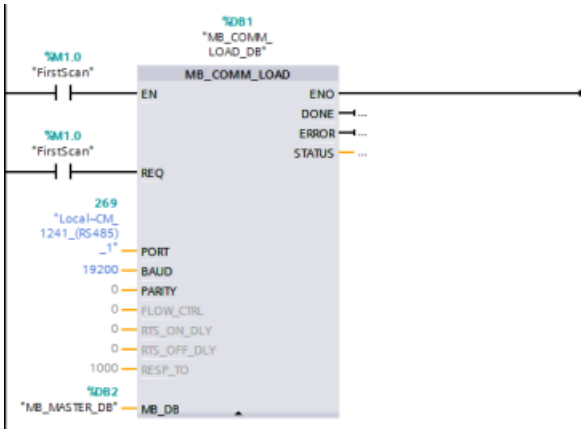
[1] Además de los errores MB_SLAVE indicados arriba, las instrucciones de comunicaciones PtP subyacentes pueden devolver errores.

13.9.4 Ejemplos de Modbus RTU antiguo

13.9.4.1 Ejemplo: programa de maestro Legacy Modbus RTU

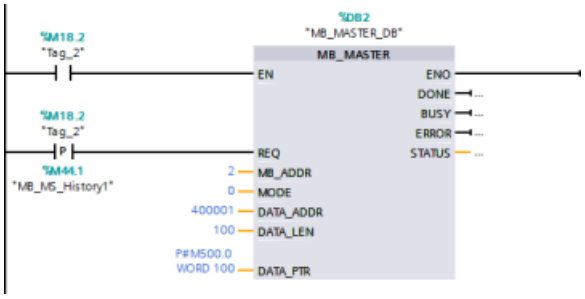
MB_COMM_LOAD se inicializa durante el arranque mediante la primera marca de exploración. La ejecución de MB_COMM_LOAD de esa forma solo debe hacerse cuando la configuración de puerto serie no vaya a cambiar en tiempo de ejecución.

Segmento 1: Configurar/inicializar el puerto de comunicación del módulo RS485 una sola vez durante el primer ciclo.

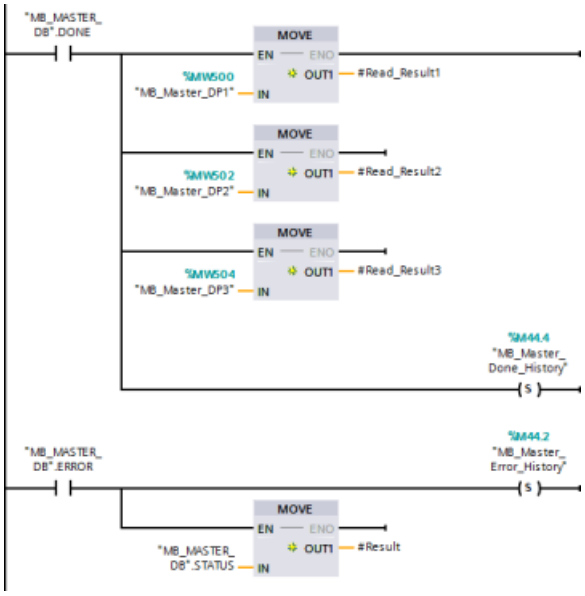


Una instrucción MB_MASTER se usa en el OB del ciclo de programa para comunicar con un solo esclavo. Se pueden usar más instrucciones MB_MASTER en el OB del ciclo de programa para comunicar con otros esclavos o se puede reutilizar un FB MB_MASTER para comunicar con más esclavos.

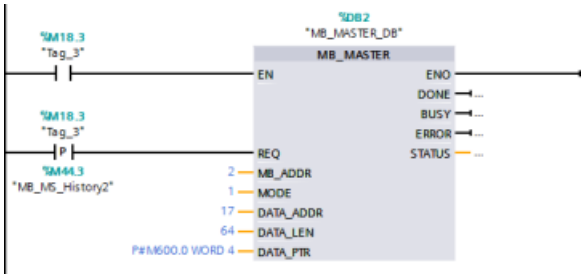
Segmento 2: Leer 100 palabras de los datos del registro de retención desde la ubicación 400001 en el esclavo #2 hasta la ubicación de memoria MW500-MW698.



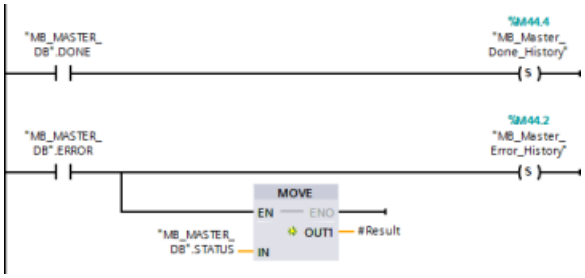
Segmento 3: Mover las 3 primeras palabras de los datos del registro de retención que se han leído a otra ubicación y activar un bit de historial DONE. Este segmento también activa un bit de historial ERROR y guarda la palabra STATUS en otra ubicación en caso de error.



Segmento 4: Escribir 64 bits de datos desde MW600-MW607 a las ubicaciones del bit de salida 00017 a 00081 en el esclavo #2.



Segmento 5: Activar un bit de historial DONE cuando haya finalizado la escritura. Si se produce un error, el programa activa un bit de historial ERROR y guarda el código STATUS.

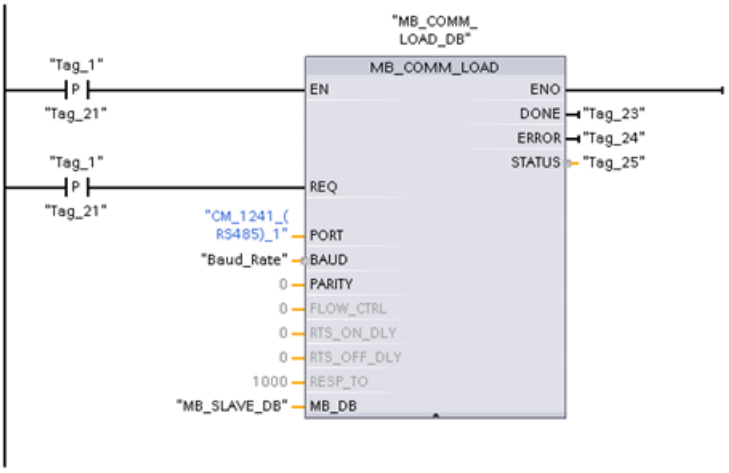


13.9.4.2 Ejemplo: programa de esclavo Legacy Modbus RTU

El MB_COMM_LOAD mostrado a continuación se inicializa cada vez que se habilita "Tag_1".

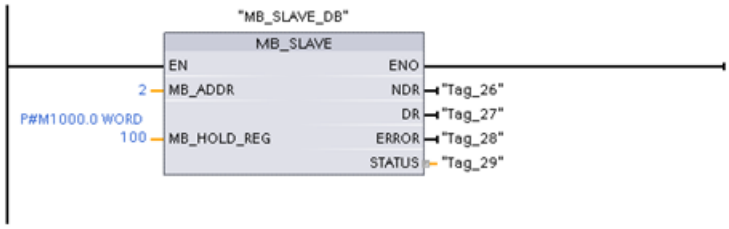
La ejecución de MB_COMM_LOAD de esa forma solo debe hacerse cuando la configuración de puerto serie cambie en tiempo de ejecución, a consecuencia de la configuración de HMI.

Segmento 1: inicializar los parámetros del módulo RS485 cada vez que un dispositivo HMI los cambia.



El MB_SLAVE mostrado a continuación se coloca en un OB cíclico que se ejecuta cada 10 ms. Con esto no se consigue que el esclavo proporcione la respuesta más rápida posible, pero proporciona buen rendimiento a 9600 baudios para mensajes cortos (20 bytes o menos en la petición).

Segmento 2: comprobar las peticiones del maestro Modbus en cada ciclo. El registro de retención Modbus está configurado para 100 palabras comenzando en MW1000.



13.10 Industrial Remote Communication (IRC)

13.10.1 Sinopsis de los CP de Telecontrol

Industrial Remote Communication ofrece un acceso seguro y económico a máquinas ampliamente distribuidas y a aplicaciones de diferentes tamaños. Industrial Remote Communication incluye las siguientes formas de comunicación utilizando módulos CP:

- TeleControl: TeleControl es la conexión de estaciones de proceso (Remote Terminal Units/ RTU) que están repartidas por una amplia área geográfica a uno o varios sistemas centralizados de control de procesos para fines de vigilancia y control. Varios componentes de transmisión distintos en la gama de productos de redes remotas soportan la comunicación remota en diferentes redes públicas y privadas. Los protocolos especiales de TeleControl efectúan intercambios de datos de proceso cíclicos o accionados por eventos, lo que permite controlar con eficiencia todo el proceso.
- TeleService: TeleService es el intercambio de datos con sistemas técnicos distantes (máquinas, plantas, equipos, etc.) para fines de detección de errores, diagnóstico, mantenimiento, reparación u optimización.
- También tiene aplicaciones adicionales para la comunicación remota, por ejemplo vigilancia, aplicaciones de retícula inteligentes y supervisión de las condiciones.

CP de TeleControl para S7-1200

Para aplicaciones de TeleControl están disponibles los siguientes procesadores de comunicaciones, muchos de los cuales también ofrecen acceso al servidor web (Página 853) del S7-1200:

- **CP 1243-1:**
  - Referencia: 6GK7 243-1BX30-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a través de la infraestructura pública (p. ej. DSL) a un centro de control con TeleControl Server Basic (TCSB versión V3)
  - Con la ayuda de la tecnología VPN y el cortafuegos, el CP permite un acceso protegido al S7-1200.
  - El CP puede utilizarse como una interfaz Ethernet adicional de la CPU para la comunicación S7.
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.
- **CP 1243-1 DNP3:**
  - Referencia: 6GK7 243-1JX30-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a centros de control mediante el protocolo DNP3.
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.

- **CP 1243-1 IEC:**
  - Referencia: 6GK7 243-1PX30-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a centros de control mediante el protocolo IEC 60870-5.
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.
- **CP 1243-1 PCC:**
  - Referencia: 6GK7 243-1HX30-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a centros de control mediante Plant Cloud Communication (PCC)
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.
- **CP 1242-7:**
  - Referencia: 6GK7 242-7KX31-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a un centro de control con TeleControl Server Basic utilizando una conexión móvil inalámbrica (GPRS) y la infraestructura pública (DSL).
- **CP 1242-7 GPRS V2:**
  - Referencia: 6GK7 242-7KX31-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a un centro de control con TeleControl Server Basic (TCSB versión V3) utilizando una conexión móvil inalámbrica (GPRS) y la infraestructura pública (DSL).
  - Con la ayuda de la tecnología VPN y el cortafuegos, el CP permite un acceso protegido al S7-1200.
  - El CP puede utilizarse como una interfaz Ethernet adicional de la CPU para la comunicación S7.
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.

- **CP 1243-7 LTE-xx:**
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a un centro de control con TeleControl Server Basic (TCSB versión V3) utilizando una conexión móvil inalámbrica (GPRS) y la infraestructura pública (DSL).
  - Compatibilidad con las siguientes especificaciones móviles inalámbricas: GSM/GPRS, UMTS (G3), LTE
  - Para cubrir países con diferentes especificaciones para conexiones móviles inalámbricas, el CP está disponible en dos variantes:
    - CP 1243-7 LTE-US:
      - Estándar norteamericano
      - Referencia: 6GK7 243-7SX30-0XE0
    - CP 1243-7 LTE-EU:
      - Estándar europeo occidental
      - Referencia: 6GK7 243-7KX30-0XE0
  - Con la ayuda de la tecnología VPN y el cortafuegos, el CP permite un acceso protegido al S7-1200.
  - El CP puede utilizarse como una interfaz Ethernet adicional de la CPU para la comunicación S7.
  - La comunicación entre el CP y la CPU pasa por puntos de datos configurables que acceden a variables PLC.
- **CP 1243-8 IRC:**
  - Referencia: 6GK7 242-8RX30-0XE0
  - Procesador de comunicaciones para conectar SIMATIC S7-1200 a una red ST7, configuración de puntos de datos y VPN.

Nota

Se requiere el software TeleControl Server Basic para aplicaciones de TeleControl en CP distintos del CP 1243-1.

Comunicación segura

El acreditado protocolo SINAUT ST7 o el protocolo DNP3 estandarizado o el protocolo IEC 60870-5 incorpora seguridad a la Industrial Remote Communication (http://w3app.siemens.com/mcms/infocenter/dokumentencenter/sc/ic/InfocenterLanguagePacks/Netzwerksicherheit/6ZB5530-1AP02-0BA4_BR_Network_Security_en_112015.pdf). La solución de TeleControl ofrece amplias medidas para impedir la falsificación y la pérdida de datos. Cada módulo de transmisión tiene una memoria grande para varios miles de frames de datos, lo que permite puentear tiempos de parada en el enlace de transmisión. Las soluciones VPN dedicadas protegen las redes especiales basadas en IP.

El procesador de comunicaciones CP 1243-1 conecta con seguridad el controlador SIMATIC S7-1200 a redes Ethernet. Con su cortafuegos integrado (Stateful Inspection) y las funciones de seguridad del protocolo VPN (IPsec), el procesador de comunicaciones ayuda a proteger las estaciones S7-1200 y las redes de nivel inferior de accesos no autorizados y también la transmisión de datos de manipulaciones y espionaje por medio del cifrado. Asimismo, el CP

puede utilizarse para integrar la estación S7-1200 en el software de centro de control TeleControl Server Basic a través de redes remotas basadas en IP.

13.10.2 Conexión a una red GSM

Comunicación WAN basada en IP mediante GPRS

Empleando el procesador de comunicaciones CP 1242-7, el S7-1200 puede conectarse a las redes GSM. El CP 1242-7 permite la comunicación WAN de estaciones remotas con un centro de control y una comunicación entre estaciones.

La comunicación entre estaciones solo es posible mediante una red GSM. Para la comunicación entre una estación remota y una sala de control, el centro de control debe tener un PC con acceso a Internet.

El CP 1242-7 soporta los servicios siguientes para la comunicación mediante la red GSM:

- GPRS (General Packet Radio Service o servicio general de radio por paquetes)
  El servicio por paquetes para la transmisión de datos "GPRS" se maneja mediante la red GSM.
- SMS (Short Message Service o servicio de mensajería corta)
  El CP 1242-7 puede recibir y enviar mensajes SMS. El interlocutor de la comunicación puede ser un teléfono móvil o un S7-1200.

El CP 1242-7 es adecuado para un uso industrial en todo el mundo y soporta las bandas de frecuencia siguientes:

- 850 MHz
- 900 MHz
- 1800 MHz
- 1900 MHz

Requisitos

El equipo usado en las estaciones o el centro de control depende de la aplicación en concreto.

- Para la comunicación con o por medio de una sala de control central, el centro de control requiere un PC con acceso a Internet.
- Aparte del equipo de la estación, una estación S7-1200 remota con un CP 1242-7 debe cumplir los requisitos siguientes para ser capaz de comunicarse mediante la red GSM:
  - Un contrato con un proveedor de red GSM adecuado
    Si se usa GPRS, el contrato debe permitir el uso del servicio GPRS.
    Si solo se dispone de una comunicación directa entre las estaciones mediante una red GSM, el proveedor de la red GSM debe asignar una dirección IP fija a los CPs. En este caso, la comunicación entre estaciones no es mediante el centro de control.
  - La tarjeta SIM perteneciente al contrato
    La tarjeta SIM se inserta en el CP 1242-7.
  - Disponibilidad local de una red GSM en el rango de la estación

13.10.3 Aplicaciones del CP 1242-7

El CP 1242-7 puede usarse para las aplicaciones siguientes:

Aplicaciones de telecontrol

- Enviar mensajes por SMS
  Mediante el CP 1242-7, la CPU de una estación S7-1200 remota puede recibir mensajes SMS de la red GSM o enviar mensajes por SMS a un teléfono móvil configurado o un S7-1200.
- Comunicación con un centro de control
  Las estaciones S7-1200 remotas se comunican mediante la red GSM e Internet con un servidor de telecontrol en la estación maestra. Para la transferencia de datos mediante GPRS, la aplicación "TELECONTROL SERVER BASIC" se instala en el servidor de telecontrol en la estación maestra. El servidor de telecontrol se comunica con un sistema de control central de nivel superior mediante la función integrada de servidor OPC.
- Comunicación entre las estaciones S7-1200 mediante una red GSM
  La comunicación entre estaciones remotas con un CP 1242-7 puede manejarse de dos formas diferentes:
  - Comunicación entre estaciones mediante una estación maestra
    En esta configuración, una conexión segura permanente entre estaciones S7-1200 que se comunican entre sí y el servidor de telecontrol se establece en la estación maestra. La comunicación entre estaciones se realiza mediante el servidor de telecontrol. El CP 1242-7 funciona en modo "Telecontrol".
  - Comunicación directa entre estaciones
    Para una comunicación directa entre estaciones sin el desvío por medio de la estación maestra se usan tarjetas SIM con una dirección IP fija que permiten a las estaciones comunicarse entre sí directamente. Los servicios de comunicación y funciones de seguridad posibles (por ejemplo VPN) dependen de lo que ofrece el proveedor de red. El CP 1242-7 funciona en modo "GPRS directo".

TeleService vía GPRS

Una conexión TeleService puede establecerse entre una estación de ingeniería con STEP 7 y una estación S7-1200 remota con un CP 1242-7 a través de la red GSM e Internet. La conexión parte de la estación de ingeniería y pasa por un servidor de telecontrol o una pasarela de TeleService que actúa de intermediario redireccionando tramas y estableciendo la autorización. Estos PCs usan las funciones de la aplicación "TELECONTROL SERVER BASIC".

Es posible utilizar la conexión de TeleService para los fines siguientes:

- Descargar la configuración o los datos de programa del proyecto de STEP 7 a la estación
- Consultar datos de diagnóstico en la estación

13.10.4 Otras propiedades del CP 1242-7

Otros servicios y funciones del CP 1242-7

- Sincronización en tiempo real del CP mediante Internet
  Es posible ajustar el tiempo en el CP del siguiente modo:
  - En el modo "Telecontrol" la hora se transfiere por el servidor de telecontrol. El CP la usa para ajustar su hora.
  - En el modo "GPRS directo" el CP puede solicitar la hora usando SNTP.

  Para sincronizar la hora de la CPU es posible leer la hora actual del CP empleando un bloque.
- Respaldo intermedio de mensajes que deben enviarse si hay problemas de conexión
- Disponibilidad aumentada gracias a la opción de conectar a un servidor de telecontrol sustituto
- Registro del volumen de datos
  El volumen de datos transferidos se registra y puede evaluarse para fines específicos.

13.10.5 Más información

Los manuales del CP, la documentación relacionada y los documentos de información del producto ofrecen información detallada:

- CP 1242-7 (http://support.automation.siemens.com/WW/view/es/45605894)
- CP 1243-7 LTE (https://support.industry.siemens.com/cs/ww/es/ps/15924)
- CP 1243-1 DNP3 (https://support.industry.siemens.com/cs/ww/es/ps/15938)
- CP 1243-8 IRC (https://support.industry.siemens.com/cs/ww/es/ps/21162)
- CP 1243-1 IEC (https://support.industry.siemens.com/cs/ww/es/ps/15942)
- Actualizaciones de firmware disponibles (https://support.industry.siemens.com/cs/ww/es/view/109482530/en)

13.10.6 Accesorios

Antena ANT794-4MR GSM/GPRS

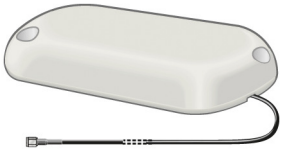
Las antenas siguientes están disponibles para ser usadas en redes GSM/GPRS y pueden instalarse tanto en el interior como en el exterior:

- Antena cuatribanda ANT794-4MR (http://support.automation.siemens.com/WW/view/es/23119005)



Nombre abreviado	Referencia	Explicación
ANT794-4MR	6NH9 860-1AA00	Antena cuatribanda (900, 1800/1900 MHz, UMTS); resistente a la intemperie para su uso en interiores y exteriores; cable de conexión de 5 m conectado permanentemente a la antena; conector SMA que incluye un soporte de instalación, tornillos y conectores de pared

- Antena plana ANT794-3M



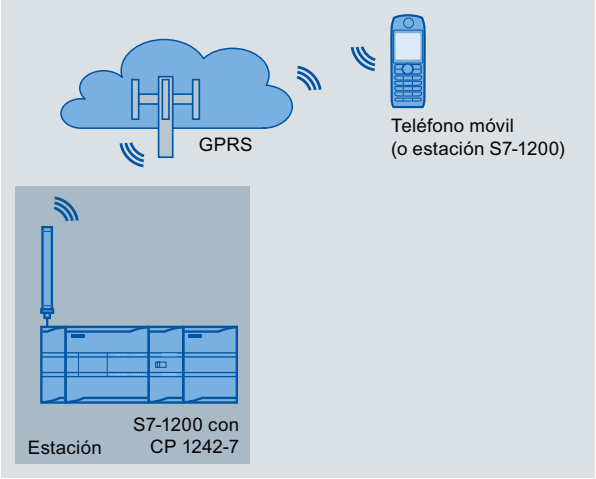
Nombre abreviado	Referencia	Explicación
ANT794-3M	6NH9 870-1AA00	Antena plana (900, 1800/1900 MHz); resistente a la intemperie para su uso en interiores y exteriores; cable de conexión de 1,2 m conectado permanentemente a la antena; conector SMA que incluye un soporte adhesivo, posibilidad de montaje con tornillos

Las antenas deben pedirse por separado.

13.10.7 Ejemplos de configuración de telecontrol

En la parte inferior encontrará varios ejemplos de configuración de estaciones con un CP 1242-7.

Enviar mensajes por SMS



Un SIMATIC S7-1200 con un CP 1242-7 puede enviar mensajes por SMS a un teléfono móvil o a una estación S7-1200 configurada.

Telecontrol mediante un centro de control

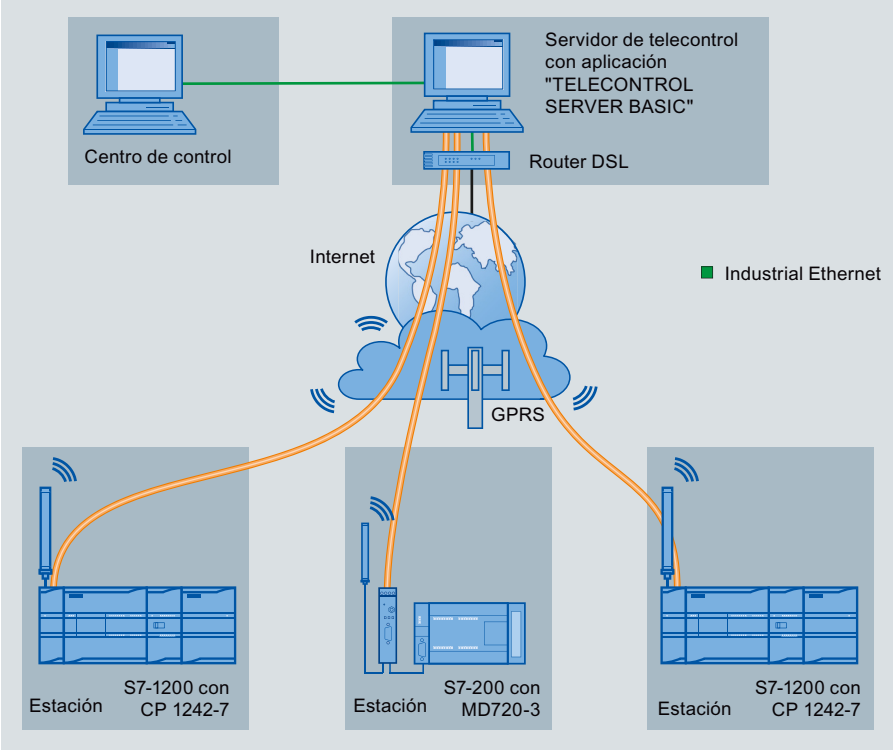


Figura 13-29 Comunicación entre las estaciones S7-1200 y un centro de control

En aplicaciones de telecontrol, las estaciones SIMATIC S7-1200 con un CP 1242-7 se comunican con un centro de control a través de la red GSM e Internet. La aplicación "TELECONTROL SERVER BASIC" (TCSB) se instala en el servidor de telecontrol en la estación maestra. Como resultado se producen las aplicaciones siguientes:

- Comunicación de telecontrol entre la estación y el centro de control
  En este caso, los datos del campo son enviados por las estaciones al servidor de telecontrol en la estación maestra a través de la red GSM e Internet. El servidor de telecontrol se usa para supervisar estaciones remotas.
- Comunicación entre una estación y una sala de control con un cliente OPC
  Igual que en el primer caso, la estación se comunica con el servidor de telecontrol. Mediante su servidor OPC integrado, el servidor de telecontrol intercambia datos con el cliente OPC de la sala de control.
  El cliente OPC y el servidor de telecontrol pueden encontrase en un mismo equipo, por ejemplo cuando TCSB se instala en un PC del centro de control con WinCC.
- Comunicación entre estaciones mediante un centro de control
  La comunicación entre estaciones es posible con estaciones S7 equipadas con un CP 1242-7. Para permitir la comunicación entre estaciones, el servidor de telecontrol transmite los mensajes de la estación emisora a la estación receptora.

Comunicación directa entre estaciones

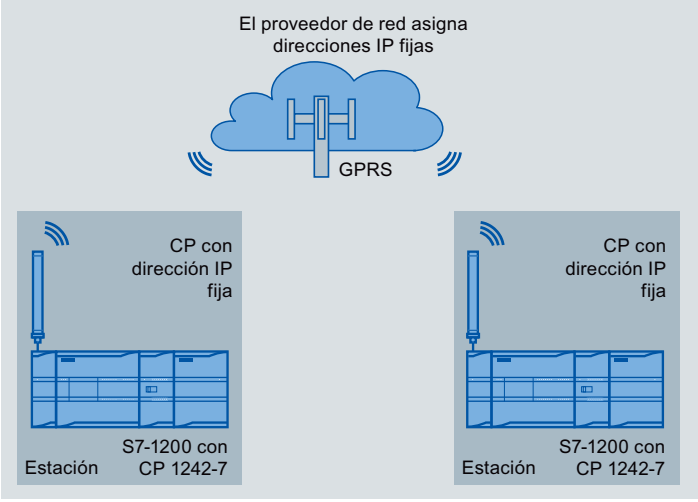


Figura 13-30 Comunicación directa entre dos estaciones S7-1200

En esta configuración, dos estaciones SIMATIC S7-1200 se comunican directamente entre sí usando el CP 1242-7 a través de la red GSM. Cada CP 1242-7 tiene asignada una dirección IP fija. El servicio relevante del proveedor de red GSM debe permitirlo.

TeleService vía GPRS

En TeleService vía GPRS, una estación de ingeniería en la que se ha instalado STEP 7 se comunica a través de la red GSM e Internet con el CP 1242-7 en el S7-1200.

Puesto que un cortafuegos suele estar cerrado para peticiones de conexión desde el exterior, se requiere una estación de conmutación entre la estación remota y la estación de ingeniería. Esta estación de conmutación puede ser un servidor de telecontrol o, si no se dispone de un servidor de telecontrol en la configuración, una pasarela de TeleService.

TeleService con servidor de telecontrol

La conexión se lleva a cabo mediante el servidor de telecontrol.

- La estación de ingeniería y el servidor de telecontrol están conectados vía Intranet (LAN) o Internet.
- El servidor de telecontrol y la estación remota están conectados vía Internet y vía red GSM.

La estación de ingeniería y el servidor de telecontrol también pueden ser el mismo equipo; en otras palabras, STEP 7 y TCSB están instalados en el mismo PC.

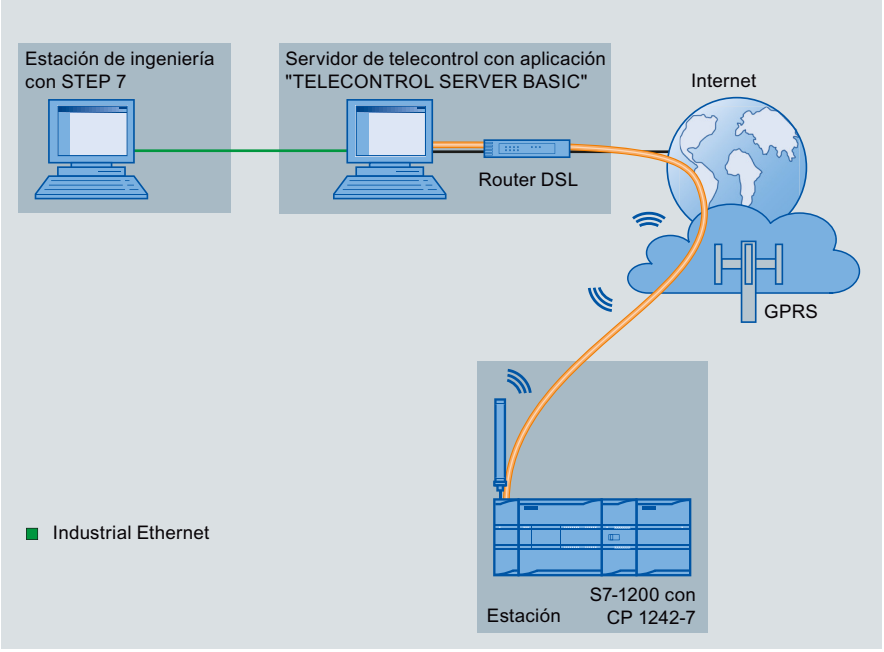


Figura 13-31 TeleService vía GPRS en una configuración con servidor de telecontrol

TeleService sin servidor de telecontrol

La conexión se lleva a cabo mediante la pasarela de TeleService.

La conexión entre la estación de ingeniería y la pasarela de TeleService puede ser local vía LAN o vía Internet.

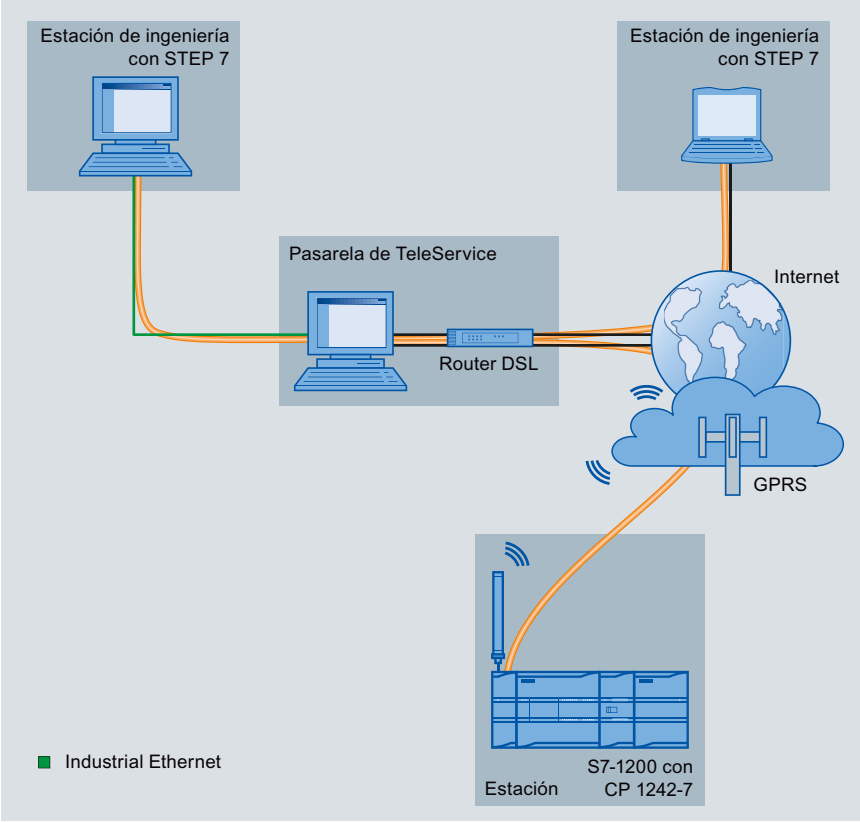


Figura 13-32 TeleService vía GPRS en una configuración con pasarela de TeleService

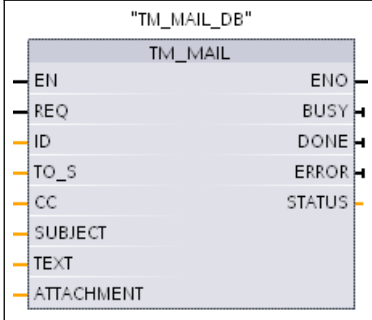
Comunicación TeleService (correo electrónico SMTP)

14

14.1 Instrucción TM_Mail (Transmitir e-mail)

Tabla 14-1 Instrucción TM_MAIL

KOP / FUP	SCL	Descripción

	`"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_,);`
		La instrucción TM_MAIL envía un mensaje de correo electrónico mediante SMTP (Simple Mail Transfer Protocol) sobre TCP/IP a través de la conexión Industrial Ethernet de la CPU. Si no se dispone de conectividad Ethernet con Internet, se puede usar un adaptador Teleservice opcional para conectarse con la red telefónica conmutada. TM_MAIL se ejecuta asíncronamente y la tarea abarca varias llamadas de TM_MAIL. Al llamar a TM_MAIL, se debe asignar un DB de instancia. **No es necesario ajustar el atributo remanente del DB de instancia**. Esto garantiza que el DB de instancia sea inicializado en la transición de la CPU de STOP a RUN y que se puede disparar una nueva operación TM_MAIL.

[1] STEP 7 crea el DB de instancia automáticamente al insertar la instrucción.

El envío de un correo electrónico se inicia con un flanco ascendente, de 0 a 1, en el parámetro de entrada REQ. La tabla siguiente muestra la relación entre los parámetros BUSY, DONE y ERROR. Es posible vigilar el progreso de ejecución de TM_MAIL y detectar que la misma haya finalizado evaluando estos parámetros en llamadas consecutivas.

Los parámetros de salida DONE, ERROR, STATUS, y SFC_STATUS solo son válidos durante un ciclo, cuando el estado del parámetro de salida BUSY cambia de 1 a 0. La lógica del programa debe guardar temporalmente los valores de estado de la salida, de modo que sea posible detectar cambios de estado en posteriores ciclos de ejecución.

Nota

TM_MAIL envía un mensaje de correo electrónico vía TCP/IP a través de la interfaz Ethernet de la CPU. Para enviar un mensaje de correo electrónico a través de la interfaz del CP (con o sin SSL), utilice la instrucción Instrucción TMAIL_C (Envía un correo electrónico utilizando la interfaz Ethernet de la CPU) (Página 682).

Tabla 14-2 Interacción de los parámetros Done, Busy y Error

DONE	BUSY	ERROR	Descripción
Irrelevante	1	Irrelevante	Tarea en curso.
1	0	0	La tarea se ha ejecutado correctamente.
0	0	1	La tarea se ha finalizado con un error. Para averiguar la causa del error, consulte el parámetro STATUS.
0	0	0	Ninguna tarea en curso

Si la CPU pasa a STOP mientras está activa la instrucción TM_MAIL, se cierra la comunicación con el servidor de correo electrónico. La comunicación con el servidor de correo electrónico también se pierde si se producen problemas de comunicación con la CPU en el bus Industrial Ethernet. En estos casos, se suspende el proceso de transmisión y el correo electrónico no llega al destinatario.

ATENCIÓN

Modificación de programas de usuario

El borrado y la sustitución de bloques de programas, las llamadas a TM_MAIL o las llamadas a los DB de instancia de TM_MAIL pueden romper las conexiones de los bloques de programas. Si no logra mantener conectados los bloques de programa, las funciones de comunicación TPC/IP pueden entrar en un estado indefinido, y posiblemente acaben provocando daños materiales. Después de transferir un bloque de programa modificado, debería reiniciar la CPU (caliente) o ejecutar un arranque en frío.

Para evitar romper las conexiones de los bloques de programas, solo debe cambiar las partes del programa de usuario que afectan directamente a las llamadas de TM_MAIL en los casos siguientes:

- La CPU está en STOP
- No se envió ningún correo electrónico (REQ y BUSY = 0)

Coherencia de datos

El parámetro de entrada ADDR_MAIL_SERVER se lee una vez iniciada la operación. Un valor nuevo no se hará efectivo hasta que la operación actual haya finalizado y se haya iniciado una nueva operación TM_MAIL.

Por el contrario, los parámetros WATCH_DOG_TIME, TO_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME y PASSWORD se leen durante la ejecución de TM_MAIL y solo se pueden modificar cuando la tarea ha finalizado (BUSY = 0).

Conexión telefónica: Configurar los parámetros de IE del adaptador de TS

Es necesario configurar los parámetros de IE del adaptador de TS para llamadas salientes con el fin de conectarse con el servidor de acceso telefónico de su proveedor de servicios de Internet. Si activa el atributo "sobre petición" para la llamada, la conexión solo se establecerá cuando se vaya a transmitir un correo electrónico. Para una conexión analógica vía módem se requiere más tiempo para el proceso de conexión (aprox. un minuto más). Hay que incluir el tiempo extra en el valor de WATCH_DOG_TIME.

Tabla 14-3 Tipos de datos para los parámetros

Parámetro y tipo		Tipos de datos	Descripción
REQ	IN	Bool	Un cambio de señal low a high (flanco ascendente) lanza la operación.
ID	IN	Int	Identificador de conexión: Véase el parámetro ID de las instrucciones TCON, TDISCON, TSEND y TRCV. Se debe usar un número que no se utiliza para instancias adicionales de esta instrucción en el programa de usuario.
TO_S	IN	String	Direcciones de destinatarios: datos STRING con una longitud máxima de 240 caracteres.
CC	IN	String	Copia CC para direcciones de destinatarios (opcional): datos STRING con una longitud máxima de 240 caracteres.
SUBJECT	IN	String	Asunto del correo electrónico: datos STRING con una longitud máxima de 240 caracteres.
TEXT	IN	String	Mensaje de texto del correo electrónico (opcional): datos STRING con una longitud máxima de 240 caracteres. Si este parámetro es una cadena vacía, el correo electrónico se enviará sin texto de mensaje.
ATTACHMENT	IN	Variant	Puntero hacia los datos adjuntos del correo electrónico: datos byte, palabra o palabra doble con una longitud máxima de 65534 bytes. Si no se asigna ningún valor, el correo electrónico se enviará sin datos adjuntos.
DONE	OUT	Bool	• 0 - Tarea no iniciada aún o en ejecución. • 1 - La tarea se ha ejecutado sin errores.
BUSY	OUT	Bool	• 0 - Ninguna operación en curso • 1- Operación en curso
ERROR	OUT	Bool	El bit ERROR es = 1 durante un ciclo tras haberse finalizado la última petición con un error. El valor del código de error en la salida STATUS solo es válido durante un ciclo en que ERROR = 1.
STATUS	OUT	Word	Valor de retorno o información de error de la instrucción TM_MAIL.

Parámetro y tipo		Tipos de datos	Descripción
ADDR_MAIL_SERVER	[1] Static	DWord	Dirección IP del servidor de correo electrónico: Se debe asignar cada fragmento de la dirección IP en forma de octeto de dos caracteres hexadecimales de 4 bits. Si el fragmento de la dirección IP = valor decimal 10, que equivale al valor hexadecimal A, se debe introducir "0A" para ese octeto. Ejemplo: Dirección IP = 192.168.0.10 ADDR_MAIL_SERVER = DW#16#C0A8000A, donde: • 192 = 16#C0, • 168 =16#A8 • 0 = 16#00 • 10 = 16#0A
WATCH_DOG_TIME	[1] Static	Time	Tiempo máximo permitido para TM_MAIL para completar todo el proceso SMTP desde la iniciación de la conexión con el SMTP hasta el final de la transmisión SMTP. Si se excede este tiempo, la ejecución de TM_MAIL finalizará con un error. El retardo real hasta que finalice TM_MAIL y se emita el error puede exceder el WATCH_DOG_TIME debido a que la operación de desconexión requiere tiempo adicional. Inicialmente debe ajustarse un tiempo de 2 minutos. Este tiempo puede ser mucho más corto para una conexión telefónica ISDN.
USERNAME	[1] Static	String	Nombre de usuario de la cuenta de correo: datos STRING con una longitud máxima de 180 caracteres.
PASSWORD	[1] Static	String	Contraseña del servidor de correo: datos STRING con una longitud máxima de 180 caracteres.
FROM	[1] Static	String	Dirección del emisor: STRING con una longitud máxima de 240 caracteres.
SFC_STATUS	[1] Static	Word	Código de condición de ejecución de los bloques de comunicación invocados

[1] Los valores de esos parámetros no se modifican en cada llamada de TM_MAIL. Los valores se asignan en el bloque de datos de instancia de TM_MAIL y solo se referencian una vez, en la primera llamada a TM_MAIL.

Autenticación SMTP

TM_MAIL soporta el método de autenticación SMTP AUTH LOGIN. Para más información sobre ese método de autenticación, consulte el manual del servidor de correo o la página web de su proveedor de servicios de Internet.

El método de autenticación AUTH LOGIN utiliza los parámetros TM_MAIL USERNAME y PASSWORD para conectarse al servidor de correo. El nombre de usuario y la contraseña deben definirse previamente en una cuenta de correo electrónico en un servidor de correo electrónico.

Si no se asigna ningún valor al parámetro USERNAME, el método de autenticación AUTH LOGIN no se usa y el correo electrónico se envía sin autenticación.

Parámetros TO_S:, CC: y FROM :

Los parámetros TO_S:, CC: y FROM: son cadenas, como se indica en los ejemplos siguientes:

TO: <wenna@mydomain.com>, <ruby@mydomain.com>,

CC: <admin@mydomain.com>, <judy@mydomain.com>,

FROM: <admin@mydomain.com>

Las siguientes reglas deben usarse al introducir esas cadenas de caracteres:

● Deben introducirse los caracteres "TO:", "CC:" y "FROM:", incluido el carácter de dos puntos.

● Antes de cada dirección debe introducirse un espacio y un corchete angular de apertura "<". Por ejemplo, debe haber un carácter de espacio entre "TO:" y <dirección de correo electrónico>.

● Después de cada dirección debe introducirse un corchete angular de cierre ">".

● Para las direcciones de TO_S: y CC: debe introducirse una coma "," después de cada dirección. Por ejemplo, "TO: <email address>," necesita una coma tras la única dirección de correo electrónico.

● Solo se puede utilizar una dirección de correo electrónico para la entrada FROM:, sin coma al final.

Debido al modo run-time y al uso de memoria, no se ejecuta una prueba de sintaxis para los parámetros de TM_MAIL TO_S:, CC: y FROM:. Si no se observan estrictamente las reglas de formato anteriores, la transacción del servidor de correo electrónico SMTP fallará.

Parámetros STATUS y SFC_STATUS

Los códigos de condición de ejecución que devuelve TM_MAIL se pueden clasificar del siguiente modo:

- W#16#0000: operación de TM_MAIL finalizada correctamente
- W#16#7xxx: estado de la operación de TM_MAIL
- W#16#8xxx: error en una llamada interna de un dispositivo de comunicación o del servidor de correo electrónico

La tabla siguiente muestra los códigos de condición de ejecución de TM_MAIL con excepción de los códigos de error de módulos de comunicación llamados internamente.

Nota

Requisitos del servidor de correo electrónico

TM_MAIL solo puede comunicarse con un servidor de correo electrónico utilizando SMTP mediante el puerto 25. El número de puerto asignado no se puede cambiar.

La mayoría de departamentos de IT y servidores de correo electrónico externos bloquean el puerto 25 para evitar que un PC infectado con un virus se convierta en un generador de correos electrónicos no autorizado.

Se puede conectar con un servidor de correo interno mediante SMTP y hacer que el servidor interno gestione las mejoras de seguridad actuales necesarias para reenviar correos electrónicos a través de Internet a un servidor de correo externo.

Ejemplo: configuración de servidor de correo electrónico interno

Si se usa Microsoft Exchange como servidor de correo interno, se puede configurar el servidor para que permita acceso SMTP desde la dirección IP asignada al PLC S7-1200. Configure la consola de administración de Exchange: Configuración del servidor > Transporte de concentradores. > Conectores de recepción > Relé. En la pestaña Red, hay un cuadro denominado "Recibir correo desde servidores remotos que tengan estas direcciones IP". Aquí se debe poner la dirección IP del PLC que esté ejecutando la instrucción TM_MAIL. No se requiere autenticación para este tipo de conexión con un servidor de Microsoft Exchange interno.

Configuración del servidor de correo electrónico

TM_MAIL solo puede usar un servidor de correo electrónico que permita la comunicación por el puerto 25, SMTP y autenticación AUTH LOGIN (opcional).

Configure una cuenta de servidor de correo electrónico compatible para que acepte el inicio de sesión SMTP remoto. Edite el DB de instancia para TM_MAIL para poner las cadenas de caracteres de TM_MAIL USERNAME y PASSWORD que se usan para autenticar la conexión con su cuenta de correo electrónico.

Tabla 14-4 Códigos de condición

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Descripción
0000	-	La operación de TM_MAIL ha finalizado sin errores. Este código cero de STATUS no garantiza que un correo electrónico realmente haya sido enviado (consulte el primer punto de la nota que sigue a esta tabla).
7001	-	TM_MAIL está activa (BUSY = 1).
7002	7002	TM_MAIL está activa (BUSY = 1).
8xxx	xxxx	La operación de TM_MAIL ha finalizado con un error en las llamadas internas de la instrucción de comunicación. Para más información sobre el parámetro SFC_STATUS, consulte la descripción del parámetro STATUS de las instrucciones de comunicación de usuario abierta PROFINET subyacente.
8010	xxxx	Fallo de conexión: para más información sobre el parámetro SFC_STATUS, consulte el parámetro STATUS de la instrucción TCON.
8011	xxxx	Error al transmitir datos: para más información sobre el parámetro SFC_STATUS, consulte el parámetro STATUS de la instrucción TSEND.
8012	xxxx	Error al recibir datos: para más información sobre el parámetro SFC_STATUS, consulte las descripciones del parámetro STATUS de la instrucción TRCV.
8013	xxxx	Fallo de conexión: para más información sobre cómo evaluar el parámetro SFC_STATUS, consulte las descripciones del parámetro STATUS de las instrucciones TCON y TDISCON.
8014	-	Fallo de conexión: es posible que haya introducido una dirección IP incorrecta para el servidor de correo electrónico (ADDR_MAIL_SERVER) o bien un tiempo muy corto (WATCH_DOG_TIME) para la conexión. Asimismo, es posible que la CPU no tenga conexión con la red o que su configuración sea incorrecta.
8015	-	Puntero no válido para el parámetro ATTACHMENT: Use un puntero de variante con una asignación de longitud y tipo de datos. Por ejemplo, "P#DB.DBX0.0" es incorrecto y "P#DB.DBX0.0 byte 256" es correcto.
82xx, 84xx, 85xx	-	El mensaje de error viene del servidor de correo electrónico y corresponde al número de error "8" del protocolo SMTP. Consulte el segundo punto de la nota que sigue a esta tabla.

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Descripción
8450	-	La operación no se ejecuta: El buzón de correo no está disponible; inténtelo más tarde.
8451	-	Operación interrumpida: Error local de procesamiento; inténtelo más tarde.
8500	-	Error de sintaxis en el comando: la causa puede ser que el servidor de correo electrónico no soporte el proceso de autenticación de LOGIN. Compruebe los parámetros de TM_MAIL. Intente enviar un correo electrónico sin autenticación. Intente reemplazar el parámetro USERNAME por una cadena vacía.
8501	-	Error de sintaxis: Parámetro o argumento incorrecto; es posible que haya tecleado una dirección incorrecta en los parámetros TO_S o CC.
8502	-	Comando desconocido o no implementado: verifique las entradas, especialmente el parámetro FROM. Posiblemente esté incompleto y usted haya omitido los caracteres "@" o ".".
8535	-	La autenticación SMTP está incompleta. Es posible que haya introducido un nombre de usuario o una contraseña incorrecta.
8550	-	No es posible acceder al servidor de correo o bien el usuario no dispone de los derechos necesarios. Puede que haya introducido un nombre de usuario o una contraseña incorrecta o su servidor de correo electrónico no admite un acceso mediante inicio de sesión. Otra causa de este error puede ser una entrada errónea del nombre de dominio después del carácter "@" en los parámetros TO_S o CC.
8552	-	Operación interrumpida: se ha excedido el tamaño de la memoria asignada; inténtelo más tarde.
8554	-	Fallo de transmisión: inténtelo más tarde.

Nota

Posibles errores de transmisión de correo electrónico no reportados

- Una entrada incorrecta de una dirección de destinatario no genera un error STATUS para TM_MAIL. En este caso, no hay garantía de que destinatarios adicionales (con direcciones de correo electrónico correctas) recibirán el correo electrónico.
- Encontrará más información sobre códigos de error SMTP en Internet o en la documentación sobre errores del servidor de correo. También es posible leer el último mensaje de error del servidor de correo. El mensaje de error se guarda en el parámetro buffer1 del DB de instancia para TM_MAIL.

Herramientas online y diagnóstico 15

15.1 LEDs de estado

La CPU y los módulos de E/S utilizan LED para indicar el estado operativo del módulo o de las E/S.

LEDs de estado de la CPU

La CPU incorpora los siguientes indicadores de estado:

- STOP/RUN
  - Luz amarilla permanente indica el estado operativo STOP
  - Luz verde permanente indica el estado operativo RUN
  - Si parpadea (alternando entre verde y amarillo), significa que la CPU está en estado operativo ARRANQUE
- ERROR
  - Luz roja intermitente señala un error, como un error interno de la CPU, un error de la Memory Card o un error de configuración (los módulos no coinciden)
  - Si parpadea en rojo durante tres segundos significa que hay un error no recurrente. Un ejemplo es que el reloj de tiempo real (RTC) se ajuste a la hora predeterminada debido a una pérdida de alimentación.
  - Estado defectuoso:
    - Luz roja permanente indica que hay un fallo de hardware
    - Todos los LED parpadean si el firmware detecta un defecto
- El LED MAINT (mantenimiento) parpadea cuando se inserta una Memory Card. Ejecute una desconexión/conexión (Power OFF - Power ON) de la CPU. La CPU pasa entonces a estado operativo STOP. Tras cambiar la CPU a estado operativo STOP, realice una de las funciones siguientes para iniciar la evaluación de la Memory Card:
  - Cambiar la CPU a estado operativo RUN
  - Realizar un borrado total (MRES)
  - Desconectar y volver a conectar la alimentación de la CPU

También se puede utilizar la instrucción LED (Página 427) para determinar el estado de los LEDs.

Tabla 15-1 LEDs de estado de la CPU

Descripción	STOP/RUN Amarillo / Verde	ERROR Rojo	MAINT Amarillo
Alimentación desconectada	Apagado	Apagado	Apagado
Arranque, autotest o actualización de firmware	Parpadeo (alternando entre amarillo y verde)	-	Apagado
Estado operativo STOP	Encendido (amarillo)	-	-

Descripción	STOP/RUN Amarillo / Verde	ERROR Rojo	MAINT Amarillo
Estado operativo RUN	Encendido (verde)	-	-
Desenchufar la Memory Card	Encendido (amarillo)	-	Parpadeo
Error	Encendido (amarillo o verde)	Parpadeo	-
Mantenimiento solicitado • E/S forzadas • Es necesario sustituir las baterías (si se ha instalado la Battery Board)	Encendido (amarillo o verde)	-	Encendido
Hardware averiado	On (amarillo)	Encendido	Apagado
Test de LEDs o firmware de la CPU defectuoso	Parpadeo (alternando entre amarillo y verde)	Parpadeo	Parpadeo
Versión desconocida o incompatible de la configuración de CPU	Encendido (amarillo)	Parpadeo	Parpadeo

La CPU incorpora asimismo dos LEDs que indican el estado de la comunicación PROFINET. Abra la tapa del bloque de terminales inferior para ver los LEDs PROFINET.

- Link (verde) se enciende para indicar una conexión correcta
- Rx/Tx (amarillo) se enciende para indicar la actividad de transmisión

La CPU y todos los módulos de señales (SM) digitales incorporan un LED I/O Channel para cada una de las entradas y salidas digitales. El LED I/O Channel (verde) se enciende o apaga para indicar el estado de la entrada o salida en cuestión.

Error "Versión de la descripción de configuración de CPU desconocida o incompatible"

El búfer de diagnóstico puede notificar el error "Versión de la descripción de configuración de CPU desconocida o incompatible", que puede producirse de una de las siguientes formas:

- Al intentar descargar un programa S7-1200 V3.0 en una CPU S7-1200 V4.x.
- Al intentar descargar un proyecto cuando la protección de datos de configuración confidenciales del PLC (Página 161) es diferente entre la CPU y el proyecto

Si ha llegado a este estado usando una tarjeta de transferencia de programa (Página 123) de una versión no válida, proceda del siguiente modo para la recuperación:

1. Retire la tarjeta de transferencia del programa.
2. Realice una transición de STOP a RUN.
3. Realice un borrado total (MRES) o una desconexión y reconexión.

Si ha llegado a este estado por una descarga de programa no válida, restablezca la configuración de fábrica de la CPU (Página 1206).

Si ha llegado a este estado por una discordancia entre la CPU y el proyecto para la protección de datos de configuración confidenciales del PLC, utilice las herramientas Online y diagnóstico (Página 1205) para establecer la contraseña de la CPU para la protección de datos de configuración confidenciales del PLC como contraseña para el proyecto o bórrela de la CPU online.

Después de recuperar la CPU de la condición de error, puede descargar un programa válido.

Comportamiento del S7-1200 tras un error crítico

Si el firmware de la CPU detecta un error grave, intentará ejecutar un rearranque en modo de defecto y, si se realiza correctamente, indicará el modo defectuoso con los LED STOP/RUN, ERROR y MAINT parpadeando constantemente. El programa de usuario y la configuración del hardware no se cargan tras un rearranque en estado Defecto.

Si la CPU completa correctamente el rearranque en el estado Defecto, la CPU realizará estas acciones:

- Pone a cero las salidas de CPU y de Signal Board
- Pone las salidas de los módulos de señales del rack central y las E/S de la periferia al ajuste definido en "Reacción a STOP de la CPU" en la configuración de dispositivo de las salidas digitales del módulo

Si el rearranque en el modo Defecto falla (por ejemplo debido a un error de hardware), los LED STOP y ERROR estarán encendidos y el LED MAINT estará apagado.

⚠ ADVERTENCIA

No se puede garantizar el funcionamiento en un estado defectuoso.

Los dispositivos de control pueden fallar y provocar condiciones no seguras, causando a su vez reacciones inesperadas de los equipos controlados. Las reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Utilice una función de parada de emergencia, dispositivos de protección electromecánicos y otras medidas redundantes de seguridad que sean independientes del PLC.

LED de estado de un módulo de señales (SM)

Además, todo SM digital incorpora un LED DIAG que indica el estado del módulo:

- Verde indica que el módulo está operativo
- Rojo indica que el módulo está averiado o no operativo

Todo SM analógico incorpora un LED I/O Channel para cada una de las entradas y salidas analógicas.

- Verde indica que el canal se ha configurado y está activo
- Rojo indica una condición de error de la entrada o salida analógica en cuestión

Además, todo SM analógico incorpora un LED DIAG que indica el estado del módulo:

- Verde indica que el módulo está operativo
- Rojo indica que el módulo está averiado o no operativo

El SM detecta la presencia o ausencia de alimentación del módulo (alimentación de campo en caso necesario).

Tabla 15-2 LED de estado de un módulo de señales (SM)

Descripción	DIAG (Rojo/verde)	I/O Channel (Rojo/verde)
Alimentación de campo desconectada *	Rojo intermitente	Rojo intermitente
No se ha configurado o se está actualizando	Verde intermitente	Apagado
Módulo configurado sin errores	Encendido (verde)	Encendido (verde)
Condición de error	Rojo intermitente	-
Error de E/S (con diagnóstico habilitado)	-	Rojo intermitente
Error de E/S (con diagnóstico inhibido)	-	Encendido (verde)

* El estado solo está soportado en módulos de señales analógicas.

Diagnóstico de módulo analógico

Los módulos analógicos tienen múltiples diagnósticos en función del módulo y el tipo de canal. Es posible habilitar y deshabilitar estos diagnósticos por separado para cada módulo y canal desde TIA Portal, en la configuración de dispositivos del proyecto o en las propiedades generales del módulo.

Error de módulo

Los fallos de alimentación se notifican del siguiente modo:

Fallo de alimentación	Error notificado
El módulo analógico con un error de diagnóstico de alimentación notifica:	Rebase por exceso: 32767 para todos los canales de entrada
	Diagnóstico de falta de alimentación, si está habilitado para módulos de salida

Error de tipo de canal

Este diagnóstico puede habilitarse por separado para cada canal por tipo y canal. Consulte la tabla inferior.

Tipo de canal	Error notificado
Entrada de tensión	Rebase por exceso: 32767
	Rebase por defecto: -32768
Entrada de corriente (de 0 a 20 mA)	Rebase por exceso: 32767
	Rebase por defecto: -32768
Entrada de corriente (de 4 a 20 mA) (para entrada < 1,185 mA)	Rotura de hilo: 32767
	Rebase por exceso: 32767
Salida de tensión (para salida > 0,5 V)	Diagnóstico de cortocircuito, si está habilitado
Salida de corriente (para salida > 1,0 mA)	Diagnóstico de circuito abierto, si está habilitado
Entrada RTD	Rotura de hilo: 32767
	Rebase por exceso: 32767
	Rebase por defecto: -32768

Tipo de canal	Error notificado
Entrada de resistor	Rotura de hilo: 32767
	Rebase por exceso: 32767
Entrada de termopar	Rotura de hilo: 32767
	Rebase por exceso: 32767
	Rebase por defecto: -32768

Un módulo de entradas analógicas con un error de diagnóstico en cualquier canal devuelve 32767 o -32768 en ese canal aunque el diagnóstico no esté habilitado. Los canales de entrada analógica devuelven 32767 cuando están desactivados.

Los módulos de entradas analógicas pueden tener errores de diagnóstico en más de un canal simultáneamente (múltiples errores). Cuando esto ocurre, solo se se notifica a la CPU el primer error. Una vez notificado el primer error, no se notifican más errores hasta que la causa del primer error se haya eliminado en el módulo. Tras haber eliminado el primer error, se notifica el segundo error si la condición de este aún existe.

LED de estado de una Signal Board (SB)

Toda SB analógica incorpora un LED I/O Channel para cada una de las entradas y salidas analógicas.

Tabla 15-3 LED de estado de una Signal Board (SB)

Descripción	I/O Channel (Rojo/verde)
No se ha configurado o se está actualizando	Apagado
Tarjeta configurado sin errores	Encendido (verde)
Error de E/S (con diagnóstico habilitado)	Rojo intermitente
Error de E/S (con diagnóstico inhibido)	Encendido (verde)

15.2 Establecer una conexión online con una CPU

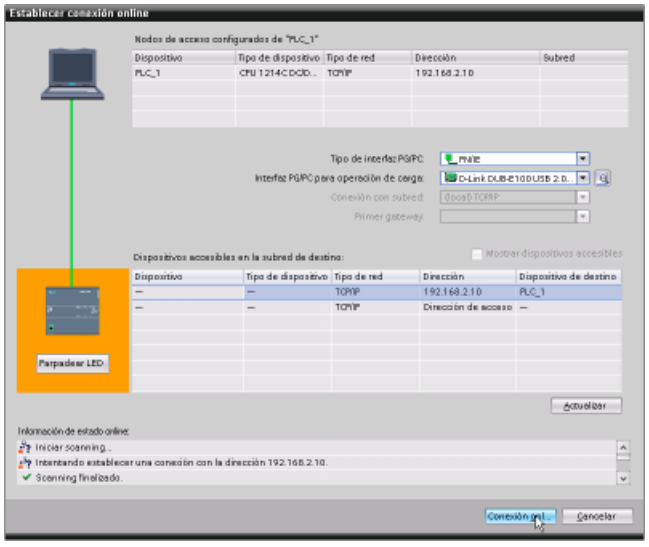
Debe establecerse una conexión online entre la programadora y la CPU para cargar programas y datos de ingeniería del proyecto, así como para las actividades siguientes:

- Comprobar programas de usuario
- Visualizar y cambiar el modo de operación de la CPU (Página 1210)
- Visualizar y ajustar la fecha y hora de la CPU (Página 1204)
- Visualizar la información del módulo
- Comparar y sincronizar (Página 1213) bloques de programa de offline a online
- Cargar y descargar bloques de programa
- Mostrar diagnóstico y el búfer de diagnóstico (Página 1211)
- Usar una tabla de observación (Página 1218) para probar el programa de usuario vigilando y modificando valores

- Utilizar una tabla de forzado permanente para forzar valores en la CPU (Página 1221)
- Restablecer la configuración de fábrica de la CPU (Página 1206)

Para establecer una conexión online con una CPU configurada, haga clic en la CPU en el árbol del proyecto y clic en el botón "Establecer conexión online" de la vista de proyectos:
Establecer conexión online

Si establece por primera vez una conexión online con esa CPU, debe seleccionar el tipo de interfaz PG/PC y la interfaz PG/PC específica en el cuadro de diálogo de conexión online antes de establecer una conexión online con una CPU encontrada en dicha interfaz.



Si la CPU tiene protección de datos de configuración confidenciales del PLC, puede que se le solicite confiar en la CPU. Se puede visualizar y verificar el certificado de la CPU y decidir si confiar en la conexión online de la CPU o interrumpirla.

Tras la conexión, los cuadros de color naranja indican una conexión online. Ahora, se pueden usar las herramientas online y de diagnóstico del árbol de proyectos, así como la Task Card de las herramientas online.

Consulte también

Protección de datos de configuración confidenciales del PLC (Página 161)

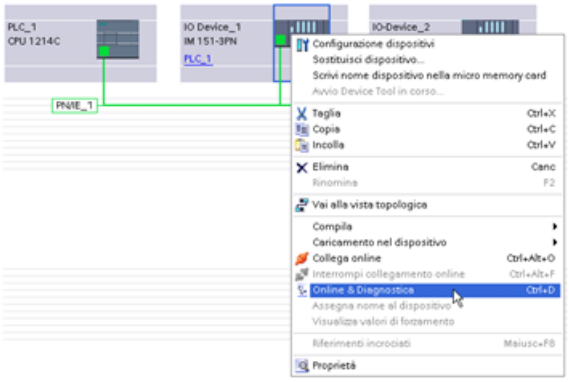
15.3 Asignar un nombre a un dispositivo PROFINET IO online

Los dispositivos de la red PROFINET deben tener asignados nombres antes de poder ser conectados a la CPU. Utilice el editor "Dispositivos y redes" para asignar nombres a los dispositivos PROFINET, en caso de que estos no hayan sido asignados o bien si el nombre del dispositivo debe ser modificado.

A cada dispositivo PROFINET IO se le debe asignar el mismo nombre en el proyecto de STEP 7 y, utilizando la herramienta "Online y diagnóstico", en la memoria de configuración del dispositivo PROFINET IO (p. ej. en la memoria de configuración de un módulo de interfaz ET200 S). Si hace

falta un nombre o este no coincide en ninguna ubicación, el modo de intercambio de datos PROFINET IO no se ejecutará.

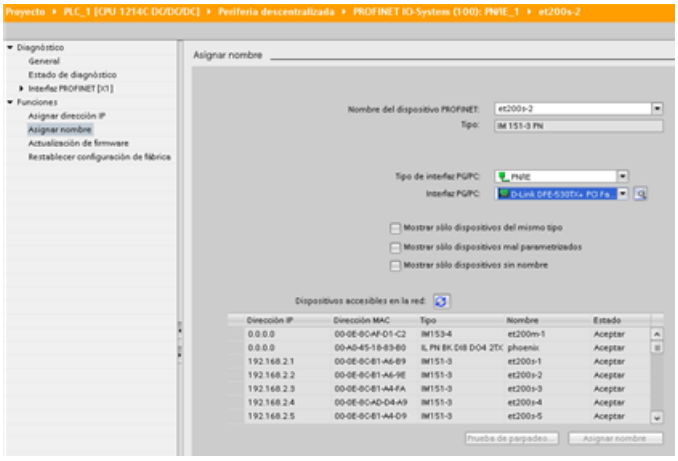
1. En el editor "Dispositivos y redes", haga clic con la tecla derecha del ratón en el dispositivo PROFINET IO deseado y seleccione "Online y diagnóstico".



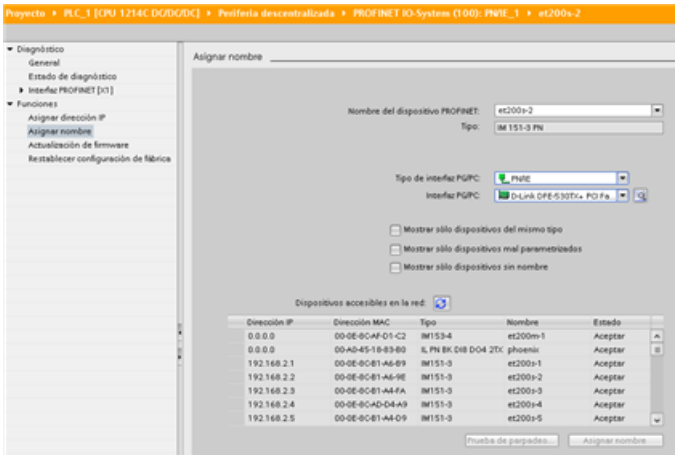
2. Seleccione los siguientes comandos de menú en el diálogo "Online y diagnóstico":

- "Funciones"
- "Asignar nombre de dispositivo PROFINET"

Haga clic en el botón "Actualizar lista" para visualizar todos los dispositivos PROFINET de la red.



3. En la lista visualizada, haga clic en el dispositivo PROFINET IO deseado y haga clic el botón "Asignar nombre" para escribir el nombre en la memoria de configuración del dispositivo PROFINET IO.



15.4 Ajustar la dirección IP y la hora

Es posible ajustar la dirección IP (Página 602) y la hora en la CPU online. Tras acceder a "Online y diagnóstico" desde el árbol de proyectos para una CPU en línea, es posible visualizar o cambiar la dirección IP. También es posible visualizar o ajustar los parámetros de fecha y hora en la CPU online.





Nota

Esta función sólo está disponible para una CPU que sólo tenga una dirección MAC (que aún no se le haya asignado una dirección IP) o en la que se hayan restablecido los valores de fábrica.

15.5 Actualización del firmware

Puede actualizar el firmware de una CPU conectada desde las herramientas online y de diagnóstico de STEP 7, siguiendo uno de los métodos siguientes:

- Actualización desde la CPU del proyecto
- Actualización desde los dispositivos accesibles en el árbol del proyecto

Actualización del firmware de una CPU del proyecto

Para realizar una actualización de firmware, proceda del siguiente modo:

1. Abra en el árbol del proyecto la CPU que corresponda a la CPU conectada.
2. Abra la vista Online y diagnóstico de la CPU conectada.
3. Seleccione "Actualización de firmware" en la carpeta "Funciones".
4. Desde la sección "Actualizador de firmware", haga clic en el botón Examinar y vaya hasta la ubicación que contiene el archivo de actualización del firmware. Podría ser una ubicación en el disco duro en la que se haya descargado un archivo de actualización de firmware para el S7-1200 (http://support.automation.siemens.com/WW/view/es/34612486/133100) desde la página web de Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/).

5. Seleccione un archivo que sea compatible con el módulo. En el caso de un archivo seleccionado, la tabla muestra los módulos compatibles.
6. Haga clic en el botón "Iniciar actualización". Siga los diálogos, si es necesario, para cambiar el estado operativo de la CPU.

STEP 7 muestra los diálogos de progreso mientras carga la actualización del firmware. Cuando haya finalizado, le indicará que arranque el módulo con el nuevo firmware.

Nota

Si no elige arrancar el módulo con el nuevo firmware, el firmware anterior permanecerá activo hasta que inicialice el módulo, por ejemplo desconectando y conectando la alimentación. El nuevo firmware solo pasará a estar activo después de inicializar el módulo.

Actualizar el firmware de los dispositivos accesibles

Para actualizar el firmware de uno o varios dispositivos accesibles, proceda del siguiente modo:

1. Abra el "Acceso online" en el árbol del proyecto.
2. Abra la interfaz de comunicación a la que está conectada la CPU.
3. Haga doble clic en "Actualizar dispositivos accesibles" y espere hasta que STEP 7 muestre los dispositivos online.
4. Expanda la CPU que desee actualizar y haga doble clic en "Online y diagnóstico".
5. Expanda "Actualización de firmware" en la carpeta "Funciones". Verá el PLC y sus módulos locales. Desde la selección de "PLC" o "Módulos locales" es posible iniciar la actualización del firmware partiendo de la sección "Actualizador de firmware", tal como se describe anteriormente.

También es posible actualizar el firmware utilizando uno de los métodos siguientes:

- Una SIMATIC Memory Card (Página 132)
- La página web estándar "Información del módulo" del servidor web (Página 867)
- La SIMATIC Automation Tool (https://support.industry.siemens.com/cs/ww/es/view/98161300)

15.6 Establecer o borrar la contraseña de protección de datos de configuración confidenciales del PLC

La función "Protección de datos de configuración confidenciales del PLC" permite proteger cada CPU de forma individual en el proyecto. Desde la configuración de dispositivos (Página 161) se puede habilitar esta protección y establecer una contraseña para la "protección de datos de configuración confidenciales del PLC". Los clientes como TIA Portal y SIMATIC Automation Tool solo pueden acceder a los datos confidenciales en el PLC mediante contraseña.

También es posible utilizar el asistente de seguridad (Página 160) para habilitar esta función y establecer la contraseña para la "protección de datos de configuración confidenciales del PLC".

Utilizar las herramientas online y de diagnóstico

Si la contraseña para la protección de los datos de configuración confidenciales del PLC no coincide con la del proyecto, la CPU no puede pasar al estado operativo RUN. Hay que establecer la contraseña correcta para la protección de los datos de configuración confidenciales del PLC o borrarla para que la CPU pase al estado operativo RUN.

Cuando la CPU está online (Página 1201), siga estos pasos para establecer o borrar la contraseña para la "protección de datos de configuración confidenciales del PLC" que se encuentran en la CPU online:

1. Ponga la CPU en estado operativo STOP.
2. Abra las herramientas Online y diagnóstico de la CPU.
3. Seleccione "Establecer contraseña de protección de datos de configuración del PLC" en el menú Funciones.
4. Haga clic en "Configuración" para establecer una contraseña o clic en "Borrar" para borrar la contraseña existente para la protección de datos de configuración confidenciales del PLC. Si "Borrar" no está disponible, la CPU online carece de contraseña para la protección de los datos de configuración confidenciales del PLC.

15.7 Restablecer la configuración de fábrica

Se puede restablecer un S7-1200 a sus ajustes originales de fábrica bajo las siguientes condiciones:

- La CPU tiene una conexión online.
- La CPU está en modo STOP.

Nota

Si la CPU está en modo RUN y se inicia la operación de restablecimiento, se puede poner en modo STOP después del acuse de recibo de una solicitud de confirmación.

Procedimiento

Para restablecer una CPU a sus ajustes de fábrica, proceda del siguiente modo:

1. Abra la vista Online y diagnóstico de la CPU.
2. Seleccione "Restablecer configuración de fábrica" en la carpeta "Funciones".
3. Seleccione la casilla de verificación "Borrar dirección IP" si desea eliminar la dirección IP.
4. Seleccione la casilla de verificación "Borrar contraseña para la protección de datos de configuración confidenciales del PLC" si desea eliminar esa contraseña. Puede que desee eliminar la contraseña, por ejemplo, si tiene previsto cargar un proyecto nuevo en la CPU o sustituir la CPU por un dispositivo nuevo (Página 1447).
5. Seleccione la casilla de verificación "Formatear Memory Card" si desea formatear la SIMATIC Memory Card insertada actualmente en la CPU online. Si está ejecutando su programa de CPU desde la SIMATIC Memory Card y desea formatear el programa, seleccione esta casilla de verificación.

6. Haga clic en el botón "Inicializar PLC".
7. En la consulta emergente haga clic en "Sí" para confirmar que desea inicializar el módulo con sus ajustes.

Resultado

El módulo cambia a modo STOP si es necesario y restablece los ajustes de fábrica. La CPU online realiza las siguientes acciones:

- Borra la memoria de trabajo y las áreas de datos remanentes
- Borra la memoria de carga si es una memoria de carga interna, borra la memoria de carga de la SIMATIC Memory Card SOLO si también se ha seleccionado "Formatear Memory Card"
- Ajusta todos los parámetros y áreas de operandos a sus valores configurados
- Borra el búfer de diagnóstico
- Reajusta la hora
- Borra los datos I&M (identificación y mantenimiento) excepto para I&M0
- Inicializa los contadores de horas de funcionamiento
- Conserva o borra la dirección IP en función de la selección que haya hecho. (La dirección MAC es fija y nunca se cambia.)
  Si no ha seleccionado "Borrar dirección IP", la CPU conserva la dirección IP, la máscara de subred y la dirección de router (si se utiliza) en los ajustes de la configuración hardware, a no ser que haya modificado esos valores en el programa de usuario o con otra herramienta, en cuyo caso la CPU restaura los valores modificados
- Borra el registro de control (Página 143), si lo hay
- Borra o conserva la contraseña de protección de datos de configuración confidenciales del PLC en función de su ajuste
- Formatea la SIMATIC Memory Card si hay una insertada en la CPU online y ha seleccionado la opción para formatear la tarjeta

15.8 Comprobar si un módulo tiene algún defecto (guardar datos de servicio)

Guardar datos de servicio

Las CPU S7-1200 V4.5 permiten guardar los datos de servicio de un módulo.

Cuando hay que realizar tareas de servicio, SIEMENS Customer Support solicita información sobre el estado de un módulo del sistema con fines de diagnóstico. Si eso ocurre en su sistema, el Customer Support puede pedirle que guarde los datos de servicio del módulo y que envíe el archivo generado.

Cómo se guardan los datos de servicio de un módulo

Se pueden guardar datos de servicio de un módulo en la vista online y de diagnóstico en las siguientes ubicaciones:

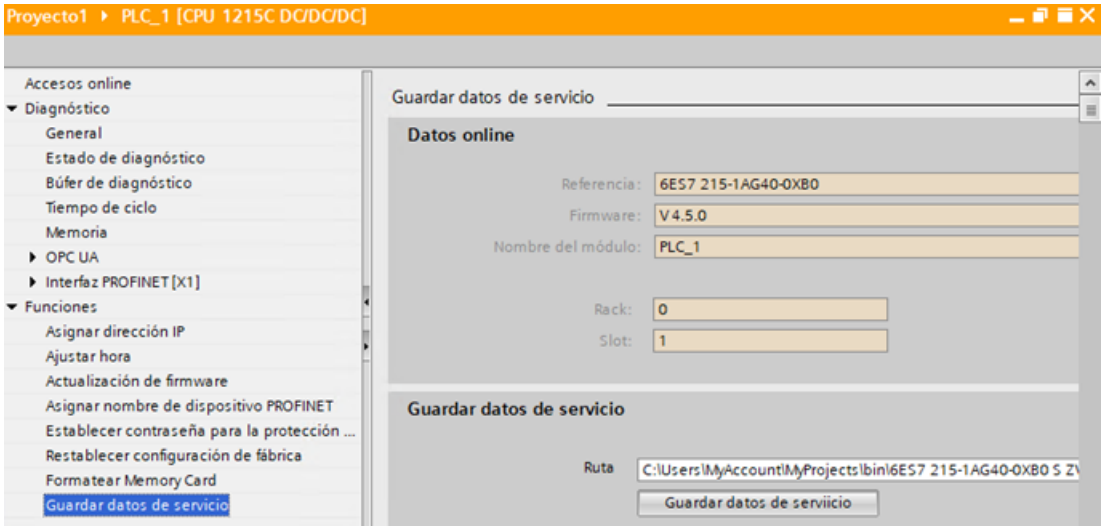
En la carpeta "Funciones" del grupo "Guardar datos de servicio", el grupo "Guardar datos de servicio" se compone de las siguientes áreas:

- Datos online
- Guardar datos de servicio

"Datos online"

El área "Datos online" muestra los siguientes datos del módulo:

- Referencia
- Versión de firmware
- Nombre del módulo (se ha definido al configurar el hardware)
- Rack
- Slot



"Guardar datos de servicio"

Proceda del siguiente modo para crear y guardar un archivo con datos de servicio especiales:

1. Seleccione el lugar del sistema de archivos en el que desea guardar el archivo:
   - Utilice la ruta predeterminada en el campo "Ruta".
   - Haga clic en el botón de tres puntos (examinar). En el cuadro de diálogo que se abre, especifique la ruta que desee e introduzca el nombre del archivo.
2. Haga clic en el botón "Guardar datos".
3. Se muestra una indicación de estado de la lectura de los datos de servicio. Una vez que la extracción de los datos de servicio ha finalizado correctamente, se muestra la pantalla que indica que el almacenamiento de los datos de servicio ha concluido correctamente.

Aspectos importantes:

- Los datos de servicio pueden extraerse de S7-1200 tanto en estado RUN como en estado STOP. No es posible leer en estado Fallo/Fatal.
- Si se ha programado un nivel de protección por contraseña en la CPU, hay que autenticar la contraseña antes de extraer los datos de servicio. La autenticación por contraseña es necesaria para todos los niveles de contraseña, dado que el proceso de extracción de los datos de servicio incluye la escritura de un registro de datos.
- TIA Portal permite guardar un archivo de datos de servicio.
- Los elementos de datos de servicio S7-1200 contenidos en el archivo de datos de servicio están cifrados.

15.9 Formatear una SIMATIC Memory Card desde STEP 7

Puede formatear la SIMATIC Memory Card de la CPU conectada desde las herramientas online y de diagnóstico de STEP 7. Para ello, proceda del siguiente modo:

1. Asegúrese de que la CPU está en estado STOP. Recuerde que si la CPU está en estado operativo RUN y se inicia una operación de formateo, STEP 7 pide permiso para poner la CPU en STOP.
2. Inserte una SIMATIC Memory Card en la CPU conectada.
3. Abra Online y diagnóstico para la CPU conectada desde la CPU del proyecto o desde los dispositivos accesibles en el árbol del proyecto del Acceso online.
4. Si la CPU no está online, seleccione "Pasar a online" para la CPU conectada.
5. Seleccione "Formatear Memory Card" en el menú "Funciones".
6. Haga clic en "Formatear".
7. Confirme la pregunta de seguridad con "Sí".

STEP 7 formatea la SIMATIC Memory Card y muestra un mensaje en la ventana de información una vez ha terminado. La CPU está en STOP cuando finaliza la operación de formateo y los LED STOP y MAINT parpadean. No es posible pasar al estado operativo RUN en este momento; hay que llevar a cabo una de las acciones siguientes:

- Retirar la Memory Card y rearrancar la CPU: Si la memoria de carga interna de la CPU contiene un programa, la CPU arranca con dicho programa.
- Rearrancar la CPU sin retirar la Memory Card: Si la memoria de carga interna de la CPU contiene un programa, la CPU lo copia en la SIMATIC Memory Card y arranca con dicho programa. Si la memoria de carga interna de la CPU no contiene ningún programa, la CPU convierte la SIMATIC Memory Card en una tarjeta de programa (Página 125) y espera una descarga.

Riesgos asociados al proceso de puesta fuera de servicio

Las CPU S7-1200 no soportan el borrado seguro de la SIMATIC Memory Card y de la flash interna. Por tanto, para evitar la pérdida de información confidencial o propietaria deberá eliminar la CPU y la SIMATIC Memory Card de forma segura durante el proceso de puesta fuera de servicio.

Nota

El formateo de una SIMATIC Memory Card no afecta al contenido de la memoria de carga interna.

Si la CPU utilizaba la memoria de carga interna cuando se insertó la SIMATIC Memory Card y no se ha rearrancado la CPU entre la inserción de la tarjeta y la ejecución de la operación de formateo, la CPU conserva el contenido de la memoria de carga interna.

15.10 Panel de control de la CPU online



El "Panel de control de la CPU" muestra el estado operativo (STOP o RUN) de la CPU online. También indica si la CPU tiene un error o si se están forzando valores.

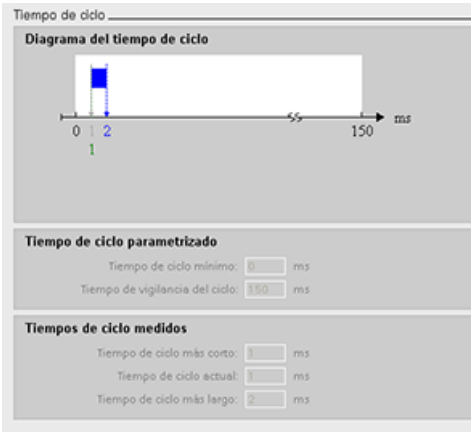
Utilice el panel de control de la CPU en la Task Card de las herramientas online para cambiar el estado operativo de una CPU online. La Task Card de las herramientas online es accesible siempre que la CPU esté online.

15.11 Vigilar el tiempo de ciclo y la carga de la memoria

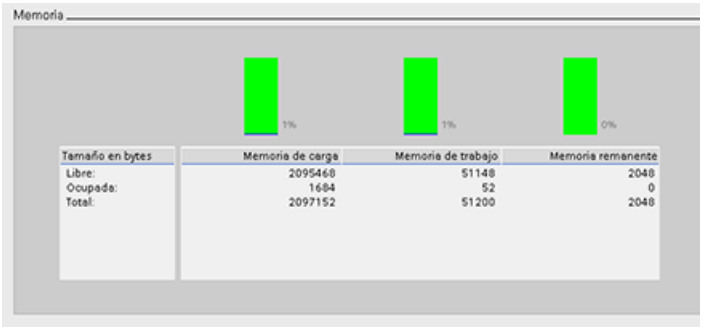
Es posible vigilar el tiempo de ciclo y la carga de la memoria de una CPU online.

Tras establecer una conexión con la CPU online, abra la Task Card de las herramientas online para visualizar los siguientes valores medidos:

- Tiempo de ciclo
- Carga de la memoria





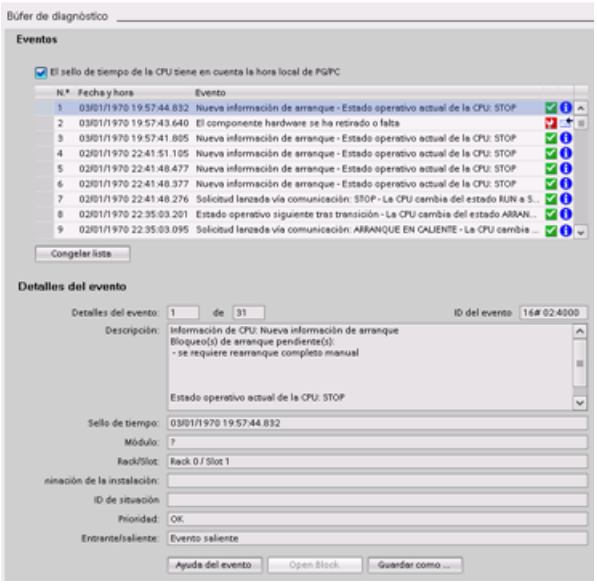




15.12 Visualizar los eventos de diagnóstico de la CPU

El búfer de diagnóstico permite consultar las actividades recientes de la CPU. El búfer de diagnóstico es accesible desde "Online y diagnóstico" para una CPU online en el árbol del proyecto. Contiene las entradas siguientes:

- Eventos de diagnóstico
- Cambios del estado operativo de la CPU (transiciones a STOP o RUN)



La primera entrada contiene el evento más reciente. Toda entrada del búfer de diagnóstico incluye la fecha y hora de registro del evento, así como una descripción.

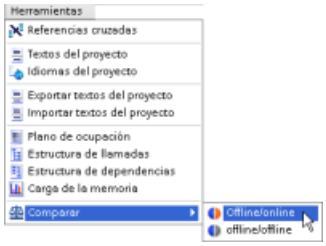
El número máximo de entradas depende de la CPU. Se soportan 50 entradas como máximo.

Solo los 10 eventos más recientes del búfer de diagnóstico se almacenan de forma permanente. Si se restablece la configuración de fábrica de la CPU, se reinicializa el búfer de diagnóstico y se borran sus entradas.

También se puede utilizar la instrucción GET_DIAG (Página 455) para recopilar la información de diagnóstico.

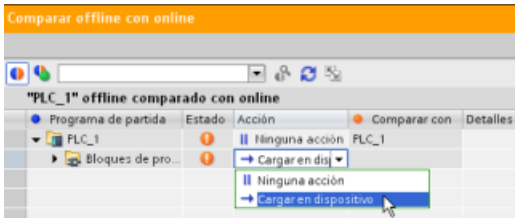
15.13 Comparar CPUs online y offline

Los bloques lógicos de una CPU online y los del proyecto se pueden comparar. Si los bloques lógicos del proyecto no coinciden con los de la CPU online, el editor de comparación permite sincronizar el proyecto con la CPU online descargando los bloques lógicos del proyecto en la CPU, o eliminando del proyecto los que no existan en la CPU online.



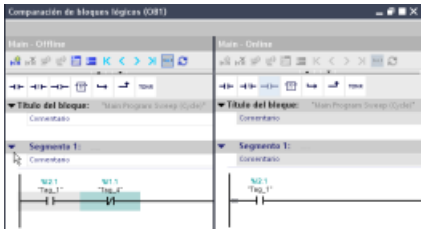
Seleccione la CPU del proyecto.

Use el comando "Comparar offline/online" para abrir el editor de comparación. (Acceda al comando desde el menú "Herramientas" o haciendo clic con el botón derecho del ratón en la CPU del proyecto.)



Haga clic en la columna "Acción" de un objeto para seleccionar si elimina el objeto, no realiza ninguna acción, o carga el objeto en el dispositivo.

Haga clic en el botón "Sincronizar" para cargar los bloques lógicos.



Haga clic con el botón derecho del ratón en un objeto en la columna "Comparar con" y seleccione el botón "Iniciar comparación detallada" para mostrar los bloques lógicos uno junto a otro.

La comparación detallada destaca las diferencias entre los bloques lógicos de la CPU online y los de la CPU del proyecto.

Nota

Acceso de lectura necesario en la CPU protegida para las operaciones de comparación offline/online

Para STEP 7 V14 o posterior, el nivel de seguridad "Acceso HMI" no es suficiente para llevar a cabo las operaciones de comparación offline/online. Hay que tener "Acceso de lectura" o "Acceso completo" para realizar las operaciones de comparación offline/online.

Consulte también Protección de acceso para la CPU (Página 163)

15.14 Realizar una comparación online/offline de la topología

Desde el sinóptico de topología de STEP 7 es posible comparar la topología offline configurada con la topología online real.

Procedimiento

Para encontrar las diferencias entre la topología configurada y la real, proceda del siguiente modo:

1. Abra la tabla topológica general de la vista topológica.
2. Haga clic en el botón "Comparación offline/online" de la barra de herramientas de la vista topológica general: 

Resultado

STEP 7 elimina las columnas "Estación interlocutora", "Interfaz del interlocutor" y "Datos del cable" de la tabla topológica general e inserta columnas de comparación para "Estado" y "Acción". Para cada dispositivo o puerto de la vista topológica general, la columna de estado muestra el estado de comparación del siguiente modo:

Icono	Significado
	Topología divergente en al menos un componente de nivel inferior
	Topología idéntica
	La información topológica solo está disponible offline o el dispositivo está deshabilitado
	La información topológica solo está disponible online
	Topología divergente
	El dispositivo no soporta funciones topológicas

Para cada puerto o dispositivo comparado, la columna de acción ofrece las siguientes posibilidades:

Icono	Significado
	No hay acciones posibles
	Adoptar la interconexión online

Para repetir la comparación, haga clic en el botón  de la barra de herramientas dentro de la vista general de topología.

Encontrará más información sobre la vista topológica, el sinóptico de topología y las comparaciones online/offline de topología en el Sistema de información de STEP 7. También encontrará información adicional en el manual PROFINET con STEP 7 V13 (https://support.industry.siemens.com/cs/ww/es/view/49948856).

15.15 Vigilar y modificar valores en la CPU

STEP 7 ofrece herramientas online para vigilar la CPU:

- Es posible visualizar o vigilar los valores actuales de las variables. La función de vigilancia no modifica la secuencia del programa. Facilita información sobre la secuencia y los datos del programa en la CPU.
- También se pueden utilizar otras funciones para controlar la secuencia y los datos del programa de usuario:
  - Es posible modificar el valor de las variables en la CPU online para observar cómo reacciona el programa de usuario.
  - Se puede forzar de forma permanente una salida periférica (como Q0.1:P o "Start":P) a un valor específico.
  - Es posible desbloquear las salidas en modo STOP.

  Nota

  Hay que prestar siempre atención al utilizar las funciones de forzado. Estas funciones pueden influir seriamente en la ejecución del programa de usuario/de sistema.

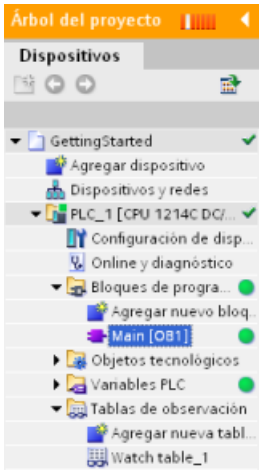
Tabla 15-4 Capacidades online de los editores de STEP 7

Editor	Vigilar	Modificar	Forzado permanente
Tabla de observación	Sí	Sí	No
Tabla de forzado permanente	Sí	No	Sí
Editor de programas	Sí	Sí	No
Tabla de variables	Sí	No	No
Editor de DB	Sí	No	No

15.15.1 Conexión online para observar los valores en la CPU

Para observar las variables debe existir una conexión online con la CPU. Haga clic en el botón "Establecer conexión online" de la barra de herramientas.





Una vez establecida la conexión con la CPU, los encabezados de las áreas de trabajo de STEP 7 aparecen en color naranja.

El árbol de proyectos muestra una comparación entre el proyecto offline y la CPU online. Un círculo verde significa que la CPU y el proyecto están sincronizados, es decir, que ambos tienen la misma configuración y programa de usuario.

Las tablas de variables muestran las variables. Las tablas de observación también pueden mostrar las variables, así como direcciones directas.





Para observar la ejecución del programa de usuario y visualizar los valores de las variables, haga clic en el botón "Observar todos" de la barra de herramientas.



El campo "Valor de observación" muestra el valor de cada variable.

15.15.2 Visualización del estado en el editor de programas

También es posible observar el estado de un máximo de 50 variables en los editores de programas KOP y FUP. Utilice la barra de editores para visualizar el editor KOP. La barra de editores permite conmutar la vista entre los editores abiertos sin tener que abrirlos o cerrarlos.

En la barra de herramientas del editor de programas, haga clic en el botón "Activar/desactivar observación" para ver el estado del programa de usuario.





La red del editor de programas indica el flujo de corriente en color verde.

También se puede hacer clic con el botón derecho del ratón en la instrucción o parámetro para modificar el valor de la instrucción.

15.15.3 Obtener una instantánea de los valores online de un DB para restaurar valores

Existe la posibilidad de obtener una instantánea de los valores reales de variables de bloques de datos desde una CPU online para utilizarlos posteriormente.

Tenga en cuenta los siguientes requisitos:

- Es necesario disponer de una conexión online con la CPU.
- El DB debe estar abierto en STEP 7.

Obtener una instantánea

Para obtener una instantánea, proceda del siguiente modo:

1. En el editor de DB, haga clic en el botón "Observar todos": La columna "Valor de observación" muestra los valores de datos reales.
2. Haga clic en el botón para obtener una instantánea de los valores reales y visualizarlos en la columna "Instantánea".

Esta instantánea puede utilizarse más adelante para actualizar los valores reales de la CPU o sustituir los valores de arranque.

Copiar los valores de la instantánea en la CPU

Para copiar los valores de una instantánea en los valores reales de las variables de bloques de datos de la CPU, haga clic en el botón siguiente:

La CPU online carga los valores de la instantánea en los valores reales. La columna "Valor de observación" muestra los valores reales de la CPU. Es posible que el ciclo cambie posteriormente los valores de la CPU respecto de los valores de la instantánea, pero en el momento de realizar la copia la CPU carga los valores de la instantánea en una descarga coherente.

Nota

Tenga en cuenta que, si la instantánea contiene información de estado, valores del temporizador o información calculada, la CPU restablece estos valores a partir del momento en que se realizó la instantánea.

Copiar los valores de la instantánea en los valores de arranque

Para copiar los valores de una instantánea en los valores de arranque de las variables de bloques de datos, haga clic en el botón siguiente: 

Después de compilar el DB y descargarlo en la CPU, el DB utiliza los nuevos valores de arranque cuando la CPU pasa a RUN.

Copiar valores individuales de instantánea u observación en los valores de arranque

El editor de bloques de datos también permite copiar valores individuales y pegarlos en los valores de arranque. Solo hay que hacer clic en un valor dentro de cualquier columna de valores y seleccionar Copiar para incorporarlo al portapapeles de Windows. Seguidamente, haga clic con el botón derecho del ratón en cualquier valor de arranque y seleccione Pegar para reemplazar dicho valor con el valor del portapapeles.

Después de compilar el DB y descargarlo en la CPU, el DB utiliza los nuevos valores de arranque cuando la CPU pasa a RUN.

15.15.4 Utilizar una tabla de observación para observar y modificar valores en la CPU

Una tabla de observación permite observar y forzar datos a medida que la CPU ejecuta el programa. Estos datos pueden ser de la memoria imagen de proceso (I o Q), M, DB o entradas físicas (I_:P), dependiendo de la función de observación o forzado. No es posible observar con precisión las salidas físicas (Q_:P) debido a que la función de vigilancia sólo puede visualizar el último valor escrito desde la memoria Q y no lee los valores reales de las salidas físicas.

La función de vigilancia no modifica la secuencia del programa. Facilita información sobre la secuencia y los datos del programa en la CPU.

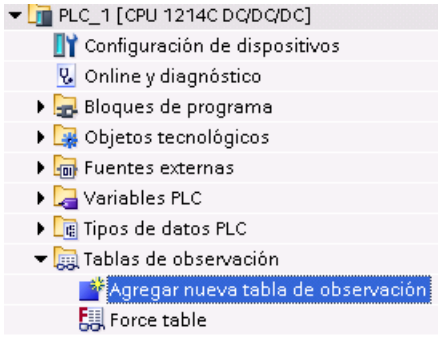
Las funciones de control permiten al usuario controlar la secuencia y los datos del programa. Debe prestar atención al utilizar las funciones de forzado. Estas funciones pueden influir seriamente en la ejecución del programa de usuario/de sistema. Las tres funciones de control son: forzar, forzar permanentemente y desbloquear salidas en STOP.

La tabla de observación permite realizar las siguientes funciones online:

- Observar el estado de las variables
- Forzar los valores de las distintas variables

Es posible seleccionar cuándo se debe observar o forzar la variable:

- Inicio del ciclo: Lee o escribe el valor al inicio del ciclo
- Fin del ciclo: Lee o escribe el valor al final del ciclo
- Cambiar a STOP



Para crear una tabla de observación, proceda del siguiente modo:

1. Haga doble clic en "Agregar nueva tabla de observación" para abrir una tabla de observación nueva.
2. Introduzca el nombre de la variable o agregue una variable a la tabla de observación.

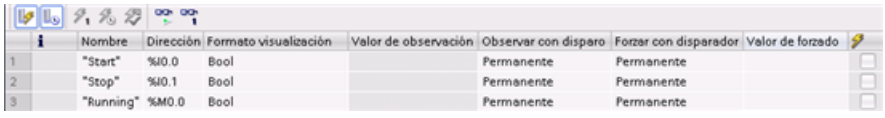
Las siguientes opciones están disponibles para observar variables:

- Observar todos: Este comando inicia la observación de las variables visibles en la tabla de observación activa.
- Observar inmediatamente: Este comando inicia la observación de las variables visibles en la tabla de observación activa. La tabla de observación observa las variables inmediatamente y una sola vez.

Las siguientes opciones están disponibles para forzar variables:

- "Forzar a 0" pone a "0" el valor de una dirección seleccionada.
- "Forzar a 1" pone a "1" el valor de una dirección seleccionada.
- "Forzar inmediatamente" cambia inmediatamente los valores de las direcciones seleccionadas durante un ciclo.
- "Forzar con disparador" cambia los valores de las direcciones seleccionadas.
  Esta función no confirma si las direcciones seleccionadas se han forzado realmente. Si se requiere una confirmación del cambio, utilice la función "Forzar inmediatamente".
- "Habilitar salidas de periferia" desactiva el comando para inhibir las salidas y está disponible sólo cuando la CPU se encuentra en estado operativo STOP.

Para observar las variables debe existir una conexión online con la CPU.



	Nombre	Dirección	Formato visualización	Valor de observación	Observar con disparo	Forzar con disparador	Valor de forzado
1	"Start"	%I0.0	Bool		Permanente	Permanente	
2	"Stop"	%I0.1	Bool		Permanente	Permanente	
3	"Running"	%M0.0	Bool		Permanente	Permanente	

Utilice los botones de la parte superior de la tabla de observación para seleccionar las diversas funciones.

Introduzca el nombre de la variable que desea observar y seleccione un formato de visualización en la lista desplegable. Si existe una conexión online con la CPU, haga clic en el botón "Observar" para visualizar el valor real del punto de datos en el campo "Valor de observación".

15.15.4.1 Utilizar un disparador para observar o forzar variables PLC

Si se utiliza un disparador es posible determinar en qué punto del ciclo debe observarse o forzarse la dirección seleccionada.

Tabla 15-5 Tipos de disparos

Disparo	Descripción
Permanente	Recoge los datos continuamente
Al inicio del ciclo	Permanente: Recoge los datos continuamente al inicio del ciclo, después de que la CPU lee las entradas
	Único: Recoge los datos al inicio del ciclo, después de que la CPU lee las entradas
Al final del ciclo	Permanente: Recoge los datos continuamente al final del ciclo, antes de que la CPU escribe en las salidas
	Único: Recoge los datos una vez al final del ciclo, antes de que la CPU escribe en las salidas
Transición a STOP	Permanente: Recoge los datos continuamente cuando la CPU pasa a STOP
	Único: Recoge los datos una vez cuando la CPU pasa a STOP

Para modificar una variable PLC en un disparador determinado, seleccione el inicio o final del ciclo.

- Forzar una salida: El mejor evento de disparo para forzar una salida es al final del ciclo, inmediatamente antes de que la CPU escribe en las salidas.
  Observe el valor de las salidas al inicio del ciclo para determinar qué valor se escribe en las salidas físicas. Asimismo, observe las salidas antes de que la CPU escriba los valores en las salidas físicas para comprobar la lógica del programa y compararla con la reacción real de las E/S.
- Forzar una entrada: El mejor evento de disparo para forzar una entrada es al inicio del ciclo, inmediatamente después de que la CPU lee las entradas y antes de que el programa utiliza los valores de entrada.
  Si sospecha que los valores cambian durante el ciclo, puede que desee observar el valor de las entradas al final del ciclo para garantizar que el valor de la entrada al final del ciclo no haya cambiado desde inicio del ciclo. Si los valores son diferentes, puede que el programa de usuario esté escribiendo en las entradas por error.

Para diagnosticar la causa de una transición a STOP de la CPU, utilice el disparador "Transición a STOP" para capturar los últimos valores de proceso.

15.15.4.2 Desbloquear las salidas en modo STOP

La tabla de observación permite escribir en las salidas cuando la CPU está en modo STOP. Esta función sirve para comprobar el cableado de las salidas y verificar que el cable conectado a un borne de salida lanza una señal "high" o "low" al terminal del dispositivo de proceso al que está conectado.

⚠ ADVERTENCIA

Riesgos asociados con la escritura de salidas físicas en estado operativo STOP

Aunque la CPU está en estado operativo STOP, la habilitación de una salida física puede activar el punto del proceso a la que está conectada, por lo que es posible que provoque un funcionamiento inesperado del equipo. Un funcionamiento inesperado del equipo puede provocar la muerte o lesiones corporales graves.

Antes de escribir en una salida de la tabla de observación, asegúrese de que al cambiar la salida física no provoque un funcionamiento inesperado del equipo. Siempre respete las precauciones de seguridad para su equipo de proceso.

El estado de las salidas se puede cambiar en estado operativo STOP si están habilitadas las salidas. Si las salidas están inhibidas, no es posible modificarlas en estado operativo STOP. Para habilitar la modificación en estado operativo STOP de las salidas de la tabla de observación, proceda del siguiente modo:

1. Seleccione el comando del menú "Modo avanzado" en el menú "Online".
2. Seleccione la opción "Desbloquear salidas de periferia" del comando "Modificar" del menú "Online" o desde el menú contextual después de hacer clic con el botón derecho del ratón en la fila de la tabla de observación.

No puede habilitar las salidas en el estado operativo STOP si ha configurado E/S descentralizadas. Si lo intenta, se devuelve un error.

Cuando la CPU pasa a estado operativo RUN se inhibe la opción "Habilitar salidas de periferia".

Si alguna entrada o salida se fuerza permanentemente, la CPU no podrá desbloquear las salidas en estado operativo STOP. Primero es preciso cancelar la función de forzado permanentemente.

15.15.5 Forzar valores permanentemente en la CPU

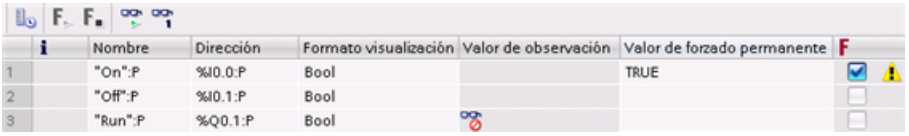
15.15.5.1 Utilizar la tabla de forzado permanente

Una tabla de forzado permanente incluye una función de "forzado permanente" que sobrescribe el valor de una entrada o salida con un valor específico para la dirección de entrada o salida de periferia. La CPU aplica el valor forzado permanentemente en la memoria imagen de proceso de las entradas antes de ejecutar el programa de usuario y en la memoria imagen de proceso de las salidas antes de escribir las salidas en los módulos.

Nota

Los valores de forzado permanente se guardan en la CPU y no en la tabla de forzado permanente.

No se puede forzar permanentemente una entrada (o dirección "I") ni una salida (o dirección "Q"). No obstante, sí que es posible forzar permanentemente una entrada o salida de periferia. La tabla de forzado permanente agrega automáticamente ":P" a la dirección (por ejemplo: "On":P o "Run":P).



	Nombre	Dirección	Formato visualización	Valor de observación	Valor de forzado permanente	F
1	"On":P	%I0.0:P	Bool		TRUE	
2	"Off":P	%I0.1:P	Bool			
3	"Run":P	%Q0.1:P	Bool			

En la celda "Valor de forzado permanente", introduzca el valor de la entrada o salida que debe forzarse. Puede emplearse la casilla de verificación de la columna "Forzado permanente" para permitir el forzado permanente de la entrada o salida.



Utilice el botón "Iniciar o reemplazar forzado permanente" para forzar permanentemente el valor de las variables en la tabla de forzado permanente. Haga clic en el botón "Finalizar forzado permanente" para inicializar el valor de las variables.

En la tabla de forzado permanente, es posible vigilar el estado del valor forzado para una entrada. Sin embargo, no es posible vigilar el valor forzado de una salida.

También es posible observar el estado del valor forzado permanentemente en el editor de programas.



Nota

Cuando una entrada o salida se fuerza permanentemente en la tabla de forzado permanente, las acciones de forzado permanente se convierten en parte de la configuración del proyecto. Si se cierra STEP 7, los elementos forzados permanentemente permanecen activados en el programa de la CPU hasta que se borran. Para borrar estos elementos forzados permanentemente es necesario utilizar STEP 7 para establecer una conexión con la CPU online y utilizar la tabla de forzado permanente para desactivar o detener la función de forzado permanente de estos elementos.

15.15.5.2 Funcionamiento de la función de forzado permanente

La CPU permite forzar permanentemente las entradas y salidas. Para ello, es preciso indicar la dirección de la entrada o salida física (I_:P o Q_:P) en la tabla de forzado permanente e iniciar la función de forzado permanente.

El valor forzado permanentemente sobrescribe las lecturas de las entradas físicas en el programa. El programa utiliza el valor forzado permanentemente para el procesamiento. Cuando el programa escribe en una salida física, el valor de forzado permanente sobrescribe el de la salida. El valor forzado permanentemente aparece en la salida física y es utilizado por el proceso.

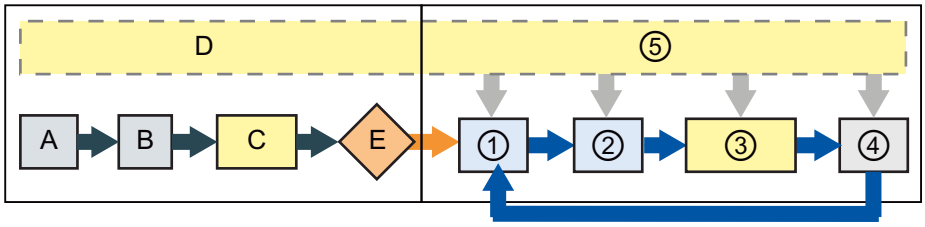
Cuando una entrada o salida se fuerza permanentemente en la tabla de forzado permanente, las acciones de forzado permanente se convierten en parte del programa de usuario. Aunque se haya cerrado el software de programación, las selecciones de forzado permanente permanecen activadas en el programa de la CPU hasta que son borradas al establecer una conexión online desde el software de programación y se para la función de forzado permanente. Los programas con entradas y/o salidas forzadas de forma permanente que se hayan cargado en una CPU diferente desde una Memory Card seguirán forzando permanentemente las E/S seleccionadas en el programa.

Si la CPU ejecuta el programa de usuario desde una Memory Card protegida contra escritura, el forzado permanente de una E/S no se puede iniciar ni cambiar desde una tabla de observación, ya que no es posible sobrescribir los valores en el programa de usuario protegido contra escritura. Todo intento de forzar permanentemente los valores protegidos contra escritura generará un error. Si se utiliza una Memory Card para transferir un programa de usuario, los elementos forzados permanentemente en esa Memory Card se transferirán a la CPU.

Nota

Las E/S digitales asignadas al HSC, PWM y PTO no se pueden forzar permanentemente

Las E/S digitales utilizadas por el contador rápido (HSC) y los dispositivos con modulación del ancho de pulso (PWM) y tren de impulsos (PTO) se asignan durante la configuración de dispositivos. Si se asignan direcciones de E/S digitales a dichos dispositivos, los valores de las direcciones de E/S asignadas no podrán modificarse utilizando la función de forzado permanente de la tabla de forzado.



Arranque

A La función de forzado permanente no afecta el borrado del área de memoria de las entradas (I).

B La función de forzado permanente no afecta la inicialización de los valores de salida.

C Durante la ejecución de los OBs de arranque, la CPU aplica el valor de forzado permanente cuando el programa de usuario accede a la entrada física.

D El almacenamiento de los eventos de alarma en la cola de espera no se ve afectado.

E La habilitación de escritura en las salidas no se ve afectada.

RUN

① Mientras escribe la memoria de las salidas (Q) en las salidas físicas, la CPU aplica el valor de forzado permanente a medida que se van actualizando las salidas.

② Al leer las entradas físicas, la CPU aplica los valores de forzado permanente inmediatamente antes de copiar las entradas en la memoria I.

③ Durante la ejecución del programa de usuario (OBs de ciclo), la CPU aplica el valor de forzado permanente cuando el programa de usuario accede a la entrada física o escribe en la salida física.

④ La función de forzado permanente no afecta el procesamiento de peticiones de comunicación ni los diagnósticos de autotest.

⑤ El procesamiento de las alarmas en cualquier parte del ciclo no se ve afectado.

15.16 Carga en estado operativo RUN

La CPU soporta la "carga en estado operativo RUN". Esta función permite efectuar cambios pequeños en el programa de usuario sin afectar demasiado a los equipos controlados. No obstante, también es posible realizar modificaciones considerables que podrían ser perjudiciales o incluso peligrosas.

⚠ ADVERTENCIA

Riesgos asociados con la carga en estado operativo RUN

Si los cambios se cargan en la CPU durante el estado operativo RUN, afectarán directamente al proceso. Si el programa se modifica estando la CPU en estado operativo RUN, es posible que se produzcan reacciones inesperadas en los equipos, lo que podría ocasionar la muerte o lesiones graves personales y/o daños materiales.

Solo el personal cualificado que tenga pleno conocimiento de los efectos que esta función puede tener en el sistema debería efectuar la carga en el estado operativo RUN.

La función "Carga en estado operativo RUN" permite realizar cambios en un programa y luego cargarlos en la CPU sin necesidad de pasar al estado operativo STOP:

- Es posible realizar cambios menores en el proceso en curso sin tener que desconectar el equipo (p. ej. modificar el valor de un parámetro).
- Esta función permite comprobar un programa más rápidamente (p. ej. invertir la lógica de un contacto normalmente abierto o cerrado).

Se pueden realizar los siguientes cambios en bloques de programa y variables y cargarlos en RUN:

- Crear, sobrescribir y eliminar funciones (FC), bloques de función (FB) y tablas de variables.
- Crear, borrar y sobrescribir bloques de datos (DB) y bloques de datos de instancia para bloques de función (FB). Puede añadirlos a las estructuras de los DB y cargarlos en estado operativo RUN. La CPU puede conservar los valores de las variables de bloques existentes e inicializar las nuevas variables de bloques de datos a sus valores iniciales o la CPU puede establecer las variables de bloques de datos en los valores iniciales en función de los ajustes de configuración (Página 1229). No es posible descargar un DB de servidor web (control o fragmento) en RUN.
- Sobrescribir bloques de organización (OB); sin embargo, no se pueden crear ni eliminar OB.

Puede cargar como máximo veinte bloques a la vez en estado operativo RUN. Si debe cargar más de veinte bloques, debe pasar la CPU al estado operativo STOP.

Si desea cargar cambios en un proceso real (a diferencia de un proceso simulado, lo cual podría ocurrir durante la comprobación de un programa), es de vital importancia analizar a fondo los posibles efectos en materia de seguridad para máquinas y operadores antes de realizar la carga.

Nota

Si la CPU está en estado operativo RUN y se realizan cambios en el programa, STEP 7 intentará siempre cargar primero en RUN. Si no desea que esto suceda, deberá poner la CPU en estado operativo STOP.

Si los cambios realizados no se soportan en "Cargar en RUN", STEP 7 pedirá al usuario que ponga la CPU a STOP.

15.16.1 Requisitos para la "Carga en estado operativo RUN"

Para poder cargar los cambios del programa en una CPU en estado operativo RUN, debe cumplir estos requisitos previos:

- Su versión de CPU es V3.0 o posterior.

  Nota

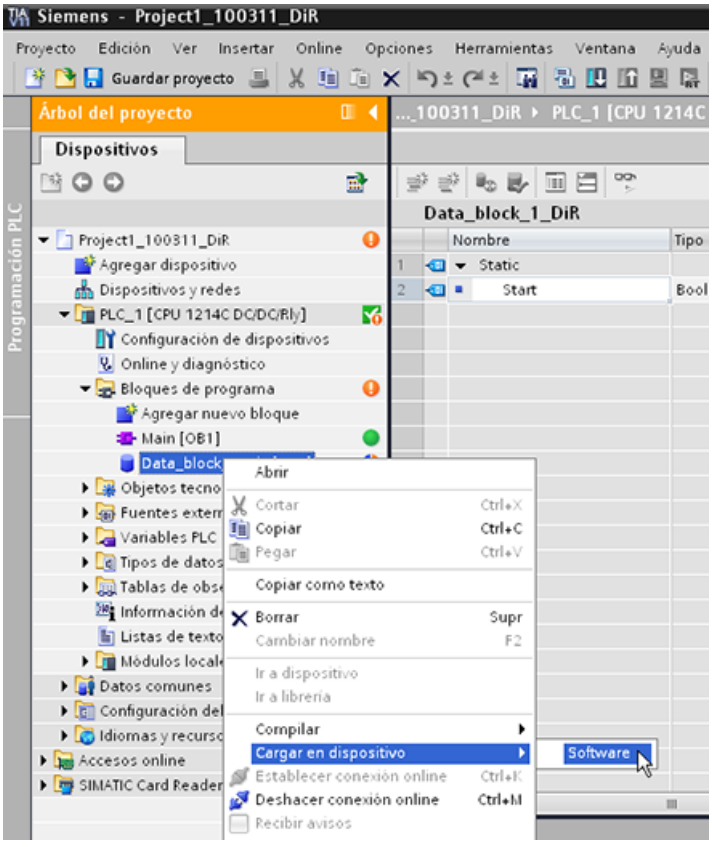
  Su versión de CPU debe ser V4.0 o superior para modificar la interfaz de bloque avanzada en estado RUN. (Página 1229)

- El programa debe compilarse correctamente.
- Se debe haber establecido correctamente la comunicación entre la programadora en la que se ejecuta STEP 7 y la CPU.

15.16.2 Modificar el programa en el estado operativo RUN

Para poner el programa en estado operativo RUN, primero hay que asegurarse de que tanto la CPU como el programa cumplen los requisitos (Página 1225) y, a continuación, llevar a cabo los pasos siguientes:

1. Para descargar el programa en RUN, seleccione uno de los métodos siguientes:
   - Seleccione el comando "Cargar en dispositivo" en el menú "Online".
   - Haga clic en el botón "Cargar en dispositivo" de la barra de herramientas.
   - En el "Árbol del proyecto" haga clic con el botón derecho del ratón en "Bloques de programa" y seleccione el comando "Cargar en dispositivo > Software".



Si el programa se compila correctamente, STEP 7 empieza a cargar el programa en la CPU.

2. Cuando STEP 7 le solicite que cargue el programa o cancele el funcionamiento, haga clic en "Cargar" para cargar el programa en la CPU.

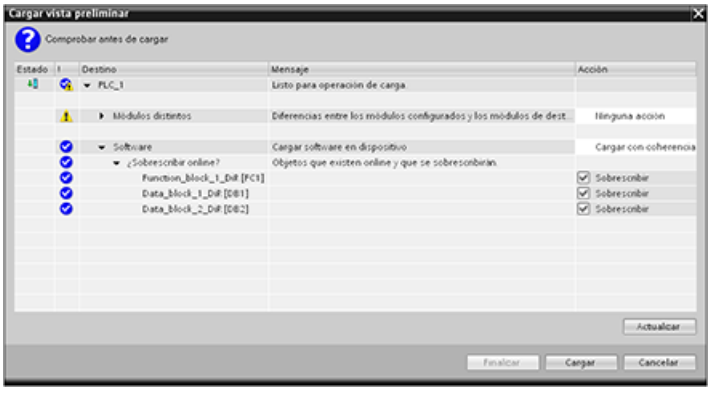
15.16.3 Descargar bloques seleccionados

Desde la carpeta Bloques de programa se puede seleccionar un único bloque o una selección de bloques para cargarlos.

Si selecciona un único bloque para cargarlo, entonces la única opción de la columna "Acción" es "Carga coherente". Puede ampliar la línea de categoría para asegurarse de que los bloques que deben cargarse. En este ejemplo se ha realizado una pequeña modificación del bloque offline y no es necesario cargar otros bloques.



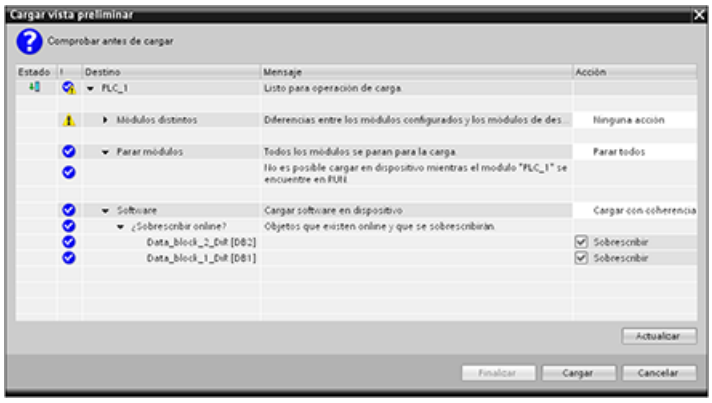
En este ejemplo se necesita más de un bloque para la descarga.



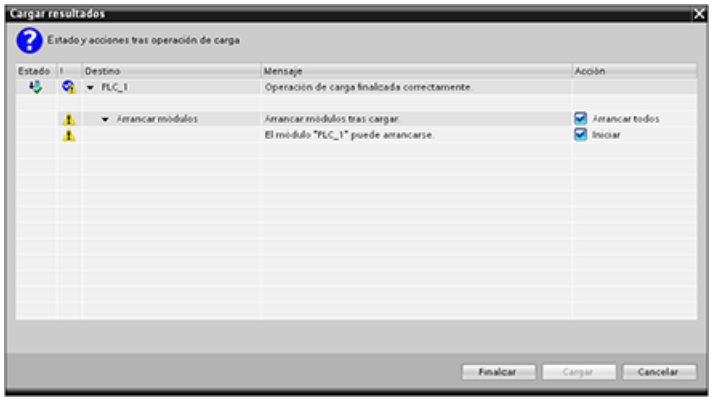
Nota

Puede cargar como máximo veinte bloques en el estado operativo RUN a la vez. Si debe cargar más de veinte bloques, debe poner la CPU en el estado operativo STOP.

Si intenta cargar en RUN, pero el sistema detecta que esto no es posible antes de la carga real, entonces en el cuadro de diálogo aparece la línea de categoría de módulos de parada.

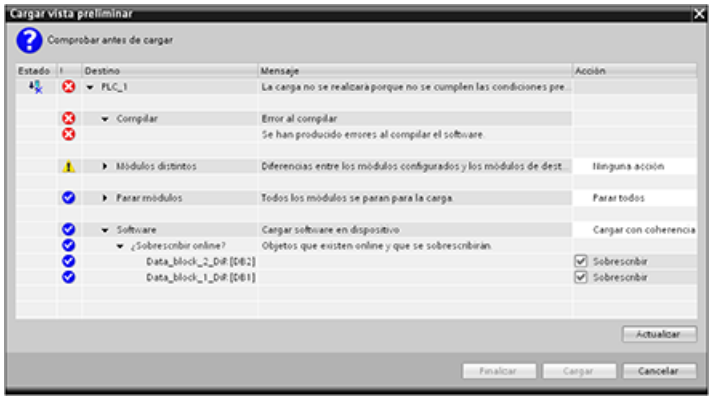


Haga clic en el botón "Cargar" y aparecerá el cuadro de diálogo "Cargar resultados". Haga clic en el botón "Finalizar" para finalizar la carga.

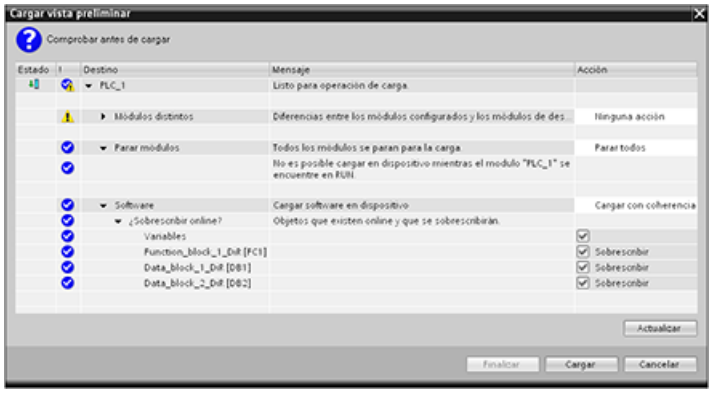


15.16.4 Descargar un único bloque seleccionado con un error de compilación en otro bloque

Si intenta realizar una carga coherente con un error de compilación en otro bloque, el cuadro de diálogo indicará un error y el botón de carga estará deshabilitado.



Debe corregir el error de compilación en el otro bloque. En ese caso, el botón "Cargar" pasa a estar activo.

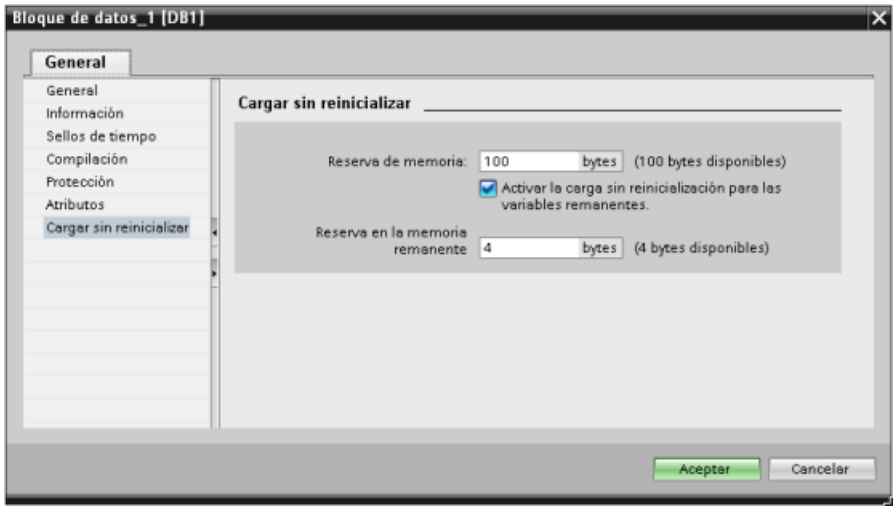


15.16.5 Modificación y carga de los bloques existentes en estado operativo RUN

La función "Carga en estado operativo RUN" le permite añadir y modificar variables en los bloques de datos y bloques de función y, a continuación, cargar el bloque cambiado en la CPU en estado operativo RUN.

Cargar sin reinicialización

Cada DB y FB tienen una cantidad de memoria reservada, que puede utilizar para añadir variables al bloque que posteriormente puede cargar en estado operativo RUN. De forma predeterminada, el tamaño inicial de la reserva de memoria es 100 bytes. Puede añadir variables adicionales a los datos hasta el tamaño de la reserva de memoria y cargar el bloque extendido en la CPU en estado operativo RUN. También puede aumentar la reserva de memoria si necesita más memoria para las variables adicionales en el bloque. Si la cantidad de variables es superior a la cantidad de memoria que ha asignado, no podrá cargar el bloque extendido en la CPU en estado operativo RUN.



La función "Carga sin reinicialización" permite extender un bloque de datos añadiendo más variables de bloques de datos y cargar el bloque de datos extendido en estado operativo RUN. De este modo, puede añadir variables al bloque de datos y cargarlo sin reinicializar el programa.

La CPU retiene los valores de las variables de bloques de datos existentes e inicializa las variables añadidas recientemente a sus valores de arranque.

Para habilitar esta función para un proyecto online con una CPU en estado operativo RUN, proceda del siguiente modo:

1. Desde la carpeta Bloques de programa del árbol del proyecto de STEP 7, abra el bloque.
2. Haga clic en el conmutador "Carga sin reinicialización" del editor de bloques para habilitar la función. (El icono tiene un cuadro todo alrededor cuando está activado:)
3. Haga clic en Aceptar cuando se le solicite que confirme su elección.
4. Añada variables a la interfaz del bloque y cargue el bloque en estado operativo RUN. Puede añadir y cargar tantas variables nuevas como lo permita su reserva de memoria.

Si ha añadido más bytes en el bloque de las que ha configurado para la reserva de memoria, STEP 7 mostrará un error cuando intente cargar el bloque en estado operativo RUN. Debe editar las propiedades del bloque para aumentar la cantidad. No puede borrar las entradas existentes ni modificar la "Reserva de memoria" del bloque mientras esté habilitada la función "Carga sin reinicialización". Para deshabilitar la función "Carga sin reinicialización", proceda del siguiente modo:

1. Haga clic en el conmutador "Carga sin reinicialización" del editor de bloques para deshabilitar la función. (El icono no tiene un cuadro todo alrededor cuando está desactivado:)
2. Haga clic en Aceptar cuando se le solicite que confirme su elección.
3. Cargue el bloque. En el cuadro de diálogo de carga, debe seleccionar "reinicializar" para cargar el bloque extendido.

A continuación, la carga reinicializa todas las variables de bloques nuevas y existentes a sus valores de arranque.

Carga de variables de bloques remanentes

Para cargar variables de bloques remanentes en estado operativo RUN, se necesita asignar una reserva de memoria remanente. Para configurar esta reserva de memoria, proceda del siguiente modo:

1. Desde la carpeta Bloques de programa del árbol del proyecto de STEP 7, haga clic con el botón derecho del ratón en el bloque y seleccione "Propiedades" en el menú contextual.
2. Seleccione la propiedad "Carga sin reinicialización".
3. Seleccione la casilla de verificación para "Activar la carga sin reinicialización para las variables remanentes".
4. Configure el número de bytes disponibles para la reserva de memoria remanente.
5. Haga clic en Aceptar para guardar los cambios.
6. Añada las etiquetas de bloque de datos remanentes al bloque de datos y cargue el bloque de datos en estado operativo RUN. Puede añadir y cargar tantas variables de bloque de datos remanentes nuevas como lo permita su reserva de memoria remanente.

Si ha añadido más bytes remanentes en el bloque de datos de las que ha configurado para la reserva de memoria remanente, STEP 7 mostrará un error cuando intente cargar el bloque en estado operativo RUN. Solo puede añadir etiquetas de bloque de datos remanentes hasta completar la reserva de memoria remanente para poder cargarlas en estado operativo RUN.

Al descargar las variables de bloque remanentes ampliadas, estas contienen sus valores actuales.

Configuración de cantidad de memoria reservada para bloques nuevos

El tamaño de reserva de memoria predeterminado para nuevos bloques de datos es de 100 bytes. Cuando cree un nuevo bloque, tendrá 100 bytes disponibles de reserva. Si desea que el tamaño de la reserva de memoria sea distinta para los bloques nuevos, puede cambiar el ajuste en la configuración de programación del PLC:

1. Desde STEP 7, seleccione el comando de menú **Opciones > Configuración**.
2. En el cuadro de diálogo Configuración, amplíe "Programación PLC" y seleccione "General".
3. En la sección "Carga sin reinicialización", introduzca el número de bytes para la reserva de memoria.

Cuando crea bloques nuevos, STEP 7 utiliza la configuración de reserva de memoria que ha introducido para los bloques nuevos.

Restricciones

Las restricciones siguientes se aplican a los bloques que están editando y cargando en estado operativo RUN:

- La ampliación de la interfaz de bloque mediante la adición de variables nuevas y la carga en estado operativo RUN solo está disponible para bloques optimizados (Página 182).
- No puede cambiar la estructura de un bloque y cargar el bloque cambiado en estado operativo RUN sin reinicializarlo. Al añadir nuevos miembros a la variable Struct (Página 115), para cambiar los nombres de variables, tamaños de matriz o estados remanentes es necesario reinicializar el bloque si lo carga en el estado operativo RUN. Las únicas modificaciones en las variables de bloque de datos existentes que puede realizar pudiendo todavía cargar el bloque en el estado operativo RUN sin efectuar la reinicialización, son los cambios en los valores de arranque (bloques de datos), valores predeterminados (bloques de función) o comentarios.
- No puede cargar más variables de bloque nuevas en estado operativo RUN de las que puede alojar la reserva de memoria.
- No puede cargar más variables de bloque remanentes nuevas en estado operativo RUN de las que puede alojar la reserva de memoria remanente.

15.16.6 Reacción del sistema si falla la descarga

Si durante el proceso inicial de carga en RUN falla una conexión de red, STEP 7 mostrará el cuadro de diálogo "Cargar vista preliminar" siguiente:



15.16.7 Factores que se deben tener en cuenta al cargar en estado operativo RUN

Antes de descargar el programa en modo RUN, considere los efectos que puede tener una modificación en modo RUN en el funcionamiento de la CPU en las situaciones siguientes:

- Si ha borrado la lógica de control de una salida, la CPU conservará el último estado de la salida hasta la próxima desconexión y conexión, o bien, hasta la próxima transición al estado operativo STOP.
- Si ha borrado un contador rápido o funciones de salida de impulsos que se estaban ejecutando, el contador o la salida de impulsos se seguirá ejecutando hasta la próxima desconexión y posterior conexión, o bien, hasta que se cambie a modo STOP.

- Cualquier lógica que dependa del estado del bit de primera consulta no se ejecutará hasta la próxima desconexión y posterior conexión, o bien hasta que se cambie de STOP a RUN. El bit de primera consulta solo se activa al cambiar a modo RUN y no se ve afectado por una descarga en modo RUN.
- Los valores actuales de los bloques de datos (DB) o variables se pueden sobrescribir.

Nota

Para poder descargar el programa en modo RUN, la CPU tiene que admitir la modificación en modo RUN, el programa debe compilarse correctamente y la comunicación entre STEP 7 y la CPU debe funcionar sin errores.

Se pueden realizar los cambios siguientes en variables y bloques de programa y descargarlos en modo RUN:

- Crear, sobrescribir y eliminar funciones (FC), bloques de función (FB) y tablas de variables.
- Crear y eliminar bloques de datos (DB). No obstante, los cambios de estructura de los DB no se pueden sobrescribir. Los valores iniciales de los DBs se pueden sobrescribir. No es posible descargar un DB de servidor web (control o fragmento) en RUN.
- Sobrescribir bloques de organización (OB); sin embargo, no se pueden crear ni eliminar OB.

Puede cargar como máximo veinte bloques en el estado operativo RUN a la vez. Si debe cargar más de veinte bloques, debe poner la CPU en el estado operativo STOP.

Una vez que inicie una carga, no puede realizar otras tareas en STEP 7 hasta que no se haya terminado la carga.

Instrucciones que pueden fallar debido a una "Carga en estado operativo RUN"

Las instrucciones siguientes pueden sufrir un error temporal cuando se activan en la CPU cambios en la carga en RUN. El error se produce al iniciar la instrucción mientras la CPU se está preparando para activar los cambios descargados. Durante este tiempo, la CPU suspende la iniciación del acceso del programa de usuario a la memoria de carga mientras completa el acceso del programa de usuario en curso a la memoria de carga. Esto sucede para que los cambios descargados puedan activarse de forma coherente.

Instrucción	Respuesta con la activación pendiente
DataLogCreate	STATUS = W#16#80C0, ERROR = TRUE
DataLogOpen	STATUS = W#16#80C0, ERROR = TRUE
DataLogWrite	STATUS = W#16#80C0, ERROR = TRUE
DataLogClose	STATUS = W#16#80C0, ERROR = TRUE
DataLogNewFile	STATUS = W#16#80C0, ERROR = TRUE
DataLogClear	STATUS = W#16#80C0, ERROR = TRUE
DataLogDelete	STATUS = W#16#80C0, ERROR = TRUE
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
Create_DB	RET_VAL = W#16#80C0
Delete_DB	RET_VAL = W#16#80C0
RTM	RET_VAL = 0x80C0

En todos los casos, la salida RLO de la instrucción será "false" cuando se produzca el error. El error es temporal. Si esto ocurre, la instrucción debería repetirse más adelante.

Nota

No debe repetirse la operación durante la ejecución actual del OB.

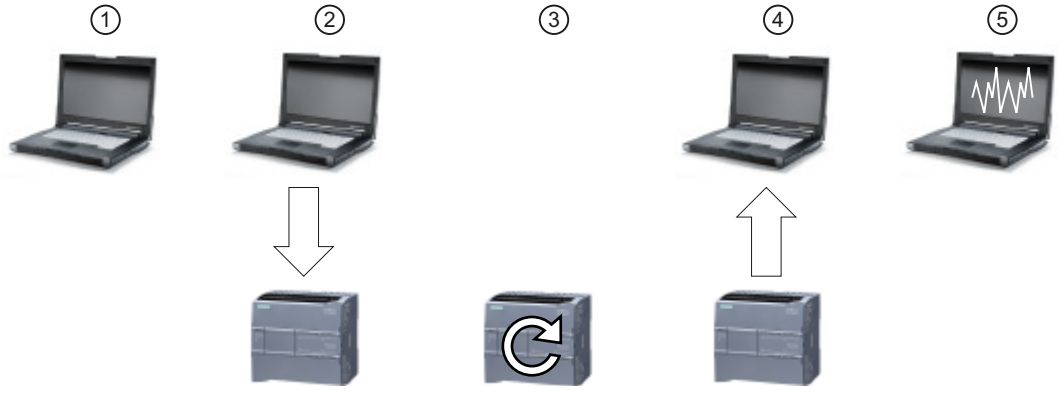
15.17 Trazas y registros de datos de la CPU en condiciones de disparo

STEP 7 proporciona funciones de analizador lógico y de trazas que le permitirán configurar variables para que el PLC pueda trazarlas o registrarlas. A continuación, puede cargar los datos de medición de seguimiento registrados en el dispositivo de programación y usar las herramientas de STEP 7 para analizar y administrar los datos y representarlos gráficamente. La carpeta Trazas del árbol del proyecto STEP 7 se usa para crear y administrar trazas.

Nota

Los datos de medición de seguimiento solo están disponibles dentro del proyecto STEP 7 y no pueden procesarse con otras herramientas.

En la figura siguiente se muestran los diferentes pasos de la función de trazas:



① Configure el seguimiento en el editor de trazas de STEP 7. Se pueden configurar las siguientes opciones:
- Valores de datos que deben grabarse
- Duración de la grabación
- Frecuencia de grabación
- Condición de disparo

② Transfiera la configuración de trazas de STEP 7 al PLC.

③ El PLC ejecuta el programa y, cuando se produce la condición de disparo, empieza a registrar los datos de trazas.

④ Transfiera los valores registrados del PLC a STEP 7.

⑤ Use las herramientas de STEP 7 para analizar los datos, visualizarlos gráficamente y guardarlos.

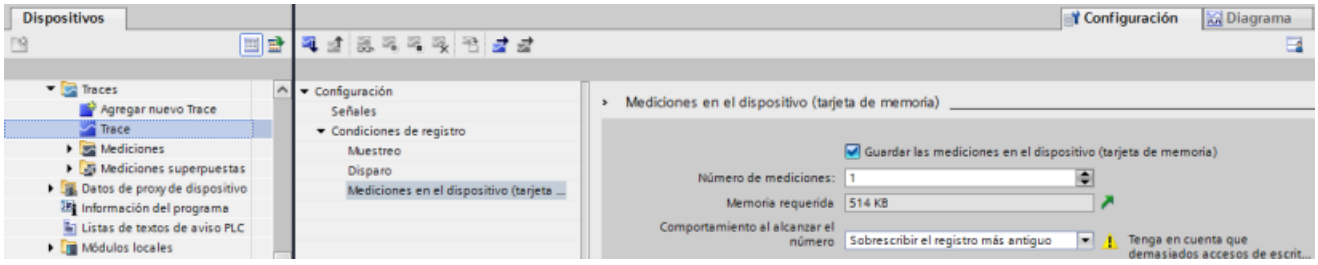
El S7-1200 soporta dos tareas de seguimiento con un máximo de 16 variables capturadas por evento de disparo. Cada orden de seguimiento proporciona 524288 bytes de memoria RAM para

el registro de valores de trace y datos generales asociados, por ejemplo direcciones de variables y sellos de tiempo.

Guardar mediciones de seguimiento en la Memory Card

La CPU S7-1200 solo puede guardar mediciones de seguimiento en la SIMATIC Memory Card. Si la CPU no dispone de Memory Card, registrará una entrada en el búfer de diagnóstico cuando el programa intente guardar mediciones de seguimiento (trace). La CPU limita el espacio asignado a las mediciones de seguimiento de tal modo que siempre debe quedar disponible 1 MB de memoria de carga externa. Si una medición de seguimiento ocupara más memoria de la máxima permitida, la CPU no guardará la medición y registrará una entrada en el búfer de diagnóstico.

Además, si se selecciona "Sobrescribir el registro más antiguo" en STEP 7, la escritura continua puede reducir la vida útil de la memoria de carga. Cuando se selecciona "Sobrescribir el registro más antiguo", la CPU reemplaza la medición más antigua por la más reciente una vez que haya guardado el número configurado de mediciones de seguimiento, y continúa siguiendo y guardando las mediciones. El hecho de sobrescribir las mediciones más antiguas ayuda a detectar problemas intermitentes.



La CPU soporta un máximo de 999 resultados de mediciones de seguimiento. Mientras la CPU guarda las mediciones de seguimiento en la memoria de carga externa, no comprueba la condición de disparo para la orden de seguimiento. Una vez la CPU termina de guardar las mediciones de seguimiento, vuelve a comprobar las condiciones de disparo.

Acceso a ejemplos

Consulte el sistema de información de STEP 7 para obtener información detallada sobre cómo programar trazas, cargar la configuración, cargar los datos de traza y visualizarlos en el analizador lógico. Es posible consultar ejemplos detallados en el capítulo "Uso de funciones online y diagnóstico > Uso de funciones de traza y analizador lógico".

Asimismo, el manual online "Industry Automation SIMATIC/SINAMICS Uso de la función Trace y de analizador lógico" (https://support.industry.siemens.com/cs/ww/es/view/64897128) también es una referencia excelente.

15.18 Determinar el tipo de condición de rotura de hilo en un módulo SM 1231

Tal y como se explica en el apartado Rangos de medida de entradas analógicas de tensión e intensidad (SB y SM) (Página 1352), el módulo SM 1231 devuelve un valor de entrada analógica de 32767 (16#7FFF) tanto para una condición de rotura de hilo como para una condición de desbordamiento. Si desea determinar cuál de las dos condiciones se ha producido, puede incluir una lógica en el programa STEP 7 para llevar a cabo esta determinación. El método para determinar el tipo de condición consta de las tareas siguientes:

- Cree un OB de alarma de error de diagnóstico que se llame siempre que se produzca un evento de diagnóstico entrante o saliente.
- Incluya una llamada de la instrucción RALRM.
- Configure un Array of Byte para el parámetro AINFO que incluya la información sobre el tipo de condición.
- Evalúe los bytes 32 y 33 de la estructura AINFO del RALRM_DB cuando la CPU dispare el OB de alarma de diagnóstico.

Crear un OB de alarma de error de diagnóstico

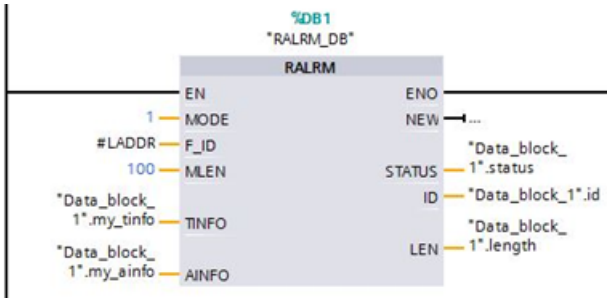
Para poder determinar cuándo se produce una condición de rotura de hilo, cree un OB de alarma de error de diagnóstico. La CPU llamará este OB siempre que se produzca un evento de diagnóstico entrante o saliente.

Cuando la CPU llame el OB de alarma de error de diagnóstico, el parámetro de entrada LADDR incluirá el identificador de hardware del módulo con el error. Encontrará el identificador de hardware del módulo SM 1231 en la configuración de dispositivos de STEP 7 para el módulo SM 1231.

Llamar la instrucción RALRM

Para programar la llamada de la instrucción RALRM, proceda del siguiente modo:

1. Agregue una llamada de RALRM en el programa STEP 7.
2. Ajuste el parámetro F_ID para el identificador de hardware en el parámetro LADDR del OB de alarma de error de diagnóstico.
3. Utilice un Array of Byte para los parámetros de entrada TINFO y AINFO. Utilice un tamaño de array de 34 bytes o más.



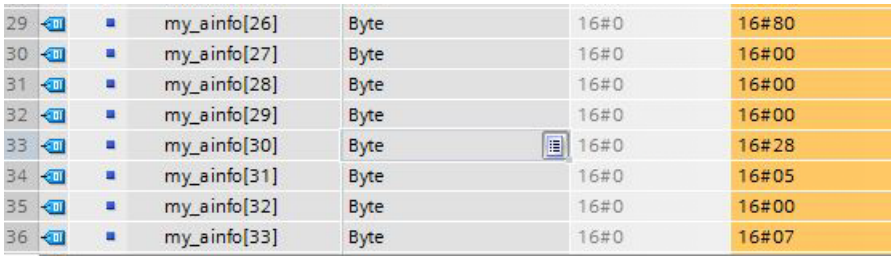
Interpretar AINFO después de una alarma de diagnóstico

El Array of Byte AINFO incluirá la información sobre el diagnóstico del módulo una vez se haya ejecutado el OB de alarma de error de diagnóstico.

Los bytes 0 - 25 son información de encabezado. Los bytes que pertenecen al diagnóstico del módulo son los siguientes:

Byte	Descripción	
26 y 27	Valor de palabra 16#8000 - indica que el diagnóstico es del estilo PROFINET	
28 y 29	Palabra que contiene el número de canal responsable de este diagnóstico	
30	Patrón de bits aaabb000 que indica el tipo de canal (aaa) y el tipo de error (bb)	
	aaa	bb
	000: reservado	00: reservado
	001: canal de entrada	01: error entrante
	010: canal de salida	10: error saliente
	011: canal de entrada/salida	11: error saliente, hay otros errores
31	Indicación del formato de datos 0: formato de datos libre 1: bit 2: dos bits 3: cuatro bits 4: byte 5: palabra (dos bytes) 6: palabra doble (cuatro bytes) 7: dos palabras dobles (ocho bytes)	
32 y 33	Palabra que define el tipo de error: 16#0000: reservado 16#0001: cortocircuito 16#0002: subtensión 16#0003: sobretensión 16#0004: sobrecarga 16#0005: sobretemperatura 16#0006: rotura de hilo 16#0007: límite superior rebasado por exceso 16#0008: límite inferior rebasado por defecto 16#0009: error	

Considere, por ejemplo, los bytes 26 - 33 de esta estructura 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

- La palabra de los bytes 26 y 27 es 16#8000, lo que significa que es un diagnóstico de estilo PROFINET.
- La palabra de los bytes 28 y 29 indica que es un diagnóstico para canal 0 o el módulo.
- El byte 30 es 16#28 que, cuando se interpreta como patrón de bits aaa bb 00, es 001 01 000. El valor indica que este diagnóstico es para un canal de entrada y es un error entrante.
- El byte 31 es 5 e indica un valor de palabra.
- El valor de palabra de los bytes 32 y 33 es 16#0007, lo que significa que se ha rebasado por exceso el límite superior.

Así pues, cuando se obtiene la información de AINFO desde un evento de alarma de error de diagnóstico es posible determinar el origen del evento de diagnóstico.

15.19 Realizar una copia de seguridad de la CPU y restaurarla

15.19.1 Opciones de copia de seguridad y restauración

Con el tiempo se llevarán a cabo numerosos cambios en el sistema de automatización, por ejemplo agregar dispositivos nuevos, reemplazar dispositivos existentes o adaptar el programa de usuario. Si estos cambios provocaran comportamientos no deseados, existe la posibilidad de restaurar la planta de automatización a una versión anterior, siempre que se haya realizado una copia de seguridad. STEP 7 y la CPU S7-1200 ofrecen diferentes opciones para realizar una copia de seguridad de la configuración hardware y del software y para restaurarlos.

Opciones de copia de seguridad

La tabla inferior ofrece un resumen de las opciones de copia de seguridad y restauración de las CPU S7:

	Instantánea de los valores observados	**Cargar desde el dispositivo (software)**	**Cargar dispositivo como estación nueva (hardware y software)**	**Descargar copia de seguridad desde un dispositivo online**
Aplicación	Restablecer un estado concreto de un bloque de datos. Los valores reales de los bloques de datos con sello de tiempo se aceptan en el proyecto.	Cargar bloques en el proyecto desde una CPU.	Cargar la configuración hardware y el software en el proyecto desde un dispositivo.	Crear una copia de seguridad completa de una CPU como punto de restauración. La copia de seguridad es coherente y no puede modificarse ni abrirse.
Requisitos	La CPU existe en un proyecto. Los bloques de datos deben ser idénticos tanto online como offline.	La CPU existe en el proyecto.	El dispositivo está disponible en el catálogo de hardware del TIA Portal. Los archivos HSP o GSD necesarios están instalados.	-
Posible en estado	RUN, STOP	RUN, STOP	RUN, STOP	STOP

	Instantánea de los valores observados	Cargar desde el dispositivo (software)	Cargar dispositivo como estación nueva (hardware y software)	Descargar copia de seguridad desde un dispositivo online
Posible para CPU F	Sí	Sí	No	Sí
La copia de seguridad puede editarse	Sí	Sí	Sí	No

Contenido de la copia de seguridad

La tabla siguiente muestra los datos que pueden descargarse y de los que pueden crearse copias de seguridad, así como las opciones respectivas:

	Instantánea de los valores observados	Cargar desde el dispositivo (software)	Cargar dispositivo como estación nueva (hardware y software)	Descargar copia de seguridad desde un dispositivo online
Valores reales de los bloques de datos	Instantánea posible	Descarga posible	Descarga posible	Copia de seguridad posible
Bloques de software	-	Descarga posible	Descarga posible	Copia de seguridad posible
Variables PLC (nombres de variables y constantes)	-	Descarga posible	Descarga posible	Copia de seguridad posible
Objetos tecnológicos	-	Descarga posible	Descarga posible	Copia de seguridad posible
Configuración hardware	-	-	Descarga posible	Copia de seguridad posible
Tablas de observación (servidor web)	-	-	Descarga no posible	Copia de seguridad posible
Datos locales, marcas, temporizadores, contadores y sinópticos del proceso	Instantánea no posible	Descarga no posible	Descarga no posible	Copia de seguridad posible
Ficheros y recetas (PLC)	-	-	-	Copia de seguridad posible
Datos generales en la SIMATIC Memory Card, p. ej. ayuda para bloques de programa o archivos GSD	-	-	-	Copia de seguridad posible

Consideraciones especiales al realizar una copia de seguridad de los valores reales

El tipo de copia de seguridad "Cargar backup del dispositivo online" realiza una copia de seguridad de los valores reales de las variables que están ajustadas como remanentes. Para garantizar la coherencia de los datos remanentes, desactive todos los accesos de escritura a datos remanentes mientras se realiza la copia de seguridad.

Una transición de STOP a RUN pone los valores reales de los datos no remanentes a sus valores de arranque. La copia de seguridad de una CPU contiene únicamente los valores de arranque de los datos no remanentes.

15.19.2 Realizar una copia de seguridad de una CPU online

Realizar una copia de seguridad de la configuración puede ser útil si se desea regresar a una configuración concreta. Es posible restablecer la configuración actual más adelante.

Requisitos

Es posible crear todas las copias de seguridad que se desee y guardar varias configuraciones de una CPU. Para realizar una copia de seguridad deben cumplirse los requisitos siguientes:

- La CPU ya está creada en el proyecto de STEP 7.
- La CPU está conectada directamente a la programadora/el PC a través de la interfaz PROFINET de la CPU. Las operaciones de copia de seguridad y restauración no soportan las interfaces PROFIBUS de los CM.
- La CPU está online. (Si no hay conexión online, el proceso de copia de seguridad establece una conexión online.)
- La CPU está en estado STOP. (Si la CPU no está en estado operativo STOP, el proceso de copia de seguridad pide permiso para poner la CPU a STOP.)

Procedimiento

Para crear una copia de seguridad de la configuración actual de una CPU, proceda del siguiente modo:

1. Seleccione la CPU en el árbol del proyecto.
2. Seleccione el comando "Cargar backup del dispositivo online" del menú "Online".
   En caso necesario habrá que introducir la contraseña de acceso de lectura a la CPU y confirmar que la CPU debe pasar a STOP.

Resultado

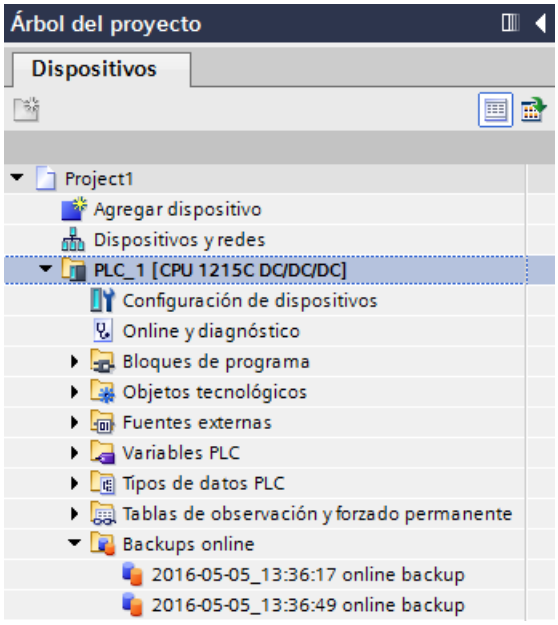
Las copias de seguridad se guardan con el nombre de la CPU y la fecha y hora de la copia de seguridad. La copia de seguridad incluye todos los datos necesarios para restaurar una configuración concreta de una CPU. La CPU guarda los datos siguientes en la copia de seguridad:

- Contenido de la Memory Card, si la hubiera
- Áreas de memoria remanente de los bloques de datos, contadores y marcas
- El resto del contenido de la memoria remanente, como parámetros de dirección IP

La copia de seguridad contiene los valores actuales de la CPU, pero no incluye el búfer de diagnóstico.

La copia de seguridad no contiene la contraseña para la protección de datos de configuración confidenciales del PLC.

La copia de seguridad se encuentra en la carpeta "Backups online" del árbol del proyecto de la CPU. La figura siguiente muestra una CPU S7-1200 para la que se han creado dos copias de seguridad:



Nota

Tenga en cuenta que también es posible crear una copia de seguridad de la CPU online desde la SIMATIC Automation Tool (SAT) o la página web estándar de copia de seguridad online del servidor web (Página 879).

Cuando se hacen copias de seguridad desde STEP 7, STEP 7 guarda los archivos dentro del proyecto de STEP 7. Cuando se hacen copias de seguridad de archivos desde el servidor web, el PC o dispositivo guarda los archivos de copia de seguridad en la carpeta predeterminada para descargas. No es posible restaurar archivos de copia de seguridad de STEP 7 desde el servidor web ni tampoco a la inversa. Sin embargo, sí es posible guardar archivos de copia de seguridad de STEP 7 directamente en la carpeta de descargas del PC o dispositivo. De hacerse así, después podrán restaurarse estos archivos desde el servidor web.

Guardar archivos de copia de seguridad en el PC o dispositivo

Para guardar un archivo de copia de seguridad en el PC o dispositivo, proceda del siguiente modo:

1. Haga clic con el botón derecho del ratón en un archivo de la carpeta "Backups online" del árbol del proyecto.
2. Seleccione "Guardar como" en el menú contextual.
3. Navegue hasta la carpeta en la que desee guardar el archivo, p. ej. la carpeta predeterminada para descargas del PC o dispositivo.
4. Haga clic en Guardar.

15.19.3 Restaurar una CPU

Si se ha realizado previamente una copia de seguridad de la configuración de la CPU, es posible transferir dicha copia a la CPU. La CPU pasa a estado operativo STOP mientras se restaura la copia de seguridad. Si se ha configurado un nivel de acceso para la CPU, habrá que introducir la contraseña de acceso de lectura a la CPU.

La copia de seguridad no contiene la contraseña para la protección de datos de configuración confidenciales del PLC.

⚠ **ADVERTENCIA**
Restaurar copias de seguridad con contenido desconocido Si se restauran copias de seguridad con contenido desconocido pueden producirse daños y lesiones graves en caso de funcionamientos incorrectos o errores de programa. Asegúrese de que la copia de seguridad contiene una configuración con contenido conocido.

Restaurar una copia de seguridad en una CPU con protección de datos de configuración confidenciales del PLC

Si la CPU tiene protección de datos de configuración confidenciales del PLC (Página 161), asegúrese de que la contraseña configurada en el archivo de copia de seguridad para la protección de datos de configuración confidenciales del PLC coincide con la de la CPU.

Si las contraseñas no coinciden, la CPU no puede pasar al estado operativo RUN.

Si se intenta restablecer un archivo de copia de seguridad con una contraseña para la protección de datos de configuración confidenciales del PLC diferente que la de la CPU, la copia de seguridad se restaurará correctamente. Sin embargo, la CPU se reiniciará en un estado de error dado que la contraseña de protección de los datos de configuración confidenciales del PLC de la CPU no coincidía con la del proyecto restaurado en la CPU.

En ese caso, debe ajustar una protección de los datos de configuración confidenciales del PLC en la CPU que coincida con la del proyecto restaurado. Se puede utilizar uno de los siguientes métodos para establecer o borrar la contraseña para la protección de datos de configuración confidenciales del PLC que se encuentra en la CPU:

- SIMATIC Automation Tool V4.0 SP3 o superior
- TIA Portal V17 o superior, Online y diagnóstico (Página 1205)
- SIMATIC Memory Card (Página 130)

Requisitos

Para restaurar una copia de seguridad deben cumplirse los requisitos siguientes:

- El proyecto de STEP 7 incluye una configuración para la CPU y una copia de seguridad creada con anterioridad.
- La CPU está conectada directamente a la programadora a través de la interfaz PROFINET de la CPU.

- La CPU está en estado operativo STOP.
- Si se ha configurado un nivel de acceso (Página 163), se conoce la contraseña para el pleno acceso a la CPU.

Procedimiento

Para restaurar una copia de seguridad, proceda del siguiente modo:

1. Abra la CPU en el árbol del proyecto para visualizar los objetos de nivel inferior.
2. Seleccione la copia de seguridad que desee restaurar en la carpeta "Backups online".
3. En el menú "Online" seleccione el comando "Cargar en dispositivo".
   - Si previamente se ha establecido una conexión online (Página 1201), se abre el cuadro de diálogo "Vista preliminar". Este cuadro de diálogo muestra alarmas y recomienda acciones necesarias para la operación de carga.
   - Si previamente no se ha establecido ninguna conexión online, se abre el cuadro de diálogo "Carga avanzada" y antes que nada debe seleccionarse la interfaz desde la que debe establecerse la conexión online con la CPU.
4. Compruebe las alarmas en el cuadro de diálogo "Vista preliminar" y seleccione las acciones en la columna "Acción", si fuera necesario.
5. Haga clic en el botón "Cargar" (el botón "Cargar" está activo en el momento en que la descarga es posible).
6. STEP 7 restaura la copia de seguridad en la CPU. Desde el cuadro de diálogo "Resultados de la operación de carga" puede comprobarse si la operación de carga ha sido correcta y realizar las acciones posteriores que sean necesarias.
7. Después de revisar el cuadro de diálogo "Resultados de la operación de carga", haga clic en el botón "Finalizar".
   Si se solicita, introduzca la contraseña de acceso completo a la CPU y confirme que la CPU puede pasar a STOP.
   STEP 7 restaura el contenido de la copia de seguridad en la CPU y la reinicia.

Nota

Tenga en cuenta que también es posible restaurar una copia de seguridad de la CPU desde la página web estándar de copia de seguridad online del servidor web (Página 879).

Datos técnicos

A

A.1 Página web de Siemens Online Support

En la página web de Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/) encontrará información técnica relacionada con estos productos.

A.2 Datos técnicos generales

Homologaciones

El diseño del sistema de automatización S7-1200 cumple las siguientes normas y especificaciones de ensayo. Los criterios de test del sistema de automatización S7-1200 se basan en estas normas y especificaciones de ensayo.

Nótese que no todos los modelos S7-1200 pueden certificarse según esas normas y el estado de certificación puede cambiar sin notificación. Es responsabilidad del usuario determinar las certificaciones aplicables consultando las clasificaciones marcadas en el producto. Contacte con el representante de Siemens más próximo para obtener una lista de las homologaciones actuales con las referencias respectivas.

Homologación CE

CE

El sistema de automatización S7-1200 satisface los requisitos y objetivos relacionados con la seguridad según las directivas CE indicadas a continuación y cumple las normas europeas (EN) armonizadas para controladores programables publicadas en los Diarios Oficiales de la Unión Europea.

- Directiva CE 2006/95/CE (Directiva de baja tensión) "Material eléctrico destinado a utilizarse con determinados límites de tensión"
  - EN 61131-2 Autómatas programables - Requisitos y ensayos de los equipos
- Directiva CE 2004/108/CE (Directiva CEM) "Compatibilidad electromagnética"
  - Norma de emisión
    EN 61000-6:+A1: Entornos industriales
  - Norma de inmunidad
    EN 61000-6-2: Entornos industriales
- Directiva CE 94/9/CE (ATEX) "Equipos y sistemas de protección para uso en atmósferas potencialmente explosivas"
  - EN 60079-0:+A11
  - EN 60079-15: Tipo de protección 'n'

La Declaración de conformidad CE se encuentra a disposición de las autoridades competentes en:

Siemens AG
Digital Industries
Factory Automation
DI FA AS SYS
Postfach 1963
D-92209 Amberg
Alemania

Homologación cULus

Underwriters Laboratories Inc. cumple:

- Underwriters Laboratories, Inc.: UL 508 Listed (Industrial Control Equipment)
- Canadian Standards Association: CSA C22.2 Number 142 (Process Control Equipment)

Nota

La gama SIMATIC S7-1200 cumple la norma CSA.

El logotipo cULus indica que Underwriters Laboratories (UL) ha examinado y certificado el S7-1200 según las normas UL 508 y CSA 22.2 No. 142.

Homologación FM

FM Approvals
Approval Standard Class Number 3600, 3611 (ANSI/UL 121201), 3810 (ANSI/UL 61010-1), CSA Standard C22.2 No. 0-10, C22.2 No. 213, C22.2 No. 61010-1
Approved for use in:
Class I, Division 2, 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 per CEC 18-150 [CA]

EXCEPCIÓN IMPORTANTE: Consulte en las especificaciones técnicas el número de entradas y salidas que pueden estar activadas simultáneamente. Algunos modelos tienen características reducidas para Ta = 60 °C.

⚠ ADVERTENCIA

Si se sustituyen componentes, podría perderse la idoneidad para Class I, Division 2 y Zone 2.

Solo un centro de asistencia Siemens autorizado puede reparar estas unidades.

Homologación IECEx

IEC 60079-0: Atmósferas explosivas - Requisitos generales

IEC 60079-15: Electrical Apparatus for Potentially Explosive Atmospheres
Type of protection 'nA'

IEC FMG 14.0012X
Ex nA IIC T3 Gc

En el producto puede indicarse información de clasificación IECEx con la información sobre ubicación peligrosa FM.

Solo están homologados los productos con clasificación IECEx. Contacte con el representante de Siemens más próximo para obtener una lista de las homologaciones actuales con las referencias respectivas.

Los modelos de tipo relé no están incluidos en la homologación IECEx.

Consulte la temperatura en el marcado específico del producto.

Monte los módulos en una envolvente adecuada que proporcione un grado de protección mínimo IP54 conforme a IEC 60079-15.

Homologación ATEX



La homologación ATEX solo es válida para los modelos DC. La homologación ATEX no es válida para los modelos AC y de relé.

EN 60079-0: Atmósferas explosivas - Requisitos generales

EN 60079-15: Material eléctrico para atmósferas de gas explosivas;
Tipo de protección 'nA'
II 3 G Ex nA IIC T4 o T3 Gc

Condiciones especiales para un uso seguro:

Instale los módulos en una caja adecuada con un grado de protección mínimo de IP54 conforme a EN 60529, o en una ubicación que aporte un grado de protección equivalente.

Los cables y conductores conectados deben ser aptos para la temperatura real medida en condiciones nominales.

Deben tomarse precauciones para que no se exceda la tensión nominal en los bornes de alimentación cuando se produzcan perturbaciones transitorias de más de 119 V.

Homologación CCCEx



Conforme a GB 3836.8 (Atmósfera explosiva - Parte 8: Equipos de protección según modo de protección "n")

GB 3836.1 (Atmósfera explosiva - Parte 1: Equipos - Requisitos generales)

Ex nA IIC T3 Gc

Condiciones específicas de uso seguro:

- El equipo solo debe usarse en un área con un grado de contaminación no superior al 2 según lo definido en la norma GB/T 17935.1-2008.
- El equipo solo debe instalarse en una envolvente que ofrezca un grado de protección no inferior a IP54 conforme a la norma GB 3836.8-2014.
- La protección transitoria suministrada debe estar ajustada a un nivel que no exceda 140 % del valor pico de tensión asignada en los bornes de alimentación del equipo.

- Los módulos marcados con un asterisco (*) están reducidos a 55 °C en montaje horizontal y a 45 °C en montaje vertical cuando están configurados para carga máxima, además no deben tener canales adyacentes. En el manual de usuario puede consultar la información de derating.
- Consulte las instrucciones para más información.

Australia y Nueva Zelanda - Marcado RCM (Regulatory Compliance Mark)



El sistema de automatización S7-1200 cumple los requisitos de los estándares según AS/NZS 61000.6.4 e IEC 610000-6-4 (clase A).

Homologación coreana



El sistema de automatización S7-1200 cumple los requisitos de la homologación coreana (KC Mark). Este sistema ha sido definido como equipo clase A para aplicaciones industriales y no está previsto para uso doméstico.

Homologación para la Unión Aduanera Euroasiática (Bielorrusia, Kazajstán, Federación Rusa)



EAC (Eurasion Conformity): Declaración de conformidad según el Reglamento Técnico de la Comunidad Económica (TR CU)

Homologación naval

Los productos S7-1200 se someten con regularidad a pruebas para obtener homologaciones especiales para aplicaciones y mercados específicos. Contacte con el representante de Siemens más próximo para obtener una lista de las homologaciones actuales con las referencias respectivas.

Sociedades de clasificación:

- ABS (American Bureau of Shipping): EE.UU.
- BV (Bureau Veritas): Francia
- DNV (Det Norske Veritas): Noruega
- GL (Germanischer Lloyd): Alemania
- LRS (Lloyds Register of Shipping): Inglaterra
- Class NK (Nippon Kaiji Kyokai): Japón
- Korean Register of Shipping: Corea
- CSS (China Classification Society): China

Entornos industriales

El sistema de automatización S7-1200 está diseñado para ser utilizado en entornos industriales.

Tabla A-1 Entornos industriales

Campo de aplicación	Requisitos de emisión	Requisitos de inmunidad
Industrial	EN 61000-6-4:2007+A1	EN 61000-6-2:2005

Compatibilidad electromagnética

La compatibilidad electromagnética (también conocida por sus siglas CEM o EMC) es la capacidad de un dispositivo eléctrico para funcionar de forma satisfactoria en un entorno electromagnético sin causar interferencias electromagnéticas (EMI) sobre otros dispositivos eléctricos de ese entorno.

Tabla A-2 Inmunidad según EN 61000-6-2

Compatibilidad electromagnética - Inmunidad según EN 61000-6-2	
EN 61000-4-2 Descargas electrostáticas	Descarga en el aire de 8 kV en todas las superficies Descarga en contactos de 6 kV en las superficies conductoras expuestas
EN 61000-4-3 Prueba de inmunidad de campos electromagnéticos de radiofrecuencia radiada	80 a 1000 MHz, 10 V/m, 80% AM a 1 kHz 1,4 a 2,0 GHz, 3 V/m, 80% AM a 1 kHz 2,0 a 2,7 GHz, 1 V/m, 80% AM a 1 kHz
EN 61000-4-4 Transitorios eléctricos rápidos	2 kV, 5 kHz con red de conexión a la alimentación AC y DC 2 kV, 5 kHz con borne de conexión a las E/S
EN 6100-4-5 Inmunidad a sobretensiones	Sistemas AC - 2 kV en modo común, 1 kV en modo diferencial Sistemas DC - 2 kV en modo común, 1 kV en modo diferencial Para los sistemas DC, consulte Inmunidad a sobretensiones a continuación.
EN 61000-4-6 Perturbaciones conducidas	150 kHz a 80 MHz, 10 V RMS, 80% AM a 1kHz
EN 61000-4-11 Inmunidad a cortes e interrupciones breves	Sistemas AC 0% durante 1 ciclo, 40% durante 12 ciclos y 70% durante 30 ciclos a 60 Hz

Tabla A-3 Emisiones conducidas y radiadas según EN 61000-6-4

Compatibilidad electromagnética - Emisiones conducidas y radiadas según EN 61000-6-4		
Emisiones conducidas EN 55016, clase A, grupo 1	De 0,15 MHz a 0,5 MHz	<79dB (µV) casi cresta; <66 dB (µV) valor medio
	De 0,5 MHz a 5 MHz	<73dB (µV) casi cresta; <60 dB (µV) valor medio
	De 5 MHz a 30 MHz	<73dB (µV) casi cresta; <60 dB (µV) valor medio

Compatibilidad electromagnética - Emisiones conducidas y radiadas según EN 61000-6-4		
Emisiones radiadas EN 55016, clase A, grupo 1	De 30 MHz a 230 MHz	<40dB (µV/m) casi cresta; medido a 10 m
	De 230 MHz a 1 GHz	<47dB (µV/m) casi cresta; medido a 10 m
	1 GHz a 3 GHz	< 76dB (µV/m) casi cresta; medido a 10 m

Inmunidad a sobretensiones

Los sistemas de cableado sujetos a sobretensiones por acoplamiento de rayos deben estar equipados con protección externa. Una especificación para la evaluación de la protección frente a sobrecargas de rayo se recoge en EN 61000-4-5, con límites operacionales establecidos en EN 61000-6-2. Las CPU S7-1200 DC y los módulos de señales requieren protección externa para garantizar el funcionamiento seguro en caso de sobretensiones definidas en este estándar.

A continuación aparece una lista de algunos dispositivos que soportan la protección necesaria de inmunidad a ondas de choque. Estos dispositivos solo proporcionan protección si están correctamente instalados conforme a las recomendaciones del fabricante. También pueden utilizarse dispositivos de otros fabricantes con especificaciones iguales o mejores:

Tabla A-4 Dispositivos que soportan la protección de inmunidad a sobretensiones

Subsistema	**Dispositivo de protección**
Alimentación de +24 V DC	BLITZDUCTOR VT, BVT AVD 24, referencia 918 422
Industrial Ethernet	DEHNpatch DPA M CLE RJ45B 48, referencia 929 121
RS-485	BLITZDUCTOR XT, Basic Unit BXT BAS, referencia 920 300
	BLITZDUCTOR XT, módulo BXT ML2 BD HFS 5, referencia 920 271
RS-232	BLITZDUCTOR XT, Basic Unit BXT BAS, referencia 920 300
	BLITZDUCTOR XT, módulo BXT ML2 BE S 12, referencia 920 222
Entradas digitales de +24 V DC	DEHN, Inc., tipo DCO SD2 E 24, referencia 917 988
Salidas digitales de +24 V DC y alimentación de sensores	DEHN, Inc., tipo DCO SD2 E 24, referencia 917 988
IO analógicas	DEHN, Inc., tipo DCO SD2 E 12, referencia 917 987
Salidas de relé	ninguna requerida

Condiciones ambientales

Tabla A-5 Transporte y almacenamiento

Condiciones ambientales: transporte y almacenamiento	
EN 60068-2-2, ensayo Bb, calor seco y EN 60068-2-1, ensayo Ab, frío	De -40 °C a +70 °C
EN 60068230, ensayo Dd, calor húmedo	De 25 °C a 55 °C, 95% de humedad
EN 60068-2-14, ensayo Na, choque de temperatura	De -40 °C a +70 °C, tiempo de permanencia 3 horas, 2 ciclos
EN 60068232, caída libre	0,3 m, 5 veces, embalado para embarque
Presión atmosférica	De 1140 a 660 hPa (equivale a una altitud de -1000 a 3500 m)

Tabla A-6 Condiciones ambientales climáticas

Condiciones ambientales: condiciones climáticas	
El sistema de automatización S7-1200 está diseñado para ser instalado al abrigo de la intemperie. Las condiciones de servicio se basan en los requisitos estipulados por la norma DIN IEC 60721-3-3: • Clase 3M3 (requisitos mecánicos) • Clase 3K3 (requisitos climáticos)	
Rango de temperatura ambiente (aire de entrada 25 mm bajo la unidad)	De -20 °C a 60 °C en montaje horizontal De -20 °C a 50 °C en montaje vertical 95% de humedad sin condensación A no ser que se especifique otra cosa
Presión atmosférica	De 1140 a 795 hPa (equivale a una altitud de -1000 a 2000 m)
Concentración de contaminantes	SO_2: < 0,5 ppm; H_2S: < 0,1 ppm; RH < 60% sin condensación
	ISA-S71.04 nivel de severidad G1, G2, G3
EN 60068214, ensayo Nb, variación de temperatura	De 0 °C a 60 °C
EN 60068227, choque mecánico	15 g, 11 ms impulso, 6 choques en c/u de 3 ejes
EN 6006826, vibración sinusoidal	Montaje en perfil DIN: 3,5 mm de 5 a 9 Hz, 1G de 8,4 a 150 Hz Montaje en panel: 7,0 mm de 5 a 8,4 Hz, 2G de 8,4 a 150 Hz 10 barridos por eje, 1 octava por minuto

Grado de contaminación/categoría de sobretensión conforme a IEC 61131-2

- Grado de contaminación 2
- Categoría de sobretensión: II

Clase de protección

- Clase de protección II según EN 61131-2 (el conductor protector no se requiere)

Grado de protección

- Protección mecánica IP20, EN 60529
- Protege los dedos contra el contacto con alta tensión, según ensayos realizados con sondas estándar. Se requiere protección externa contra polvo, impurezas, agua y objetos extraños de < 12,5mm de diámetro.

Tensiones nominales

Tabla A-7 Tensiones nominales

Tensión nominal	**Tolerancia**
24 V DC	De 20,4 V DC a 28,8 V DC
120/230 V AC	De 85 V AC a 264 V AC, de 47 a 63 Hz

Nota

Cuando un contacto mecánico activa la potencia de salida en la CPU S7-1200 o en cualquier módulo de ampliación digital, envía una señal "1" a las salidas digitales durante unos 50 microsegundos. Esto podría originar un funcionamiento inesperado de los equipos o del proceso, lo que podría causar la muerte o lesiones graves al personal o daños a los equipos. Esto debe preverse, especialmente si se utilizan dispositivos que reaccionan a impulsos de corta duración.

Protección contra inversión de polaridad

Se suministra circuitería de protección contra inversión de polaridad en cada par de bornes de alimentación de +24 V DC o de alimentación de entrada de usuario para CPU, módulos de señales (SM) y Signal Boards (SB). No obstante, el sistema puede sufrir daños si se cablean pares de bornes distintos en polaridades opuestas.

Algunos puertos de entrada de la alimentación de 24 V DC del sistema S7-1200 están interconectados, teniendo un circuito lógico común que conecta varios bornes M. Por ejemplo, los circuitos siguientes están interconectados si no tienen aislamiento galvánico según las hojas de datos técnicos: la fuente de alimentación de 24 V DC de la CPU, la alimentación de sensores de la CPU, la entrada de alimentación de la bobina de relé de un SM o la fuente de alimentación de una entrada analógica no aislada. Todos los bornes M sin aislamiento galvánico deben conectarse al mismo potencial de referencia externo.

⚠ ADVERTENCIA

Si los bornes M sin aislamiento galvánico se conectan a diferentes potenciales de referencia, circularán corrientes indeseadas que podrían averiar o causar reacciones inesperadas en el PLC y los equipos conectados.

Si no se cumplen estas directrices, es posible que se produzcan averías o reacciones inesperadas que podrían causar la muerte, lesiones corporales graves y/o daños materiales.

Asegúrese que todos los bornes M sin aislamiento galvánico de un sistema S7-1200 están conectados al mismo potencial de referencia.

Salidas DC

No se suministra circuitería de protección contra cortocircuitos en las salidas DC de las CPU, módulos de señales (SM) y Signal Boards (SB).

Vida útil de los relés

La figura siguiente muestra los datos típicos de rendimiento estimados a partir de ensayos. El rendimiento real puede variar dependiendo de la aplicación. Un circuito de protección externo adaptado a la carga mejorará la vida útil de los contactos. Los contactos normalmente cerrados tienen una vida útil típica de aproximadamente un tercio de los contactos normalmente abiertos en condiciones inductivas y de carga de lámpara.

Un circuito de protección externo aumentará la vida útil de los contactos.

Tabla A-8 Datos de rendimiento típicos

Datos para la selección de un actuador				
Corriente térmica continua		2 A máx.		
Capacidad de conmutación y vida útil de los contactos				
	Para carga óhmica	Tensión	Intensidad	Número de ciclos de operación (típico)
		24 V DC	2,0 A	0,1 millón
		24 V DC	1,0 A	0,2 millones
		24 V DC	0,5 A	1,0 millón
		48 V AC	1,5 A	1,5 millones
		60 V AC	1,5 A	1,5 millones
		120 V AC	2,0 A	1,0 millón
		120 V AC	1,0 A	1,5 millones
		120 V AC	0,5 A	2,0 millones
		230 V AC	2,0 A	1,0 millón
		230 V AC	1,0 A	1,5 millones
		230 V AC	0,5 A	2,0 millones
	Para carga inductiva (conforme a IEC 947-5-1 DC13/AC15)	Tensión	Intensidad	Número de ciclos de operación (típico)
		24 V DC	2,0 A	0,05 millones
		24 V DC	1,0 A	0,1 millón
		24 V DC	0,5 A	0,5 millones
		24 V AC	1,5 A	1,0 millón
		48 V AC	1,5 A	1,0 millón
		60 V AC	1,5 A	1,0 millón
		120 V AC	2,0 A	0,7 millones
		120 V AC	1,0 A	1,0 millón
		120 V AC	0,5 A	1,5 millones
		230 V AC	2,0 A	0,7 millones
		230 V AC	1,0 A	1,0 millón
		230 V AC	0,5 A	1,5 millones
Activación de una entrada digital		Posible		
Frecuencia de conmutación				
	Mecánica	Máx. 10 Hz		
	Con carga óhmica	Máx. 1 Hz		
	Con carga inductiva (conforme a IEC 947-5-1 DC13/AC15)	Máx. 0,5 Hz		
	Con carga de lámpara	Máx. 1 Hz		

Retención de la memoria interna de la CPU

- Vida útil de los datos remanentes y de registro: 10 años
- Datos remanentes tras desconexión (POWER OFF), resistencia (ciclos de escritura): 2 millones de ciclos
- Datos del registro: resistencia (ciclos de escritura): 500 millones de entradas

Nota

Efecto de los registros en la memoria interna de la CPU

Cada escritura en el registro consume como mínimo 2 KB de memoria. Si el programa escribe pequeñas cantidades de datos con frecuencia, estará consumiendo como mínimo 2 KB de memoria en cada escritura. Es conveniente acumular los pequeños elementos de datos en un bloque de datos (DB), y escribir el bloque de datos en el registro en intervalos menos frecuentes.

Si el programa escribe muchas entradas en el registro con mucha frecuencia, es aconsejable utilizar una tarjeta de memoria SD extraíble.

A.3 Asignación de pines del puerto X1 de la interfaz PROFINET

La CPU S7-1200 se conecta a la red PROFINET por medio de un conector hembra RJ45 estándar. La asignación de pines del conector depende del tipo de CPU.

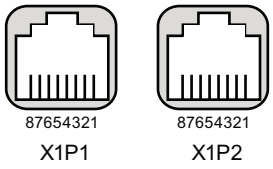
CPU de un puerto

Las CPU de un puerto (CPU 1211C, CPU 1212C y CPU 1214C) tienen una configuración de pines Ethernet MDI estándar, del siguiente modo:

Pin	Nombre de señal	Descripción	Asignación del conector hembra RJ45
1	TD+	Datos de transmisión	 X1P1
2	TD-		
3	RD+	Datos de recepción	
4	GND	Tierra	
5	GND		
6	RD-	Datos de recepción	
7	GND	Tierra	
8	GND		

CPU de dos puertos

Las CPU de dos puertos (CPU 1215C y CPU 1217C) tienen una configuración de pines Ethernet MDI-X estándar, del siguiente modo:

Pin	Nombre de señal	Descripción	Asignación del conector hembra RJ45
1	RD+	Datos de recepción	
2	RD-		
3	TD+	Datos de transmisión	
4	GND	Tierra	
5	GND		
6	TD-	Datos de transmisión	
7	GND	Tierra	
8	GND		

Nota

CPU Dual-Port

El número que aparece debajo del pin indica que la CPU dual port no tiene ningún cruce entre los pines. Hay un switch Ethernet interno en las unidades: las parejas TD+/- y RD+/- no están cruzadas internamente.

Autonegotiation

Si la configuración del puerto permite la autonegotiation, la CPU S7-1200 detecta automáticamente el tipo de cable e intercambia las líneas de transmisión y recepción, si fuera necesario. Si la configuración del puerto desactiva la autonegotiation, la CPU también desactiva este intercambio automático. Los ajustes de autonegotiation del puerto se configuran en el cuadro de diálogo de opciones del puerto del TIA Portal. Es una opción avanzada específica del puerto para la interfaz PROFINET (X1) de las propiedades de la CPU. Encontrará más información en "Configurar el puerto PROFINET" en el apartado 11.2.3.4: Configurar una dirección IP para una CPU del proyecto" (Página 603).

A.4 CPU 1211C

A.4.1 Especificaciones generales y propiedades

Tabla A-9 Especificaciones generales

Datos técnicos	CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Referencia	6ES7211-1BE40-0XB0	6ES7211-1HE40-0XB0	6ES7211-1AE40-0XB0
Dimensiones A x A x P (mm)	90 x 100 x 75		

Datos técnicos	**CPU 1211C AC/DC/Relay**	**CPU 1211C DC/DC/relé**	**CPU 1211C DC/DC/DC**
Peso de envío	420 gramos	380 gramos	370 gramos
Disipación de potencia	10 W	8 W	
Intensidad disponible (bus CM)	750 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	300 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		

Tabla A-10 Propiedades de la CPU

Datos técnicos		**Descripción**
Memoria de usuario (Consulte "Datos técnicos generales" (Página 1245), "Remanencia de la memoria interna de la CPU".)	Trabajo	50 KB
	Carga	1 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	14 KB
E/S digitales integradas		6 entradas/4 salidas
E/S analógicas integradas:		2 entradas
Tamaño de la memoria imagen de proceso		1024 bytes de entradas (I)/1024 bytes de salidas (Q)
Área de marcas (M)		4096 bytes
Memoria temporal (local)		• 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados)
Ampliación con módulos de señales		Ninguno
Ampliación con SB, CB o BB		1 máx.
Ampliación con módulos de comunicación		3 CM máx.
Contadores rápidos		Se han configurado un máximo de 6 para usar cualquier entrada integrada o de SB. Consulte "Asignación de los pines de entrada de hardware" (Página 563) para la CPU 1211C: Asignaciones de direcciones predeterminadas del HSC. 100/[1]80 kHz (de Ia.0 a Ia.5)
Salidas pulsadas[2]		Se han configurado un máximo de 4 para usar cualquier salida integrada o de SB 100 kHz (de Qa.0 a Qa.3)
Entradas de captura de impulsos		6
Alarmas de retardo		4 en total con resolución de 1 ms
Alarmas cíclicas		4 en total con resolución de 1 ms
Alarmas de flanco		6 ascendentes y 6 descendentes (10 y 10 con Signal Board opcional)
Memory Card		SIMATIC Memory Card (opcional)
Precisión del reloj en tiempo real		+/- 60 segundos/mes
Tiempo de respaldo del reloj de tiempo real		20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)

[1] La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

Tabla A-11 Rendimiento

Tipo de instrucción		**Velocidad de ejecución**	
		Direccionamiento directo (I, Q y M)	**Accesos a DB**
Boole		0,08 µs/instrucción	
Desplaza-miento	Move_Bool	0,3 µs/instrucción	1,17 µs/instrucción
	Move_Word	0,137 µs/instrucción	1,0 µs/instrucción
	Move_Real	0,72 µs/instrucción	1,0 µs/instrucción
Funciones matemáticas	Sumar números reales	1,48 µs/instrucción	1,78 µs/instrucción

Nota

Muchas variables afectan a los tiempos medidos. Los tiempos de ejecución son válidos para las instrucciones más rápidas de esta categoría y para programas que no contienen errores.

A.4.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1211C

Tabla A-12 Bloques, temporizadores y contadores soportados por la CPU 1211C

Elemento		**Descripción**
Bloques	Tipo	OB, FB, FC, DB
	Tamaño	Hasta el tamaño de la memoria de trabajo
	Cantidad	Un total de hasta 1024 bloques (OB + FB + FC + DB)
	Rango de direcciones para FB, FC y DB	FB y FC: de 1 a 65535 (p. ej., del FB 1 al FB 65535) DB: de 1 a 59999
	Profundidad de anidamiento	16 desde OB de ciclo o de arranque 6 desde cualquier OB de evento de alarma[1]
	Observar	Se puede observar a la vez el estado de 2 bloques lógicos

Elemento		Descripción
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarma de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

[1] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Tabla A-13 Comunicación

Datos técnicos	Descripción
Número de puertos	1
Tipo	Ethernet
Dispositivo HMI	4
Programadora (PG)	1
Conexiones	• 8 conexiones para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV • 8 conexiones CPU-to-CPU (cliente o servidor) para datos GET/PUT • 6 conexiones para asignación dinámica de memoria a GET/PUT u Open User Communication • 64 conexiones para certificados de seguridad máx.

Datos técnicos	Descripción
Transferencia de datos	10/100 Mb/s
Aislamiento (señal externa a lógica)	Transformador aislado galvánicamente, 1500 V AC (ensayo de tipo)[1]
Tipo de cable	CAT5e apantallado
Interfaces	
Número de interfaces PROFINET	1
Número de interfaces PROFIBUS	0
Interfaz	
Hardware de la interfaz	
Número de puertos	1
Switch integrado	No
RJ-45 (Ethernet)	Sí; X1
Protocolos	
Controlador PROFINET IO	Sí
Dispositivo PROFINET IO	Sí
Comunicación SIMATIC	Sí
Comunicación IE abierta	Sí
Servidor web	Sí
Redundancia de medios	No
Controlador PROFINET IO	
Servicios	
Comunicación PG/OP	Sí
Routing S7	Sí
Modo isócrono	No
Comunicación IE abierta	Sí
IRT	No
MRP	No
PROFIenergy	Sí. La CPU S7-1200 solo soporta la unidad PROFIenergy (con funcionalidad de I-device).
Arranque priorizado	Sí (máx. 16 dispositivos PROFINET)
Número de dispositivos IO conectables como máximo	16
Número de dispositivos IO que pueden conectarse para RT, máx.	16
De los cuales en línea, máx.	16
Número de dispositivos IO que pueden activarse/desactivarse simultáneamente, máx.	8
Tiempos de actualización	El valor mínimo del tiempo de actualización también depende del componente de comunicación establecido para PROFINET IO, del número de dispositivos IO y de la cantidad de datos de usuario configurados.
Con RT	
Tiempo de ciclo de emisión de 1 ms	1 ms a 512 ms
Dispositivo PROFINET IO	
Servicios	
Comunicación PG/OP	Sí

Datos técnicos		Descripción
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT, soportado		No
MRP, soportado		No
PROFIenergy		Sí
Shared Device		Sí
Número de controladores IO con Shared Device, máx.		2
Comunicación SIMATIC		
Comunicación S7 como servidor		Sí
Comunicación S7 como cliente		Sí
Datos de usuario por tarea, máx.		Consulte la Ayuda en pantalla (comunicación S7, tamaño de datos de usuario)
Comunicación IE abierta		
TCP/IP:		Sí
	Longitud de datos, máx.	8 KB
	Varias conexiones pasivas por puerto soportadas	Sí
ISO-on-TCP (RFC1006):		Sí
	Longitud de datos, máx.	8 KB
UDP		Sí
	Longitud de datos, máx.	1472 bytes
DHCP		No
SNMP		Sí
DCP		Sí
LLDP		Sí

[1] El aislamiento del puerto Ethernet está diseñado para limitar el riesgo por tensiones peligrosas durante defectos de red de corta duración. No es conforme con los requisitos de seguridad usuales para el aislamiento a tensión de línea a frecuencia industrial.

Tabla A-14 Alimentación eléctrica

Datos técnicos		CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Rango de tensión		De 85 a 264 V AC	De 20,4 V DC a 28,8 V DC	
Frecuencia de línea		De 47 a 63 Hz	--	
Intensidad de entrada	CPU solo a carga máx.	60 mA a 120 V AC 30 mA a 240 V AC	300 mA a 24 V DC	300 mA a 24 V DC
	CPU con todos los accesorios de ampliación a carga máx.	180 mA a 120 V AC 90 mA a 240 V AC	900 mA a 24 V DC	
Corriente de irrupción (máx.)		20 A a 264 V AC	12 A a 28,8 V DC	
$I^2 t$		0,8 $A^2 s$	0,5 $A^2 s$	

Datos técnicos	CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Aislamiento (alimentación de entrada a lógica)	1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional	0,5 mA máx.	--	
Tiempo de mantenimiento (pérdida de alimentación)	20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario	3 A, 250 V, de acción lenta		

Tabla A-15 Alimentación de sensores

Datos técnicos	CPU 1211C AC/DC/Relay	CPU 1211C DC/DC/relé	CPU 1211C DC/DC/DC
Rango de tensión	De 20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	300 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual que la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		

A.4.3 Entradas y salidas digitales

Tabla A-16 Entradas digitales

Datos técnicos	CPU 1211C AC/DC/relé, CPU 1211C DC/DC/relé y CPU 1211C DC/DC/DC
Número de entradas	6
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)
Tensión nominal	24 V DC a 4 mA, nominal
Tensión continua admisible	30 V DC, máx.
Sobretensión transitoria	35 V DC durante 0,5 s
Señal 1 lógica (mín.)	15 V DC a 2,5 mA
Señal 0 lógica (máx.)	5 V DC a 1 mA
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	100/80 kHz (de Ia.0 a Ia.5)
Número de entradas ON simultáneamente	6 a 60 °C en horizontal o 50 °C en vertical
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado, 50 m apantallado para entradas HSC

Tabla A-17 Salidas digitales

Datos técnicos	**CPU 1211C AC/DC/relé y CPU 1211C DC/DC/relé**	**CPU 1211C DC/DC/DC**
Número de salidas	4	
Tipo	Relé mecánico	Estado sólido MOSFET (fuente)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--	0,1 V DC máx.
Intensidad (máx.)	2,0 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.
Corriente de fuga por salida	--	10 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	8 A durante máx. 100 ms
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y circuito lógico)	707 V DC (examen de tipo)
Grupos de aislamiento	1	
Tensión de bloqueo inductiva	--	L+ menos 48 V DC, disipación de 1 W
Frecuencia máxima de conmutación de relé	1 Hz	--
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.	1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Frecuencia de tren de impulsos	No recomendada [1]	100 kHz (de Qa.0 a Qa.3)[2], 2 Hz mín.
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	--
Vida útil de los contactos bajo carga nominal	100 000 ciclos apertura/cierre	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	
Control de una entrada digital	Sí	
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	4 a 60 °C en horizontal o 50 °C en vertical	
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	

[1] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

[2] En función del receptor de impulsos y del cable, una resistencia de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a perturbaciones.

A.4.4 Entradas analógicas

Tabla A-18 Entradas analógicas

Datos técnicos	Descripción
Número de entradas	2
Tipo	Tensión (asimétrica)
Rango total	De 0 a 10 V
Rango total (palabra de datos)	De 0 a 27648
Rango de sobreimpulso	De 10,001 a 11,759 V
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	De 11,760 a 11,852 V
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Tensión soportada máxima	35 V DC
Filtrado	Ninguno, débil, medio o fuerte Consulte la tabla de Respuesta a un escalón (ms) para las entradas analógicas de la CPU (Página 1263).
Supresión de ruido	10, 50 o 60 Hz
Impedancia	≥100 KΩ
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Longitud de cable (metros)	100 m, par trenzado apantallado

A.4.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU

Tabla A-19 Respuesta a un escalón (ms), 0 V a 10 V medido a 95%

Selección de filtrado (valor medio de muestreo)	Supresión de frecuencias (tiempo de integración)		
	60 Hz	50 Hz	10 Hz
Ninguno (1 ciclo): Sin media	50 ms	50 ms	100 ms
Débil (4 ciclos): 4 muestreos	60 ms	70 ms	200 ms
Medio (16 ciclos): 16 muestreos	200 ms	240 ms	1150 ms
Fuerte (32 ciclos): 32 muestreos	400 ms	480 ms	2300 ms
Tiempo de muestreo	**4,17 ms**	**5 ms**	**25 ms**

A.4.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU

Tabla A-20 Tiempo de muestreo para las entradas analógicas integradas en la CPU

Supresión de frecuencias (selección del tiempo de integración)	Tiempo de muestreo
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

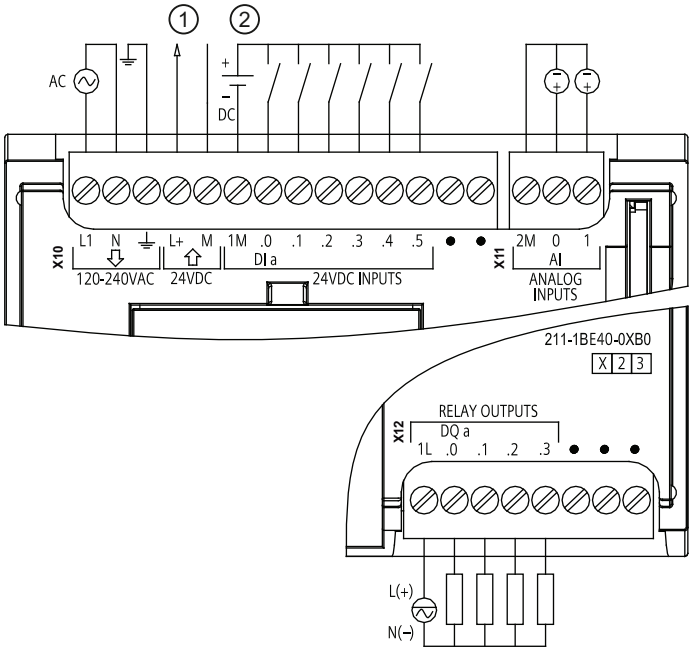
A.4.4.3 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A-21 Representación de entradas analógicas para tensión (CPU)

Sistema		Rango de medida de tensión	
Decimal	**Hexadecimal**	**De 0 a 10 V**	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

A.4.5 Diagramas de cableado de la CPU 1211C

Tabla A-22 CPU 1211C AC/DC/relé (6ES7211-1BE40-0XB0)



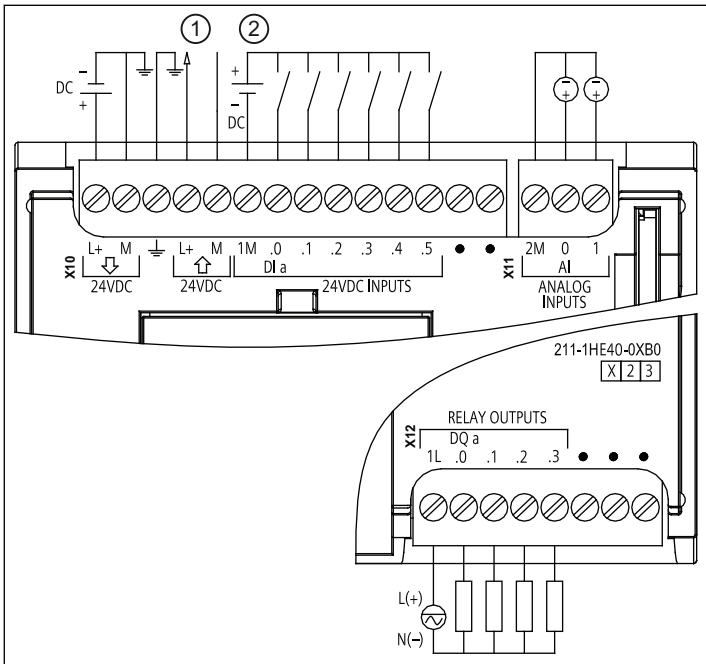
①	Alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación del sensor.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: Los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: El borne L1 o N (L2) se puede conectar a una fuente de tensión de hasta 240 V AC. El borne se puede considerar L2 y no es necesario que esté puesto a tierra. No se necesita polarización para los bornes L1 y N (L2).	
Nota 3: Consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-23 Asignación de pines de conectores para CPU 1211C AC/DC/relé (6ES7211-1BE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V DC	--	DQ a.3
6	1M	--	Sin conexión
7	DI a.0	--	Sin conexión
8	DI a.1	--	Sin conexión
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--

Pin	X10	X11 (oro)	X12
12	DI a.5	--	--
13	Sin conexión	--	--
14	Sin conexión	--	--

Tabla A-24 CPU 1211C DC/DC/relé (6ES7211-1HE40-0XB0)



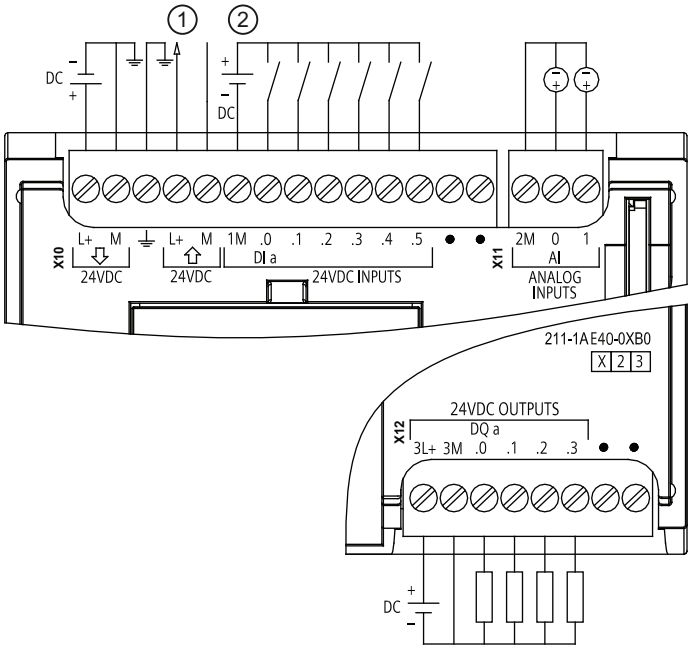
①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-25 Asignación de pines de conectores para CPU 1211C DC/DC/relé (6ES7211-1HE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V DC	--	DQ a.3
6	1M	--	Sin conexión
7	DI a.0	--	Sin conexión
8	DI a.1	--	Sin conexión
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--

Pin	X10	X11 (oro)	X12
13	Sin conexión	--	--
14	Sin conexión	--	--

Tabla A-26 CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)



①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-27 Asignación de pines de conectores para CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Tierra funcional	AI 1	DQ a.0
4	L+ / salida de sensor 24 V DC	--	DQ a.1
5	M / salida de sensor 24 V DC	--	DQ a.2
6	1M	--	DQ a.3
7	DI a.0	--	Sin conexión
8	DI a.1	--	Sin conexión
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Sin conexión	--	--
14	Sin conexión	--	--

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

A.5 CPU 1212C

A.5.1 Especificaciones generales y propiedades

Tabla A-28 General

Datos técnicos	CPU 1212C AC/DC/relé	CPU 1212C DC/DC/relé	CPU 1212C DC/DC/DC
Referencia	6ES7212-1BE40-0XB0	6ES7212-1HE40-0XB0	6ES7212-1AE40-0XB0
Dimensiones A x A x P (mm)	90 x 100 x 75		
Peso de envío	425 gramos	385 gramos	370 gramos
Disipación de potencia	11 W	9 W	
Intensidad disponible (SM y bus CM)	1000 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	300 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		

Tabla A-29 Propiedades de la CPU

Datos técnicos		Descripción
Memoria de usuario (Consulte "Datos técnicos generales (Página 1245)", "Remanencia de la memoria interna de la CPU".)	Trabajo	75 KB
	Carga	2 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	14 KB
E/S digitales integradas		8 entradas/6 salidas
E/S analógicas integradas:		2 entradas
Tamaño de la memoria imagen de proceso		1024 bytes de entradas (I)/1024 bytes de salidas (Q)
Área de marcas (M)		4096 bytes
Memoria temporal (local)		• 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados)
Ampliación con módulos de señales		2 SM máx.

Datos técnicos	Descripción
Ampliación con SB, CB o BB	1 máx.
Ampliación con módulos de comunicación	3 CM máx.
Contadores rápidos	Se han configurado un máximo de 6 para usar cualquier entrada integrada o de SB. Consulte "Asignación de los pines de entrada de hardware" (Página 563) para la CPU 1212C: Asignaciones de direcciones predeterminadas del HSC. • 100/[1]80 kHz (de Ia.0 a Ia.5) • 30 /[1]20 kHz (de Ia.6 a Ia.7)
Salidas pulsadas[2]	Se han configurado un máximo de 4 para usar cualquier salida integrada o de SB • 100 kHz (de Qa.0 a Qa.3) • 20 kHz (de Qa.4 a Qa.5)
Entradas de captura de impulsos	8
Alarmas de retardo	4 en total con resolución de 1 ms
Alarmas cíclicas	4 en total con resolución de 1 ms
Alarmas de flanco	8 ascendentes y 8 descendentes (12 y 12 con Signal Board opcional)
Memory Card	SIMATIC Memory Card (opcional)
Precisión del reloj en tiempo real	+/- 60 segundos/mes
Tiempo de respaldo del reloj de tiempo real	20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)

1 La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

2 Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

Tabla A-30 Rendimiento

Tipo de instrucción		Velocidad de ejecución	
		Direccionamiento directo (I, Q y M)	Accesos a DB
Boole		0,08 µs/instrucción	
Desplazamiento	Move_Bool	0,3 µs/instrucción	1,17 µs/instrucción
	Move_Word	0,137 µs/instrucción	1,0 µs/instrucción
	Move_Real	0,72 µs/instrucción	1,0 µs/instrucción
Funciones matemáticas	Sumar números reales	1,48 µs/instrucción	1,78 µs/instrucción

Nota

Muchas variables afectan a los tiempos medidos. Los tiempos de ejecución son válidos para las instrucciones más rápidas de esta categoría y para programas que no contienen errores.

A.5.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1212C

Tabla A-31 Bloques, temporizadores y contadores soportados por la CPU 1212C

Elemento		Descripción
Bloques	Tipo	OB, FB, FC, DB
	Tamaño	Hasta el tamaño de la memoria de trabajo
	Cantidad	Un total de hasta 1024 bloques (OB + FB + FC + DB)
	Rango de direcciones para FB, FC y DB	FB y FC: de 1 a 65535 (p. ej., del FB 1 al FB 65535) DB: de 1 a 59999
	Profundidad de anidamiento	16 desde OB de ciclo o de arranque 6 desde cualquier OB de evento de alarma[1]
	Observar	Se puede observar a la vez el estado de 2 bloques lógicos
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarma de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

[1] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Tabla A-32 Comunicación

Datos técnicos	Descripción
Número de puertos	1
Tipo	Ethernet
Dispositivo HMI	4
Programadora (PG)	1
Conexiones	• 8 conexiones para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV • 8 conexiones CPU-to-CPU (cliente o servidor) para datos GET/PUT • 6 conexiones para asignación dinámica de memoria a GET/PUT u Open User Communication • 64 conexiones para certificados de seguridad máx.
Transferencia de datos	10/100 Mb/s
Aislamiento (señal externa a lógica)	Transformador aislado galvánicamente, 1500 V AC (ensayo de tipo)[1]
Tipo de cable	CAT5e apantallado
Interfaces	
Número de interfaces PROFINET	1
Número de interfaces PROFIBUS	0
Interfaz	
Hardware de la interfaz	
Número de puertos	1
Switch integrado	No
RJ-45 (Ethernet)	Sí; X1
Protocolos	
Controlador PROFINET IO	Sí
Dispositivo PROFINET IO	Sí
Comunicación SIMATIC	Sí
Comunicación IE abierta	Sí
Servidor web	Sí
Redundancia de medios	No
Controlador PROFINET IO	
Servicios	
Comunicación PG/OP	Sí
Routing S7	Sí
Modo isócrono	No
Comunicación IE abierta	Sí
IRT	No
MRP	No
PROFIenergy	Sí. La CPU S7-1200 solo soporta la unidad PROFIenergy (con funcionalidad de I-device).
Arranque priorizado	Sí (máx. 16 dispositivos PROFINET)
Número de dispositivos IO conectables como máximo	16

Datos técnicos		Descripción
Número de dispositivos IO que pueden conectarse para RT, máx.		16
De los cuales en línea, máx.		16
Número de dispositivos IO que pueden activarse/ desactivarse simultáneamente, máx.		8
Tiempos de actualización		El valor mínimo del tiempo de actualización también depende del componente de comunicación establecido para PROFINET IO, del número de dispositivos IO y de la cantidad de datos de usuario configurados.
Con RT		
Tiempo de ciclo de emisión de 1 ms		1 ms a 512 ms
Dispositivo PROFINET IO		
Servicios		
Comunicación PG/OP		Sí
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT, soportado		No
MRP, soportado		No
PROFIenergy		Sí
Shared Device		Sí
Número de controladores IO con Shared Device, máx.		2
Comunicación SIMATIC		
Comunicación S7 como servidor		Sí
Comunicación S7 como cliente		Sí
Datos de usuario por tarea, máx.		Consulte la Ayuda en pantalla (comunicación S7, tamaño de datos de usuario)
Comunicación IE abierta		
TCP/IP:		Sí
	Longitud de datos, máx.	8 KB
	Varias conexiones pasivas por puerto soportadas	Sí
ISO-on-TCP (RFC1006):		Sí
	Longitud de datos, máx.	8 KB
UDP		Sí
	Longitud de datos, máx.	1472 bytes
DHCP		No
SNMP		Sí
DCP		Sí
LLDP		Sí

[1] El aislamiento del puerto Ethernet está diseñado para limitar el riesgo por tensiones peligrosas durante defectos de red de corta duración. No es conforme con los requisitos de seguridad usuales para el aislamiento a tensión de línea a frecuencia industrial.

Tabla A-33 Alimentación eléctrica

Datos técnicos		**CPU 1212C AC/DC/relé**	**CPU 1212C DC/DC/relé**	**CPU 1212C DC/DC/DC**
Rango de tensión		De 85 a 264 V AC	De 20,4 V DC a 28,8 V DC	
Frecuencia de línea		De 47 a 63 Hz	--	
Intensidad de entrada (carga máx.)	solo CPU	80 mA a 120 V AC 40 mA a 240 V AC	400 mA a 24 V DC	
	CPU con todos los accesorios de ampliación	240 mA a 120 V AC 120 mA a 240 V AC	1200 mA a 24 V DC	
Corriente de irrupción (máx.)		20 A a 264 V AC	12 A a 28,8 V DC	
I^2t		0,8 A^2 s	0,5 A^2 s	
Aislamiento (alimentación de entrada a lógica)		1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional		0,5 mA máx.	--	
Tiempo de mantenimiento (pérdida de alimentación)		20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario		3 A, 250 V, de acción lenta		

Tabla A-34 Alimentación de sensores

Datos técnicos	**CPU 1212C AC/DC/relé**	**CPU 1212C DC/DC/relé**	**CPU 1212C DC/DC/DC**
Rango de tensión	De 20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	300 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual que la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		

A.5.3 Entradas y salidas digitales

Tabla A-35 Entradas digitales

Datos técnicos	**CPU 1212C AC/DC/relé, DC/DC/relé y DC/DC/DC**
Número de entradas	8
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)
Tensión nominal	24 V DC a 4 mA, nominal
Tensión continua admisible	30 V DC, máx.
Sobretensión transitoria	35 V DC durante 0,5 s
Señal 1 lógica (mín.)	15 V DC a 2,5 mA
Señal 0 lógica (máx.)	5 V DC a 1 mA
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)

Datos técnicos	**CPU 1212C AC/DC/relé, DC/DC/relé y DC/DC/DC**
Grupos de aislamiento	1
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	100/80 kHz (de Ia.0 a Ia.5) 30/20 kHz (de Ia.6 a Ia.7)
Número de entradas ON simultáneamente	4 (sin puntos adyacentes) a 60 °C horizontal o 50 °C vertical De 8 a 55 °C horizontal o 45 °C vertical
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado, 50 m apantallado para entradas HSC

Tabla A-36 Salidas digitales

Datos técnicos	**CPU 1212C AC/DC/relé y DC/DC/relé**	**CPU 1212C DC/DC/DC**
Número de salidas	6	
Tipo	Relé mecánico	Estado sólido MOSFET (fuente)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--	0,1 V DC máx.
Intensidad (máx.)	2,0 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.
Corriente de fuga por salida	--	10 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	8 A durante máx. 100 ms
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y circuito lógico)	707 V DC (examen de tipo)
Grupos de aislamiento	2	1
Aislamiento (entre grupos)	1500 V AC[1]	--
Tensión de bloqueo inductiva	--	L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.	1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Retardo de conmutación (Qa.4 a Qa.5)	10 ms máx.	5 µs máx. OFF a ON 20 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	--
Frecuencia de tren de impulsos	No recomendada[2]	100 kHz (de Qa.0 a Qa.3)[3], 2 Hz mín. 20 kHz (de Qa.4 a Qa.5)[3]
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	--
Vida útil de los contactos bajo carga nominal	100 000 ciclos apertura/cierre	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí	

Datos técnicos	CPU 1212C AC/DC/relé y DC/DC/relé	CPU 1212C DC/DC/DC
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	3 (sin puntos adyacentes) a 60 °C horizontal o 50 °C vertical De 6 a 55 °C horizontal o 45 °C vertical	
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	

[1] El aislamiento de relé entre grupos separa la tensión de línea de SELV/PELV y separa diferentes fases de hasta 250 V AC de línea a tierra.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

[3] En función del receptor de impulsos y del cable, una resistencia de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a perturbaciones.

A.5.4 Entradas analógicas

Tabla A-37 Entradas analógicas

Datos técnicos	Descripción
Número de entradas	2
Tipo	Tensión (asimétrica)
Rango total	De 0 a 10 V
Rango total (palabra de datos)	De 0 a 27648
Rango de sobreimpulso	De 10,001 a 11,759 V
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	De 11,760 a 11,852 V
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Tensión soportada máxima	35 V DC
Filtrado	Ninguno, débil, medio o fuerte Consulte la tabla de Respuesta a un escalón (ms) para las entradas analógicas de la CPU (Página 1276).
Supresión de ruido	10, 50 o 60 Hz
Impedancia	≥100 KΩ
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Longitud de cable (metros)	100 m, par trenzado apantallado

A.5.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU

Tabla A-38 Respuesta a un escalón (ms), 0 V a 10 V medido a 95%

Selección de filtrado (valor medio de muestreo)	**Supresión de frecuencias (tiempo de integración)**		
	60 Hz	**50 Hz**	**10 Hz**
Ninguno (1 ciclo): Sin media	50 ms	50 ms	100 ms
Débil (4 ciclos): 4 muestreos	60 ms	70 ms	200 ms
Medio (16 ciclos): 16 muestreos	200 ms	240 ms	1150 ms
Fuerte (32 ciclos): 32 muestreos	400 ms	480 ms	2300 ms
Tiempo de muestreo	**4,17 ms**	**5 ms**	**25 ms**

A.5.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU

Tabla A-39 Tiempo de muestreo para las entradas analógicas integradas en la CPU

Supresión de frecuencias (selección del tiempo de integración)	**Tiempo de muestreo**
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

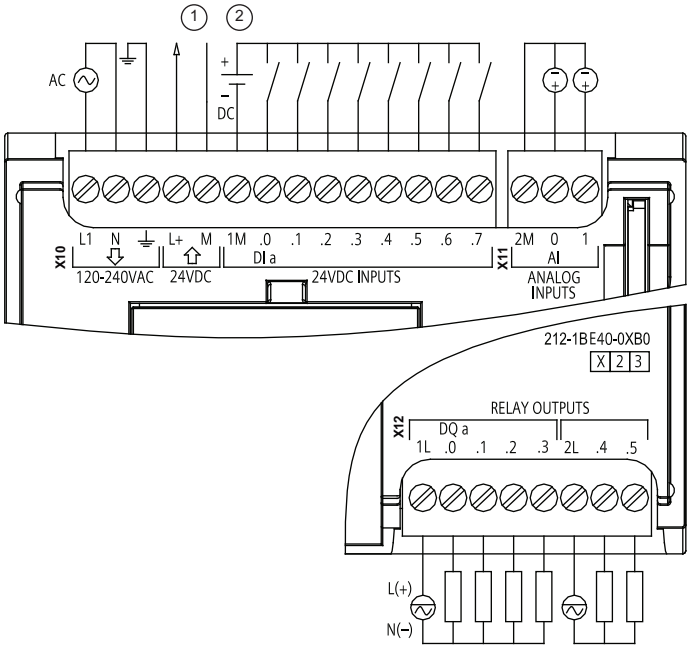
A.5.4.3 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A-40 Representación de entradas analógicas para tensión (CPU)

Sistema		**Rango de medida de tensión**	
Decimal	**Hexadecimal**	**De 0 a 10 V**	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

A.5.5 Diagramas de cableado de la CPU 1212C

Tabla A-41 CPU 1212C AC/DC/relé (6ES7212-1BE40-0XB0)



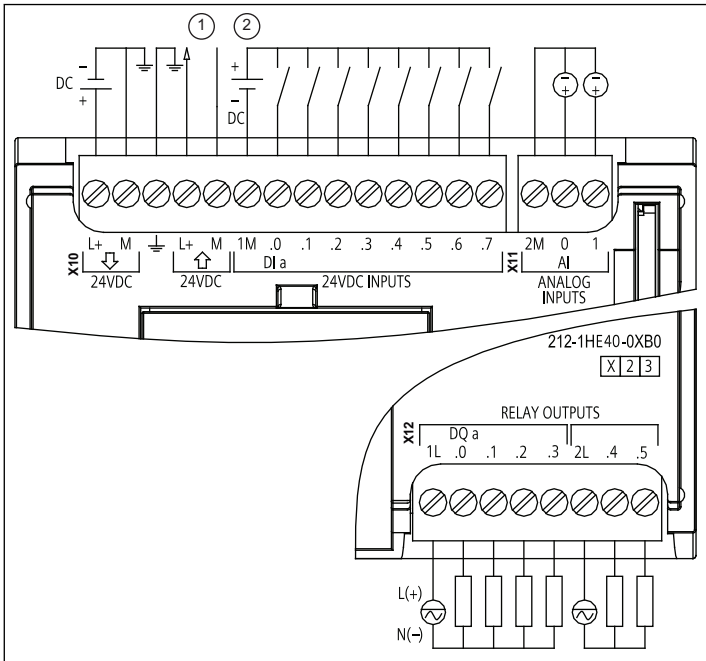
①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: Los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: El borne L1 o N (L2) se puede conectar a una fuente de tensión de hasta 240 V AC. El borne se puede considerar L2 y no es necesario que esté puesto a tierra. No se necesita polarización para los bornes L1 y N (L2).	
Nota 3: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-42 Asignación de pines de conectores para CPU 1212C AC/DC/relé (6ES7212-1BE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V 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	--	--

Pin	X10	X11 (oro)	X12
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tabla A-43 CPU 1212C DC/DC/relé (6ES7212-1HE40-0XB0)



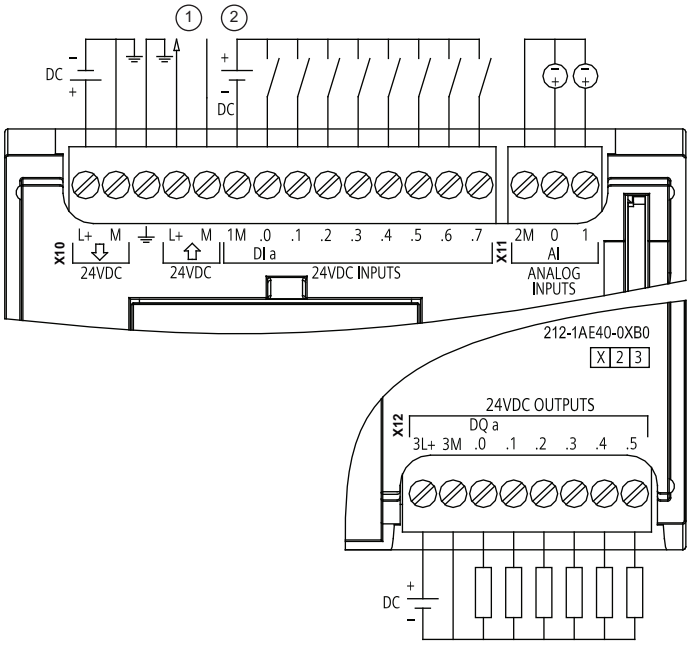
①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-44 Asignación de pines de conectores para CPU 1212C DC/DC/relé (6ES7212-1HE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V 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	--	--

Pin	X10	X11 (oro)	X12
13	DI a.6	--	--
14	DI a.7	--	--

Tabla A-45 CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)



①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-46 Asignación de pines de conectores para CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Tierra funcional	AI 1	DQ a.0
4	L+ / salida de sensor 24 V DC	--	DQ a.1
5	M / salida de sensor 24 V 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	--	--

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

A.6 CPU 1214C

A.6.1 Especificaciones generales y propiedades

Tabla A-47 General

Datos técnicos	CPU 1214C AC/DC/relé	CPU 1214C DC/DC/relé	CPU 1214C DC/DC/DC
Referencia	6ES7214-1BG40-0XB0	6ES7214-1HG40-0XB0	6ES7214-1AG40-0XB0
Dimensiones A x A x P (mm)	110 x 100 x 75		
Peso de envío	475 gramos	435 gramos	415 gramos
Disipación de potencia	14 W	12 W	
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada		

Tabla A-48 Propiedades de la CPU

Datos técnicos		Descripción
Memoria de usuario (Consulte "Datos técnicos generales" (Página 1245), "Remanencia de la memoria interna de la CPU".)	Trabajo	100 KB
	Carga	4 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	14 KB
E/S digitales integradas		14 entradas/10 salidas
E/S analógicas integradas:		2 entradas
Tamaño de la memoria imagen de proceso		1024 bytes de entradas (I)/1024 bytes de salidas (Q)
Área de marcas (M)		8192 bytes
Memoria temporal (local)		• 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados)
Ampliación con módulos de señales		8 SM máx.

Datos técnicos	Descripción
Ampliación con SB, CB o BB	1 máx.
Ampliación con módulos de comunicación	3 CM máx.
Contadores rápidos	Se han configurado un máximo de 6 para usar cualquier entrada integrada o de SB. Consulte "Asignación de los pines de entrada de hardware" (Página 563) para la CPU 1214C: Asignaciones de direcciones predeterminadas del HSC. • 100/[1]80 kHz (de Ia.0 a Ia.5) • 30/[1]20 kHz (de Ia.6 a Ib.5)
Salidas pulsadas[2]	Se han configurado un máximo de 4 para usar cualquier salida integrada o de SB • 100 kHz (de Qa.0 a Qa.3) • 20 kHz (de Qa.4 a Qb.1)
Entradas de captura de impulsos	14
Alarmas de retardo	4 en total con resolución de 1 ms
Alarmas cíclicas	4 en total con resolución de 1 ms
Alarmas de flanco	12 ascendentes y 12 descendentes (16 y 16 con Signal Board opcional)
Memory Card	SIMATIC Memory Card (opcional)
Precisión del reloj en tiempo real	+/- 60 segundos/mes
Tiempo de respaldo del reloj de tiempo real	20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)

1 La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

2 Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

Tabla A-49 Rendimiento

Tipo de instrucción		Velocidad de ejecución	
		Direccionamiento directo (I, Q y M)	Accesos a DB
Boole		0,08 µs/instrucción	
Desplazamiento	Move_Bool	0,3 µs/instrucción	1,17 µs/instrucción
	Move_Word	0,137 µs/instrucción	1,0 µs/instrucción
	Move_Real	0,72 µs/instrucción	1,0 µs/instrucción
Funciones matemáticas	Sumar números reales	1,48 µs/instrucción	1,78 µs/instrucción

Nota

Muchas variables afectan a los tiempos medidos. Los tiempos de ejecución son válidos para las instrucciones más rápidas de esta categoría y para programas que no contienen errores.

A.6.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1214C

Tabla A-50 Bloques, temporizadores y contadores soportados por la CPU 1214C

Elemento		Descripción
Bloques	Tipo	OB, FB, FC, DB
	Tamaño	OB, FB, FC: 64 KB DB: hasta el tamaño de la memoria de trabajo
	Cantidad	Un total de hasta 1024 bloques (OB + FB + FC + DB)
	Rango de direcciones para FB, FC y DB	FB y FC: de 1 a 65535 (p. ej., del FB 1 al FB 65535) DB: de 1 a 59999
	Profundidad de anidamiento	16 desde OB de ciclo o de arranque 6 desde cualquier OB de evento de alarma[1]
	Observar	Se puede observar a la vez el estado de 2 bloques lógicos
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarmas de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

[1] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Tabla A-51 Comunicación

Datos técnicos	Descripción
Número de puertos	1
Tipo	Ethernet
Dispositivo HMI	4
Programadora (PG)	1
Conexiones	• 8 conexiones para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV • 8 conexiones CPU-to-CPU (cliente o servidor) para datos GET/PUT • 6 conexiones para asignación dinámica de memoria a GET/PUT u Open User Communication • 64 conexiones para certificados de seguridad máx.
Transferencia de datos	10/100 Mb/s
Aislamiento (señal externa a lógica)	Transformador aislado galvánicamente, 1500 V AC (ensayo de tipo)[1]
Tipo de cable	CAT5e apantallado
Interfaces	
Número de interfaces PROFINET	1
Número de interfaces PROFIBUS	0
Interfaz	
Hardware de la interfaz	
Número de puertos	1
Switch integrado	No
RJ-45 (Ethernet)	Sí; X1
Protocolos	
Controlador PROFINET IO	Sí
Dispositivo PROFINET IO	Sí
Comunicación SIMATIC	Sí
Comunicación IE abierta	Sí
Servidor web	Sí
Redundancia de medios	No
Controlador PROFINET IO	
Servicios	
Comunicación PG/OP	Sí
Routing S7	Sí
Modo isócrono	No
Comunicación IE abierta	Sí
IRT	No
MRP	No
PROFIenergy	Sí. La CPU S7-1200 solo soporta la unidad PROFIenergy (con funcionalidad de I-device).
Arranque priorizado	Sí (máx. 16 dispositivos PROFINET)
Número de dispositivos IO conectables como máximo	16

Datos técnicos		Descripción
Número de dispositivos IO que pueden conectarse para RT, máx.		16
De los cuales en línea, máx.		16
Número de dispositivos IO que pueden activarse/ desactivarse simultáneamente, máx.		8
Tiempos de actualización		El valor mínimo del tiempo de actualización también depende del componente de comunicación establecido para PROFINET IO, del número de dispositivos IO y de la cantidad de datos de usuario configurados.
Con RT		
Tiempo de ciclo de emisión de 1 ms		1 ms a 512 ms
Dispositivo PROFINET IO		
Servicios		
Comunicación PG/OP		Sí
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT, soportado		No
MRP, soportado		No
PROFIenergy		Sí
Shared Device		Sí
Número de controladores IO con Shared Device, máx.		2
Comunicación SIMATIC		
Comunicación S7 como servidor		Sí
Comunicación S7 como cliente		Sí
Datos de usuario por tarea, máx.		Consulte la Ayuda en pantalla (comunicación S7, tamaño de datos de usuario)
Comunicación IE abierta		
TCP/IP:		Sí
	Longitud de datos, máx.	8 KB
	Varias conexiones pasivas por puerto soportadas	Sí
ISO-on-TCP (RFC1006):		Sí
	Longitud de datos, máx.	8 KB
UDP		Sí
	Longitud de datos, máx.	1472 bytes
DHCP		No
SNMP		Sí
DCP		Sí
LLDP		Sí

[1] El aislamiento del puerto Ethernet está diseñado para limitar el riesgo por tensiones peligrosas durante defectos de red de corta duración. No es conforme con los requisitos de seguridad usuales para el aislamiento a tensión de línea a frecuencia industrial.

Tabla A-52 Alimentación eléctrica

Datos técnicos		**CPU 1214C AC/DC/relé**	**CPU 1214C DC/DC/relé**	**CPU 1214C DC/DC/DC**
Rango de tensión		De 85 a 264 V AC	De 20,4 V DC a 28,8 V DC	
Frecuencia de línea		De 47 a 63 Hz	--	
Intensidad de entrada (carga máx.)	solo CPU	100 mA a 120 V AC 50 mA a 240 V AC	500 mA a 24 V DC	
	CPU con todos los accesorios de ampliación	300 mA a 120 V AC 150 mA a 240 V AC	1500 mA a 24 V DC	
Corriente de irrupción (máx.)		20 A a 264 V AC	12 A a 28,8 V DC	
I^2t		0,8 A^2 s	0,5 A^2 s	
Aislamiento (alimentación de entrada a lógica)		1500 V AC	Sin aislamiento	
Corriente de fuga a tierra, línea AC a tierra funcional		0,5 mA máx.	-	
Tiempo de mantenimiento (pérdida de alimentación)		20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario		3 A, 250 V, de acción lenta		

Tabla A-53 Alimentación de sensores

Datos técnicos	**CPU 1214C AC/DC/relé**	**CPU 1214C DC/DC/relé**	**CPU 1214C DC/DC/DC**
Rango de tensión	De 20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	400 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual que la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		

A.6.3 Entradas y salidas digitales

Tabla A-54 Entradas digitales

Datos técnicos	**CPU 1214C AC/DC/relé**	**CPU 1214C DC/DC/relé**	**CPU 1214C DC/DC/DC**
Número de entradas	14		
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)		
Tensión nominal	24 V DC a 4 mA, nominal		
Tensión continua admisible	30 V DC, máx.		
Sobretensión transitoria	35 V DC durante 0,5 s		
Señal 1 lógica (mín.)	15 V DC a 2,5 mA		
Señal 0 lógica (máx.)	5 V DC a 1 mA		
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)		

Datos técnicos	**CPU 1214C AC/DC/relé**	**CPU 1214C DC/DC/relé**	**CPU 1214C DC/DC/DC**
Grupos de aislamiento	1		
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0		
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	100/80 kHz (de Ia.0 a Ia.5) 30/20 kHz (de Ia.6 a Ib.5)		
Número de entradas ON simultáneamente	• 7 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 14 a 55 °C en horizontal o 45 °C en vertical		
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado, 50 m apantallado para entradas HSC		

Tabla A-55 Salidas digitales

Datos técnicos	**CPU 1214C AC/DC/relé y DC/DC/relé**	**CPU 1214C DC/DC/DC**
Número de salidas	10	
Tipo	Relé mecánico	Estado sólido MOSFET (fuente)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--	0,1 V DC máx.
Intensidad (máx.)	2,0 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.
Corriente de fuga por salida	--	10 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	8 A durante máx. 100 ms
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y circuito lógico)	707 V DC (examen de tipo)
Grupos de aislamiento	2	1
Aislamiento (entre grupos)	1500 V AC[1]	--
Tensión de bloqueo inductiva	--	L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.	1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Retardo de conmutación (Qa.4 a Qb.1)	10 ms máx.	5 µs máx. OFF a ON 20 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	--
Frecuencia de tren de impulsos	No recomendada [2]	100 kHz (de Qa.0 a Qa.3)[3], 2 Hz mín. 20 kHz (de Qa.4 a Qb.1)[3]
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	--
Vida útil de los contactos bajo carga nominal	100 000 ciclos apertura/cierre	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	

Datos técnicos	CPU 1214C AC/DC/relé y DC/DC/relé	CPU 1214C DC/DC/DC
Control de una entrada digital	Sí	
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	• 5 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 10 a 55 °C en horizontal o 45 °C en vertical	
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	

[1] El aislamiento de relé grupo a grupo separa la tensión de línea de SELV/PELV y separa diferentes fases de hasta 250 V AC de línea a tierra.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

[3] En función del receptor de impulsos y del cable, una resistencia de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a perturbaciones.

A.6.4 Entradas analógicas

Tabla A-56 Entradas analógicas

Datos técnicos	Descripción
Número de entradas	2
Tipo	Tensión (asimétrica)
Rango total	De 0 a 10 V
Rango total (palabra de datos)	De 0 a 27648
Rango de sobreimpulso	De 10,001 a 11,759 V
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	De 11,760 a 11,852 V
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Tensión soportada máxima	35 V DC
Filtrado	Ninguno, débil, medio o fuerte Consulte la tabla de Respuesta a un escalón (ms) para las entradas analógicas de la CPU (Página 1288).
Supresión de ruido	10, 50 o 60 Hz
Impedancia	≥100 KΩ
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Longitud de cable (metros)	100 m, par trenzado apantallado

A.6.4.1 Respuesta a un escalón de las entradas analógicas integradas en la CPU

Tabla A-57 Respuesta a un escalón (ms), 0 V a 10 V medido a 95%

Selección de filtrado (valor medio de muestreo)	**Supresión de frecuencias (tiempo de integración)**		
	60 Hz	**50 Hz**	**10 Hz**
Ninguno (1 ciclo): Sin media	50 ms	50 ms	100 ms
Débil (4 ciclos): 4 muestreos	60 ms	70 ms	200 ms
Medio (16 ciclos): 16 muestreos	200 ms	240 ms	1150 ms
Fuerte (32 ciclos): 32 muestreos	400 ms	480 ms	2300 ms
Tiempo de muestreo	**4,17 ms**	**5 ms**	**25 ms**

A.6.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU

Tabla A-58 Tiempo de muestreo para las entradas analógicas integradas en la CPU

Supresión de frecuencias (selección del tiempo de integración)	**Tiempo de muestreo**
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

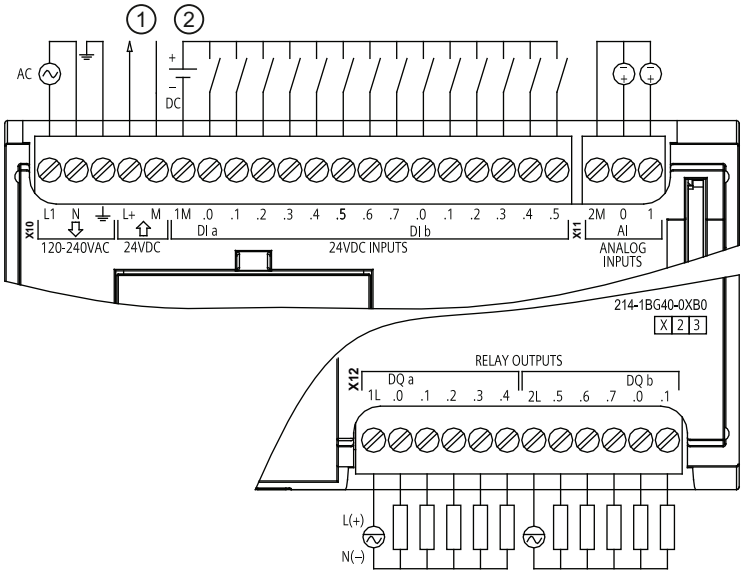
A.6.4.3 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A-59 Representación de entradas analógicas para tensión (CPU)

Sistema		**Rango de medida de tensión**	
Decimal	**Hexadecimal**	**De 0 a 10 V**	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

A.6.5 Diagramas de cableado de la CPU 1214C

Tabla A-60 CPU 1214C AC/DC/relé (6ES7214-1BG40-0XB0)



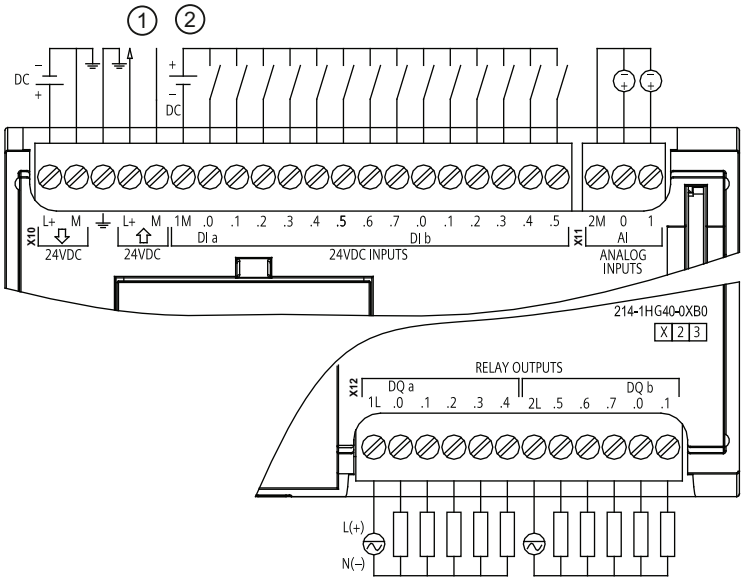
1	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
2	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: Los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: El borne L1 o N (L2) se puede conectar a una fuente de tensión de hasta 240 V AC. El borne se puede considerar L2 y no es necesario que esté puesto a tierra. No se necesita polarización para los bornes L1 y N (L2).	
Nota 3: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-61 Asignación de pines de conectores para CPU 1214C AC/DC/relé (6ES7214-1BG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V 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

Pin	X10	X11 (oro)	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	--	--

Tabla A-62 CPU 1214C DC/DC/relé (6ES7214-1HG40-0XB0)



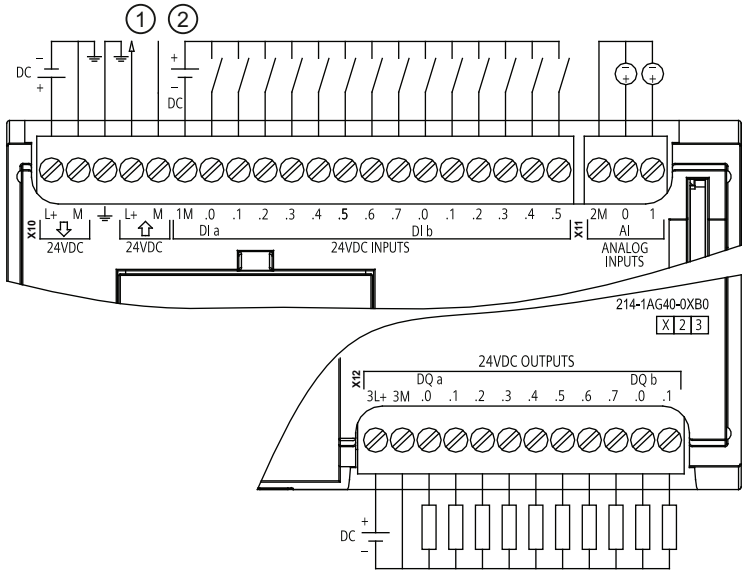
①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-63 Asignación de pines de conectores para CPU 1214C DC/DC/relé (6ES7214-1HG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DQ a.0
3	Tierra funcional	AI 1	DQ a.1
4	L+ / salida de sensor 24 V DC	--	DQ a.2
5	M / salida de sensor 24 V DC	--	DQ a.3
6	1M	--	DQ a.4
7	DI a.0	--	2L
8	DI a.1	--	DQ a.5

Pin	X10	X11 (oro)	X12
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	--	--

Tabla A-64 CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)



①	Salida de alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación de sensores.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-65 Asignación de pines de conectores para CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Tierra funcional	AI 1	DQ a.0
4	L+ / salida de sensor 24 V DC	--	DQ a.1
5	M / salida de sensor 24 V DC	--	DQ a.2

Pin	X10	X11 (oro)	X12
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
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	--	--

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

A.7 CPU 1215C

A.7.1 Especificaciones generales y propiedades

Tabla A-66 General

Datos técnicos	**CPU 1215C AC/DC/relé**	**CPU 1215C DC/DC/relé**	**CPU 1215C DC/DC/DC**
Referencia	6ES7215-1BG40-0XB0	6ES7215-1HG40-0XB0	6ES7215-1AG40-0XB0
Dimensiones A x A x P (mm)	130 x 100 x 75		
Peso de envío	585 gramos	550 gramos	520 gramos
Disipación de potencia	14 W	12 W	
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)		
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)		
Consumo de corriente de las entradas digitales (24 V DC)	4 mA / entrada utilizada		

Tabla A-67 Propiedades de la CPU

Datos técnicos		**Descripción**
Memoria de usuario (Consulte "Datos técnicos generales (Página 1245)", "Remanencia de la memoria interna de la CPU".)	Trabajo	125 KB
	Carga	4 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	14 KB
E/S digitales integradas		14 entradas/10 salidas
E/S analógicas integradas:		2 entradas/2 salidas
Tamaño de la memoria imagen de proceso		1024 bytes de entradas (I)/1024 bytes de salidas (Q)
Área de marcas (M)		8192 bytes
Memoria temporal (local)		• 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados)
Ampliación con módulos de señales		8 SM máx.
Ampliación con SB, CB o BB		1 máx.
Ampliación con módulos de comunicación		3 CM máx.
Contadores rápidos		Se han configurado un máximo de 6 para usar cualquier entrada integrada o de SB. Consulte "Asignación de los pines de entrada de hardware" (Página 563) para la CPU 1215C: Asignaciones de direcciones predeterminadas del HSC. • 100/[1]80 kHz (de Ia.0 a Ia.5) • 30/[1]20 kHz (de Ia.6 a Ib.5)
Salidas pulsadas[2]		Se han configurado un máximo de 4 para usar cualquier salida integrada o de SB • 100 kHz (de Qa.0 a Qa.3) • 20 kHz (de Qa.4 a Qb.1)
Entradas de captura de impulsos		14
Alarmas de retardo		4 en total con resolución de 1 ms
Alarmas cíclicas		4 en total con resolución de 1 ms
Alarmas de flanco		12 ascendentes y 12 descendentes (16 y 16 con Signal Board opcional)
Memory Card		SIMATIC Memory Card (opcional)
Precisión del reloj en tiempo real		+/- 60 segundos/mes
Tiempo de respaldo del reloj de tiempo real		20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)

[1] La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

Tabla A-68 Rendimiento

Tipo de instrucción		**Velocidad de ejecución**	
		Direccionamiento directo (I, Q y M)	**Accesos a DB**
Boole		0,08 µs/instrucción	
Desplaza-miento	Move_Bool	0,3 µs/instrucción	1,17 µs/instrucción
	Move_Word	0,137 µs/instrucción	1,0 µs/instrucción
	Move_Real	0,72 µs/instrucción	1,0 µs/instrucción
Funciones matemáticas	Sumar números reales	1,48 µs/instrucción	1,78 µs/instrucción

Nota

Muchas variables afectan a los tiempos medidos. Los tiempos de ejecución son válidos para las instrucciones más rápidas de esta categoría y para programas que no contienen errores.

A.7.2 Temporizadores, contadores y bloques lógicos soportados por la CPU 1215C

Tabla A-69 Bloques, temporizadores y contadores soportados por la CPU 1215C

Elemento		**Descripción**
Bloques	Tipo	OB, FB, FC, DB
	Tamaño	OB, FB, FC: 64 KB DB: hasta el tamaño de la memoria de trabajo
	Cantidad	Un total de hasta 1024 bloques (OB + FB + FC + DB)
	Rango de direcciones para FB, FC y DB	FB y FC: de 1 a 65535 (p. ej., del FB 1 al FB 65535) DB: de 1 a 59999
	Profundidad de anidamiento	16 desde OB de ciclo o de arranque 6 desde cualquier OB de evento de alarma[1]
	Observar	Se puede observar a la vez el estado de 2 bloques lógicos

Elemento		Descripción
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarmas de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

[1] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Tabla A-70 Comunicación

Datos técnicos	Descripción
Número de puertos	2
Tipo	Ethernet
Dispositivo HMI	4
Programadora (PG)	1
Conexiones	• 8 conexiones para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRCV • 8 conexiones CPU-to-CPU (cliente o servidor) para datos GET/PUT • 6 conexiones para asignación dinámica de memoria a GET/PUT u Open User Communication • 64 conexiones para certificados de seguridad máx.

Datos técnicos	Descripción
Transferencia de datos	10/100 Mb/s
Aislamiento (señal externa a lógica)	Transformador aislado galvánicamente, 1500 V AC (ensayo de tipo)[1]
Tipo de cable	CAT5e apantallado
Interfaces	
Número de interfaces PROFINET	1
Número de interfaces PROFIBUS	0
Interfaz	
Hardware de la interfaz	
Número de puertos	2
Switch integrado	Sí
RJ-45 (Ethernet)	Sí; X1
Protocolos	
Controlador PROFINET IO	Sí
Dispositivo PROFINET IO	Sí
Comunicación SIMATIC	Sí
Comunicación IE abierta	Sí
Servidor web	Sí
Redundancia de medios	Sí
Controlador PROFINET IO	
Servicios	
Comunicación PG/OP	Sí
Routing S7	Sí
Modo isócrono	No
Comunicación IE abierta	Sí
IRT	No
MRP	Sí, como cliente MRP
PROFIenergy	Sí. La CPU S7-1200 solo soporta la unidad PROFIenergy (con funcionalidad de I-device).
Arranque priorizado	Sí (máx. 16 dispositivos PROFINET)
Número de dispositivos IO conectables como máximo	16
Número de dispositivos IO que pueden conectarse para RT, máx.	16
De los cuales en línea, máx.	16
Número de dispositivos IO que pueden activarse/desactivarse simultáneamente, máx.	8
Tiempos de actualización	El valor mínimo del tiempo de actualización también depende del componente de comunicación establecido para PROFINET IO, del número de dispositivos IO y de la cantidad de datos de usuario configurados.
Con RT	
Tiempo de ciclo de emisión de 1 ms	1 ms a 512 ms
Dispositivo PROFINET IO	
Servicios	
Comunicación PG/OP	Sí

Datos técnicos		**Descripción**
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT, soportado		No
MRP, soportado		Sí
PROFIenergy		Sí
Shared Device		Sí
Número de controladores IO con Shared Device, máx.		2
Comunicación SIMATIC		
Comunicación S7 como servidor		Sí
Comunicación S7 como cliente		Sí
Datos de usuario por tarea, máx.		Consulte la Ayuda en pantalla (comunicación S7, tamaño de datos de usuario)
Comunicación IE abierta		
TCP/IP:		Sí
	Longitud de datos, máx.	8 KB
	Varias conexiones pasivas por puerto soportadas	Sí
ISO-on-TCP (RFC1006):		Sí
	Longitud de datos, máx.	8 KB
UDP:		Sí
	Longitud de datos, máx.	1472 bytes
DHCP		No
SNMP		Sí
DCP		Sí
LLDP		Sí

[1] El aislamiento del puerto Ethernet está diseñado para limitar el riesgo por tensiones peligrosas durante defectos de red de corta duración. No es conforme con los requisitos de seguridad usuales para el aislamiento a tensión de línea a frecuencia industrial.

Tabla A-71 Alimentación eléctrica

Datos técnicos		**CPU 1215C AC/DC/relé**	**CPU 1215C DC/DC/relé**	**CPU 1215C DC/DC/DC**
Rango de tensión		De 85 a 264 V AC	De 20,4 V DC a 28,8 V DC	
Frecuencia de línea		De 47 a 63 Hz	--	
Intensidad de entrada (carga máx.)	solo CPU	100 mA a 120 V AC 50 mA a 240 V AC	500 mA a 24 V DC	
	CPU con todos los accesorios de ampliación	300 mA a 120 V AC 150 mA a 240 V AC	1500 mA a 24 V DC	
Corriente de irrupción (máx.)		20 A a 264 V AC	12 A a 28,8 V DC	
I^2t		0,8 A^2 s	0,5 A^2 s	
Aislamiento (alimentación de entrada a lógica)		1500 V AC	Sin aislamiento	

Datos técnicos	CPU 1215C AC/DC/relé	CPU 1215C DC/DC/relé	CPU 1215C DC/DC/DC
Corriente de fuga a tierra, línea AC a tierra funcional	0,5 mA máx.	-	
Tiempo de mantenimiento (pérdida de alimentación)	20 ms a 120 V AC 80 ms a 240 V AC	10 ms a 24 V DC	
Fusible interno, no reemplazable por el usuario	3 A, 250 V, de acción lenta		

Tabla A-72 Alimentación de sensores

Datos técnicos	CPU 1215C AC/DC/relé	CPU 1215C DC/DC/relé	CPU 1215C DC/DC/DC
Rango de tensión	De 20,4 a 28,8 V DC	L+ menos 4 V DC mín.	
Intensidad de salida nominal (máx.)	400 mA (protegido contra cortocircuito)		
Ruido de rizado máx. (<10 MHz)	< 1 V de pico a pico	Igual que la línea de entrada	
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento		

A.7.3 Entradas y salidas digitales

Tabla A-73 Entradas digitales

Datos técnicos	CPU 1215C AC/DC/relé	CPU 1215C DC/DC/relé	CPU 1215C DC/DC/DC
Número de entradas	14		
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)		
Tensión nominal	24 V DC a 4 mA, nominal		
Tensión continua admisible	30 V DC, máx.		
Sobretensión transitoria	35 V DC durante 0,5 s		
Señal 1 lógica (mín.)	15 V DC a 2,5 mA		
Señal 0 lógica (máx.)	5 V DC a 1 mA		
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)		
Grupos de aislamiento	1		
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0		
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	100/80 kHz (de Ia.0 a Ia.5) 30/20 kHz (de Ia.6 a Ib.5)		
Número de entradas ON simultáneamente	• 7 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 14 a 55 °C en horizontal o 45 °C en vertical		
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado, 50 m apantallado para entradas HSC		

Tabla A-74 Salidas digitales

Datos técnicos	**CPU 1215C AC/DC/relé y CPU 1215C DC/DC/relé**	**CPU 1215C DC/DC/DC**
Número de salidas	10	
Tipo	Relé mecánico	Estado sólido MOSFET (fuente)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	--	0,1 V DC máx.
Intensidad (máx.)	2,0 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	5 W
Resistencia en estado ON	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.
Corriente de fuga por salida	--	10 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	8 A durante máx. 100 ms
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y circuito lógico)	707 V DC (examen de tipo)
Grupos de aislamiento	2	1
Aislamiento (entre grupos)	1500 V AC[1]	--
Tensión de bloqueo inductiva	--	L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.0 a Qa.3)	10 ms máx.	1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Retardo de conmutación (Qa.4 a Qb.1)	10 ms máx.	5 µs máx. OFF a ON 20 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	--
Frecuencia de tren de impulsos	No recomendada [2]	100 kHz (de Qa.0 a Qa.3)[3], 2 Hz mín. 20 kHz (de Qa.4 a Qb.1)[3]
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	--
Vida útil de los contactos bajo carga nominal	100 000 ciclos apertura/cierre	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	
Control de una entrada digital	Sí	
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	• 5 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 10 a 55 °C en horizontal o 45 °C en vertical	
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	

[1] El aislamiento de relé entre grupos separa la tensión de línea de SELV/PELV y separa diferentes fases de hasta 250 V AC de línea a tierra.

[2] Para modelos de CPU con salidas de relé, se debe instalar una Signal Board (SB) digital para emplear las salidas de impulsos.

[3] En función del receptor de impulsos y del cable, una resistencia de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a perturbaciones.

A.7.4 Entradas y salidas analógicas

Tabla A-75 Entradas analógicas

Datos técnicos	Descripción
Número de entradas	2
Tipo	Tensión (asimétrica)
Rango total	De 0 a 10 V
Rango total (palabra de datos)	De 0 a 27648
Rango de sobreimpulso	De 10,001 a 11,759 V
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	De 11,760 a 11,852 V
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Tensión soportada máxima	35 V DC
Filtrado	Ninguno, débil, medio o fuerte Consulte la tabla de Respuesta a un escalón (ms) para las entradas analógicas de la CPU (Página 1300).
Supresión de ruido	10, 50 o 60 Hz
Impedancia	≥100 KΩ
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Longitud de cable (metros)	100 m, par trenzado apantallado

A.7.4.1 Respuesta a escalón de las entradas analógicas integradas en la CPU

Tabla A-76 Respuesta a un escalón (ms), 0 V a 10 V medido a 95%

Selección de filtrado (valor medio de muestreo)	Supresión de frecuencias (tiempo de integración)		
	60 Hz	50 Hz	10 Hz
Ninguno (1 ciclo): Sin media	50 ms	50 ms	100 ms
Débil (4 ciclos): 4 muestreos	60 ms	70 ms	200 ms
Medio (16 ciclos): 16 muestreos	200 ms	240 ms	1150 ms
Fuerte (32 ciclos): 32 muestreos	400 ms	480 ms	2300 ms
Tiempo de muestreo	**4,17 ms**	**5 ms**	**25 ms**

A.7.4.2 Tiempo de muestreo para los puertos analógicos integrados en la CPU

Tabla A-77 Tiempo de muestreo para las entradas analógicas integradas en la CPU

Supresión de frecuencias (selección del tiempo de integración)	Tiempo de muestreo
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.7.4.3 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A-78 Representación de entradas analógicas para tensión (CPU)

Sistema		Rango de medida de tensión	
Decimal	**Hexadecimal**	**De 0 a 10 V**	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

A.7.4.4 Especificaciones de salidas analógicas

Tabla A-79 Salidas analógicas

Datos técnicos	Descripción
Número de salidas	2
Tipo	Intensidad
Rango total	0 a 20 mA
Rango total (palabra de datos)	de 0 a 27.648
Rango de sobreimpulso	De 20,01 a 23,52 mA
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	Ver nota al pie [1]
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Impedancia de salida	≤500 Ω máx.
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo

Datos técnicos	Descripción
Tiempo de estabilización	2 ms
Longitud de cable (metros)	100 m, par trenzado apantallado

[1] En una situación de desbordamiento, las salidas analógicas se comportarán según los valores de las propiedades de configuración del dispositivo. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

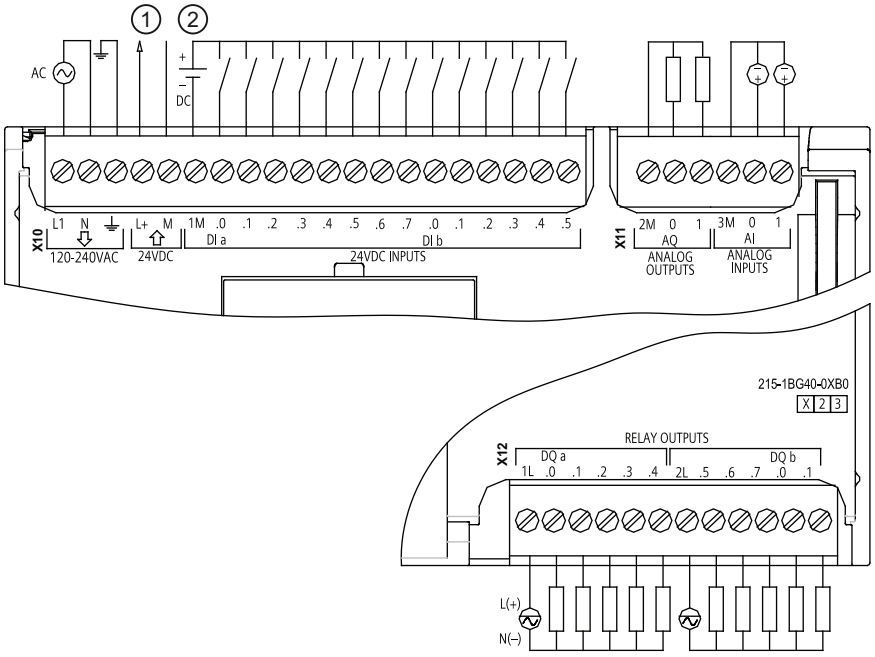
Tabla A-80 Representación de salidas analógicas para intensidad (CPU 1215C y CPU 1217C)

Sistema		Rango de salida de intensidad	
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	
32511	7EFF	23,52 mA	Rango de sobreimpulso
27649	6C01		
27648	6C00	20 mA	Rango nominal
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Valores negativos		Los valores negativos no se soportan	

[1] En una situación de rebase por exceso, las salidas analógicas se comportarán según los valores de las propiedades de configuración del dispositivo. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

A.7.5 Diagramas de cableado de la CPU 1215C

Tabla A-81 CPU 1215C AC/DC/relé (6ES7215-1BG40-0XB0)



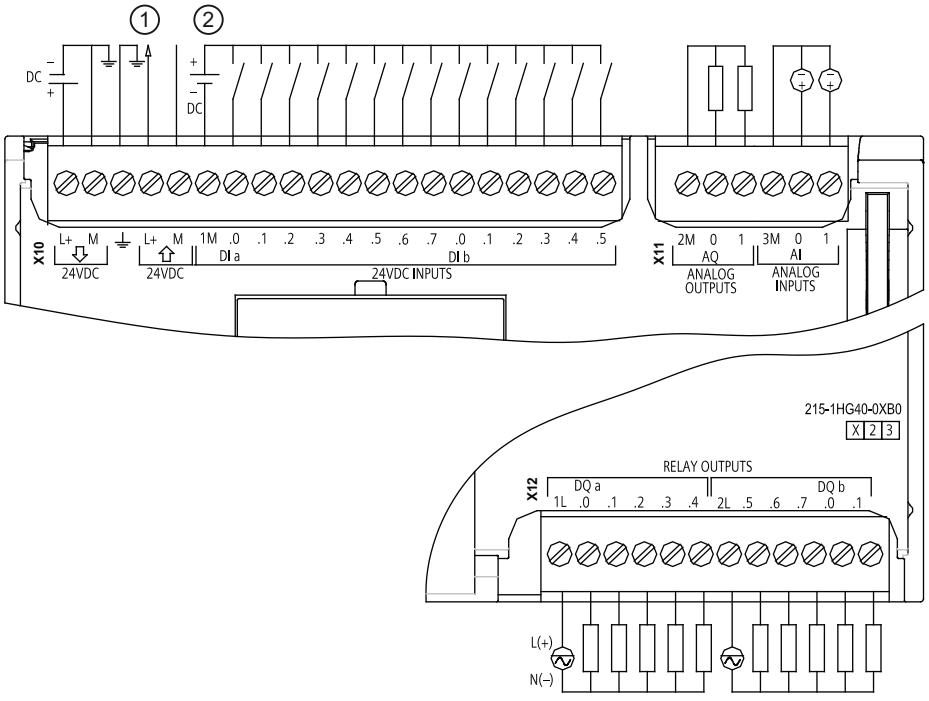
①	Alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación del sensor.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: Los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: El borne L1 o N (L2) se puede conectar a una fuente de tensión de hasta 240 V AC. El borne se puede considerar L2 y no es necesario que esté puesto a tierra. No se necesita polarización para los bornes L1 y N (L2).	
Nota 3: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-82 Asignación de pines de conectores para CPU 1215C AC/DC/relé (6ES7215-1BG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L1 /120-240 V AC	2 M	1L
2	N / 120 - 240 V AC	AQ 0	DQ a.0
3	Tierra funcional	AQ 1	DQ a.1
4	L+ / salida de sensor 24 V DC	3M	DQ a.2
5	M / salida de sensor 24 V 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

Pin	X10	X11 (oro)	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	--	--

Tabla A-83 CPU 1215C DC/DC/relé (6ES7215-1HG40-0XB0)



1	Alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación del sensor.
2	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

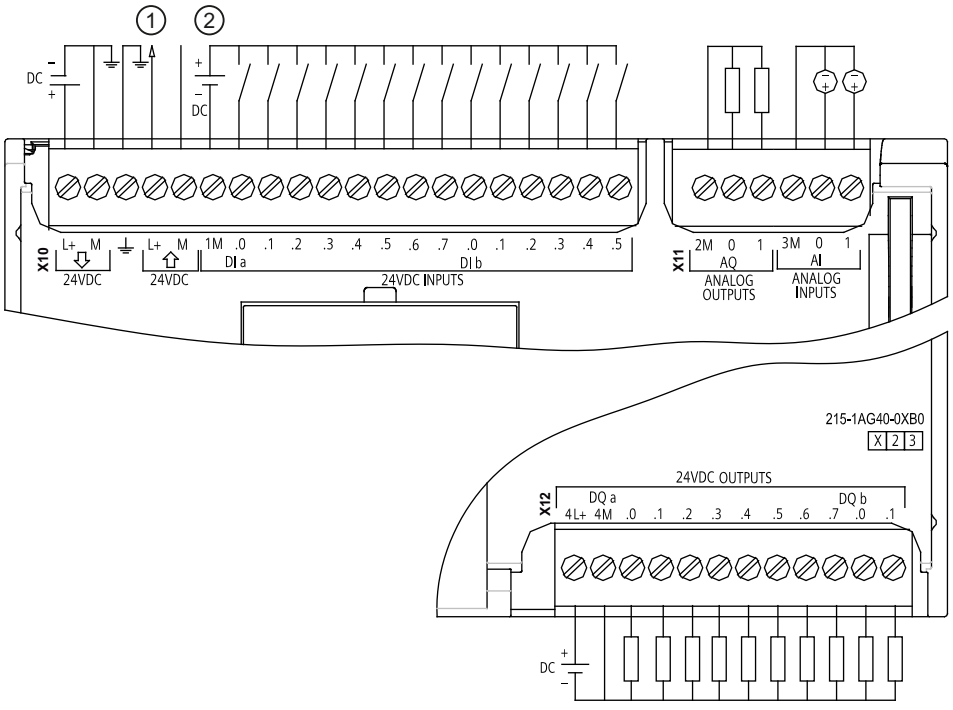
Nota 1: Los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.

Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.

Tabla A-84 Asignación de pines de conectores para CPU 1215C DC/DC/relé (6ES7215-1HG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AQ 0	DQ a.0
3	Tierra funcional	AQ 1	DQ a.1
4	L+ / salida de sensor 24 V DC	3M	DQ a.2
5	M / salida de sensor 24 V 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	--	--

Tabla A-85 CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)



1	Alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación del sensor.
2	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".
Nota 1: los conectores X11 deben ser de oro. Consulte el anexo C, Repuestos, para ver la referencia.	
Nota 2: consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

Tabla A-86 Asignación de pines de conectores para CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

Pin	X10	X11 (oro)	X12
1	L1 / 24 V DC	2 M	4L+
2	M / 24 V DC	AQ 0	4M
3	Tierra funcional	AQ 1	DQ a.0
4	L+ / salida de sensor 24 V DC	3M	DQ a.1
5	M / salida de sensor 24 V 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

Pin	X10	X11 (oro)	X12
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	--	--

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

A.8 CPU 1217C

A.8.1 Especificaciones generales y propiedades

Tabla A-87 General

Datos técnicos	CPU 1217C DC/DC/DC
Referencia	6ES7217-1AG40-0XB0
Dimensiones A x A x P (mm)	150 x 100 x 75
Peso de envío	530 gramos
Disipación de potencia	12 W
Intensidad disponible (SM y bus CM)	1600 mA máx. (5 V DC)
Intensidad disponible (24 V DC)	400 mA máx. (alimentación de sensores)
Consumo de corriente de las entradas digitales (24 V DC)	4 mA/entrada utilizada

Tabla A-88 Propiedades de la CPU

Datos técnicos		Descripción
Memoria de usuario (Consulte "Datos técnicos generales (Página 1245)", "Remanencia de la memoria interna de la CPU".)	Trabajo	150 KB
	Carga	4 MB, interna, ampliable hasta tamaño de tarjeta SD
	Remanente	14 KB
E/S digitales integradas		14 entradas/10 salidas

Datos técnicos	Descripción
E/S analógicas integradas:	2 entradas/2 salidas
Tamaño de la memoria imagen de proceso	1024 bytes de entradas (I)/1024 bytes de salidas (Q)
Área de marcas (M)	8192 bytes
Memoria temporal (local)	• 16 KB para arranque y ciclo (incluyendo los FB y FC asociados) • 6 KB para cada uno de los niveles de prioridad de alarma (incluidos los FB y FC asociados)
Ampliación con módulos de señales	8 SM máx.
Ampliación con SB, CB o BB	1 máx.
Ampliación con módulos de comunicación	3 CM máx.
Contadores rápidos	Hasta 6 configuradas para cualquier entrada incorporada o de SB (véase la tabla de configuración de hardware de entradas digitales (DI) de la CPU 1217C) (Página 1313) • 1 MHz (de Ib.2 a Ib.5) • 100/[1]80 kHz (de Ia.0 a Ia.5) • 30/[1]20 kHz (de Ia.6 a Ib.1)
Salidas de impulsos	Hasta 4 configuradas para cualquier salida incorporada o de SB (véase la tabla de configuración de hardware de salidas digitales (DO) de la CPU 1217C) (Página 1313) • 1 MHz (de Qa.0 a Qa.3) • 100 kHz (de Qa.4 a Qb.1)
Entradas de captura de impulsos	14
Alarmas de retardo	4 en total con resolución de 1 ms
Alarmas cíclicas	4 en total con resolución de 1 ms
Alarmas de flanco	12 ascendentes y 12 descendentes (16 y 16 con Signal Board opcional)
Memory Card	SIMATIC Memory Card (opcional)
Precisión del reloj en tiempo real	+/- 60 segundos/mes
Tiempo de respaldo del reloj de tiempo real	20 días típ./12 días mín. a 40 °C (condensador de alto rendimiento sin mantenimiento)

[1] La velocidad más lenta se aplica cuando el HSC está configurado para el modo de operación en cuadratura.

Tabla A-89 Rendimiento

Tipo de instrucción		Velocidad de ejecución	
		Direccionamiento directo (I, Q y M)	Accesos a DB
Boole		0,08 µs/instrucción	
Desplazamiento	Move_Bool	0,3 µs/instrucción	1,17 µs/instrucción
	Move_Word	0,137 µs/instrucción	1,0 µs/instrucción
	Move_Real	0,72 µs/instrucción	1,0 µs/instrucción
Funciones matemáticas	Sumar números reales	1,48 µs/instrucción	1,78 µs/instrucción

Nota

Muchas variables afectan a los tiempos medidos. Los tiempos de ejecución son válidos para las instrucciones más rápidas de esta categoría y para programas que no contienen errores.

A.8.2 Temporizadores, contadores y bloques lógicos admitidos por la CPU 1217C

Tabla A-90 Bloques, temporizadores y contadores admitidos por la CPU 1217C

Elemento		Descripción
Bloques	Tipo	OB, FB, FC, DB
	Tamaño	OB, FB, FC: 64 KB DB: hasta el tamaño de la memoria de trabajo
	Cantidad	Un total de hasta 1024 bloques (OB + FB + FC + DB)
	Rango de direcciones para FB, FC y DB	FB y FC: de 1 a 65535 (p. ej., del FB 1 al FB 65535) DB: de 1 a 59999
	Profundidad de anidamiento	16 desde OB de ciclo o de arranque 6 desde cualquier OB de evento de alarma[1]
	Observar	Se puede observar a la vez el estado de 2 bloques lógicos
OB	Ciclo del programa	Varios
	Arranque	Varios
	Alarmas de retardo	4 (1 por evento)
	Alarmas cíclicas	4 (1 por evento)
	Alarmas de proceso	50 (1 por evento)
	Alarmas de error de tiempo	1
	Alarmas de error de diagnóstico	1
	Enchufe o desenchufe de módulos	1
	Fallo de rack o estación	1
	Hora	Varios
	Estado	1
	Actualizar	1
	Perfil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
	MC-PostServo	1
Temporizadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, 16 bytes por temporizador

Elemento		Descripción
Contadores	Tipo	CEI
	Cantidad	Solo limitada por el tamaño de la memoria
	Almacenamiento	Estructura en DB, tamaño dependiente del tipo de contaje • SInt, USInt: 3 bytes • Int, UInt: 6 bytes • DInt, UDInt: 12 bytes

[1] Los programas de seguridad utilizan dos niveles de anidamiento. Por ello, el programa de usuario tiene una profundidad de anidamiento de cuatro en programas de seguridad.

Tabla A-91 Comunicación

Datos técnicos	Descripción
Número de puertos	2
Tipo	Ethernet
Dispositivo HMI	4
Programadora (PG)	1
Conexiones	• 8 conexiones para Open User Communication (activa o pasiva): TSEND_C, TRCV_C, TCON, TDISCON, TSEND y TRC • 8 conexiones CPU-to-CPU (cliente o servidor) para datos GET/PUT • 6 conexiones para asignación dinámica de memoria a GET/PUT u Open User Communication • 64 conexiones para certificados de seguridad máx.
Transferencia de datos	10/100 Mb/s
Aislamiento (señal externa a lógica)	Transformador aislado galvánicamente, 1500 V AC (ensayo de tipo)[1]
Tipo de cable	CAT5e apantallado
Interfaces	
Número de interfaces PROFINET	1
Número de interfaces PROFIBUS	0
Interfaz	
Hardware de la interfaz	
Número de puertos	2
Switch integrado	Sí
RJ-45 (Ethernet)	Sí; X1
Protocolos	
Controlador PROFINET IO	Sí
Dispositivo PROFINET IO	Sí
Comunicación SIMATIC	Sí
Comunicación IE abierta	Sí
Servidor web	Sí
Redundancia de medios	Sí
Controlador PROFINET IO	
Servicios	
Comunicación PG/OP	Sí

Datos técnicos		Descripción
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT		No
MRP		Sí, como cliente MRP
PROFIenergy		Sí. La CPU S7-1200 solo soporta la unidad PROFIenergy (con funcionalidad de I-device).
Arranque priorizado		Sí (máx. 16 dispositivos PROFINET)
Número de dispositivos IO conectables como máximo		16
Número de dispositivos IO que pueden conectarse para RT, máx.		16
De los cuales en línea, máx.		16
Número de dispositivos IO que pueden activarse/desactivarse simultáneamente, máx.		8
Tiempos de actualización		El valor mínimo del tiempo de actualización también depende del componente de comunicación establecido para PROFINET IO, del número de dispositivos IO y de la cantidad de datos de usuario configurados.
Con RT		
Tiempo de ciclo de emisión de 1 ms		1 ms a 512 ms
Dispositivo PROFINET IO		
Servicios		
Comunicación PG/OP		Sí
Routing S7		Sí
Modo isócrono		No
Comunicación IE abierta		Sí
IRT, soportado		No
MRP, soportado		Sí
PROFIenergy		Sí
Shared Device		Sí
Número de controladores IO con Shared Device, máx.		2
Comunicación SIMATIC		
Comunicación S7 como servidor		Sí
Comunicación S7 como cliente		Sí
Datos de usuario por tarea, máx.		Consulte la Ayuda en pantalla (comunicación S7, tamaño de datos de usuario)
Comunicación IE abierta		
TCP/IP:		Sí
	Longitud de datos, máx.	8 KB
	Varias conexiones pasivas por puerto soportadas	Sí
ISO-on-TCP (RFC1006):		Sí
	Longitud de datos, máx.	8 KB
UDP:		Sí

Datos técnicos		Descripción
	Longitud de datos, máx.	1472 bytes
DHCP		No
SNMP		Sí
DCP		Sí
LLDP		Sí

[1] El aislamiento del puerto Ethernet está diseñado para limitar el riesgo por tensiones peligrosas durante defectos de red de corta duración. No es conforme con los requisitos de seguridad usuales para el aislamiento a tensión de línea a frecuencia industrial.

Tabla A-92 Alimentación eléctrica

Datos técnicos		CPU 1217C DC/DC/DC
Rango de tensión		De 20,4 V DC a 28,8 V DC
Frecuencia de línea		--
Intensidad de entrada (carga máx.)	solo CPU	600 mA a 24 V DC
	CPU con todos los accesorios de ampliación	1600 mA a 24 V DC
Corriente de irrupción (máx.)		12 A a 28,8 V DC
$I^2 t$		0,5 $A^2 s$
Aislamiento (alimentación de entrada a lógica)		Sin aislamiento
Tiempo de mantenimiento (de pérdida de alimentación)		10 ms a 24 V DC
Fusible interno, no reemplazable por el usuario		3 A, 250 V, de acción lenta

Tabla A-93 Alimentación de sensores

Datos técnicos	CPU 1217C DC/DC/DC
Rango de tensión	L+ menos 4 V DC mín.
Intensidad de salida nominal (máx.)	400 mA (protegido contra cortocircuito)
Ruido de rizado máx. (<10 MHz)	Igual que la línea de entrada
Aislamiento (lógica de la CPU a alimentación de sensores)	Sin aislamiento

A.8.3 Entradas y salidas digitales

Tabla A-94 Entradas digitales

Datos técnicos	CPU 1217C DC/DC/DC
Número de entradas	14: Total: 10: Sumidero/fuente (tipo 1 IEC sumidero) 4: Diferencial (RS422/RS485)
Tipo: Sumidero/fuente (IEC tipo 1, sumidero)	De Ia.0 a Ia.7, de Ib.0 a Ib.1
Tensión nominal	24 V DC a 4 mA, nominal
Tensión continua admisible	30 V DC, máx.
Sobretensión transitoria	35 V DC durante 0,5 s
Señal 1 lógica (mín.)	15 V DC a 2,5 mA
Señal 0 lógica (máx.)	5 V DC a 1 mA
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Frecuencias de entrada de reloj HSC (máx.) (señal 1 lógica = 15 a 26 V DC)	100/80 kHz (de Ia.0 a Ia.5) 30/20 kHz (de Ia.6 a Ib.1)
Tipo: Entrada diferencial (RS422/RS485)	De Ib.2 to Ib.5 (de .2+ .2- a .5+ .5-)
Rango de tensión en modo común	De -7 V a +12 V, 1 segundo, 3 VRMS continuo (características de RS422/RS485)
Terminación y polarización integradas	De 390 Ω a 2M en Ib'-', de 390 Ω a +5 V en Ib'-', (sin polarización cuando el circuito de terminación/polarización está abierto) 220 Ω entre Ib'+' y Ib'-'
Impedancia de entrada del receptor	100 Ω incluidas la polarización y la terminación
Umbral/sensibilidad del receptor diferencial	+/- 0,2 V mín., 60 mV de histéresis típica (características de RS422/RS485)
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Tiempos de filtro	Configuración en us: 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0 Configuración en ms: 0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Frecuencias de entrada de reloj HSC (máx.)	Fase simple: 1 MHz (de Ib.2 a Ib.5) Fase en cuadratura: 1 MHz (de Ib.2 a Ib.5)
Sesgo entre canales de entrada diferencial	40 ns máx.
Especificaciones generales (todas las entradas digitales)	

Datos técnicos	CPU 1217C DC/DC/DC
Número de entradas ON simultáneamente	5 entradas de sumidero/fuente (no adyacentes) y 4 entradas diferenciales a 60 °C en horizontal o 50 °C en vertical 14 a 55 °C en horizontal o 45 °C en vertical
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado 50 m apantallado para entradas HSC (sumidero/fuente) 50 m par trenzado y apantallado para todas las entradas diferenciales

Tabla A-95 Tabla de configuración de hardware de entradas digitales de la CPU 1217C

Entrada	Tipo y frecuencia
DIa.0	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.1	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.2	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.3	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.4	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.5	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 100 kHz máx.
DIa.6	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 30 kHz máx.
DIa.7	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 30 kHz máx.
DIb.0	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 30 kHz máx.
DIb.1	Tipo: 24 V, entrada tipo 1 sumidero/fuente Frecuencia de entrada del contador rápido: 30 kHz máx.
DIb.2+ .2-	Tipo: Entrada diferencial de RS422/RS485 Frecuencia de entrada del contador rápido: 1 MHz máx.
DIb.3+ .3-	Tipo: Entrada diferencial de RS422/RS485 Frecuencia de entrada del contador rápido: 1 MHz máx.
DIb.4+ .4-	Tipo: Entrada diferencial de RS422/RS485 Frecuencia de entrada del contador rápido: 1 MHz máx.
DIb.5+ .5-	Tipo: Entrada diferencial de RS422/RS485 Frecuencia de entrada del contador rápido: 1 MHz máx.

Tabla A-96 Salidas digitales

Datos técnicos	**CPU 1217C DC/DC/DC**
Número de salidas	10 en total 6: Estado sólido MOSFET (fuente) 4: Diferencial (RS422/RS485)
Tipo: Estado sólido - MOSFET (salida en fuente)	De Qa.4 a Qb.1
Rango de tensión	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	20 V DC mín.
Señal 0 lógica con carga de 10 KΩ	0,1 V DC máx.
Intensidad (máx.)	0,5 A
Carga de lámparas	5 W
Resistencia en estado ON	0,6 Ω máx.
Corriente de fuga por salida	10 µA máx.
Extracorriente de conexión	8 A durante máx. 100 ms
Protección contra sobrecargas	No
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Tensión de bloqueo inductiva	L+ menos 48 V DC, disipación de 1 W
Retardo de conmutación (Qa.4 a Qb.1)	1,0 µs máx., OFF a ON 3,0 µs máx., ON a OFF
Frecuencia máxima de conmutación de relé	--
Frecuencia de tren de impulsos	100 kHz máx. (de Qa.4 a Qb.1)[1], 2 Hz mín.
Tipo: Salida diferencial (RS422/RS485)	De Qa.0 a Qa.3 (de .0+ 0- a .3+ .3-)
Rango de tensión en modo común	De -7 V a +12 V, 1 segundo, 3 VRMS continuo (características de RS422/RS485)
Tensión de salida diferencial del transmisor	2 V mín. a RL = 100 Ω, 1,5 V mín. a RL = 54 Ω (características de RS422/RS485)
Terminación integrada	100 Ω entre Qa'+' y Qa'-'
Impedancia de salida de driver	100 Ω incluida la terminación
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Retardo de conmutación (de DQa.0 a DQa.3)	100 ns máx.
Sesgo entre canales de salida diferencial	40 ns máx.
Frecuencia del tren de impulsos	1 MHz (de Qa.0 a Qa.3), 2 Hz mín.
Especificaciones generales (todas las salidas digitales)	
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí

Datos técnicos	CPU 1217C DC/DC/DC
Salidas paralelas para control de carga redundante	Sí (solo Qa.4 a Qb.1; con el mismo neutro)
Salidas paralelas para carga aumentada	No
Número de salidas ON simultáneamente	3 salidas MOSFET (fuente) de estado sólido (no adyacentes) y 4 salidas diferenciales a 60 °C en horizontal o 50 °C en vertical 10 a 55 °C en horizontal o 45 °C en vertical
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado

[1] En función del receptor de impulsos y del cable, un resistor de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a interferencias.

Tabla A-97 Tabla de configuración de hardware de salidas digitales (DQ) de la CPU 1217C

Salida	Tipo y frecuencia
DQa.0+ .0-	Tipo: Salida diferencial de RS422/RS485 Frecuencia de tren de impulsos: 1 MHz máx., 2 Hz mín.
DQa.1+ .1-	Tipo: Salida diferencial de RS422/RS485 Frecuencia de tren de impulsos: 1 MHz máx., 2 Hz mín.
DQa.2+ .2-	Tipo: Salida diferencial de RS422/RS485 Frecuencia de tren de impulsos: 1 MHz máx., 2 Hz mín.
DQa.3+ .3-	Tipo: Salida diferencial de RS422/RS485 Frecuencia de tren de impulsos: 1 MHz máx., 2 Hz mín.
DQa.4	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.
DQa.5	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.
DQa.6	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.
DQa.7	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.
DQb.0	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.
DQb.1	Tipo: Salida en fuente de 24 V Frecuencia de tren de impulsos: 100 kHz máx., 2 Hz mín.

A.8.4 Entradas y salidas analógicas

A.8.4.1 Datos técnicos de la entrada analógica

Tabla A-98 Entradas analógicas

Datos técnicos	Descripción
Número de entradas	2
Tipo	Tensión (asimétrica)

Datos técnicos	Descripción
Rango total	De 0 a 10 V
Rango total (palabra de datos)	De 0 a 27648
Rango de sobreimpulso	De 10,001 a 11,759 V
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	De 11,760 a 11,852 V
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Tensión soportada máxima	35 V DC
Filtrado	Ninguno, débil, medio o fuerte Consulte la tabla de Respuesta a un escalón (ms) para las entradas analógicas de la CPU (Página 1317).
Supresión de ruido	10, 50 o 60 Hz
Impedancia	≥100 KΩ
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Longitud de cable (metros)	100 m, par trenzado apantallado

A.8.4.2 Respuesta a escalón de las entradas analógicas integradas en la CPU

Tabla A-99 Respuesta a un escalón (ms), 0 V a 10 V medido a 95%

Selección de filtrado (valor medio de muestreo)	Supresión de frecuencias (tiempo de integración)		
	60 Hz	50 Hz	10 Hz
Ninguno (1 ciclo): Sin media	50 ms	50 ms	100 ms
Débil (4 ciclos): 4 muestreos	60 ms	70 ms	200 ms
Medio (16 ciclos): 16 muestreos	200 ms	240 ms	1150 ms
Fuerte (32 ciclos): 32 muestreos	400 ms	480 ms	2300 ms
Tiempo de muestreo	**4,17 ms**	**5 ms**	**25 ms**

A.8.4.3 Tiempo de muestreo para los puertos analógicos integrados en la CPU

Tabla A-100 Tiempo de muestreo para las entradas analógicas integradas en la CPU

Supresión de frecuencias (selección del tiempo de integración)	Tiempo de muestreo
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.8.4.4 Rangos de medida de entradas analógicas de tensión (CPU)

Tabla A-101 Representación de entradas analógicas para tensión (CPU)

Sistema		**Rango de medida de tensión**	
Decimal	**Hexadecimal**	**De 0 a 10 V**	
32767	7FFF	11,852 V	Rebase por exceso
32512	7F00		
32511	7EFF	11,759 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Valores negativos		Los valores negativos no se soportan	

A.8.4.5 Especificaciones de salidas analógicas

Tabla A-102 Salidas analógicas

Datos técnicos	**Descripción**
Número de salidas	2
Tipo	Intensidad
Rango total	0 a 20 mA
Rango total (palabra de datos)	de 0 a 27.648
Rango de sobreimpulso	De 20,01 a 23,52 mA
Rango de sobreimpulso (palabra de datos)	De 27649 a 32511
Rango de desbordamiento	Ver nota al pie [1]
Rango de desbordamiento (palabra de datos)	De 32512 a 32767
Resolución	10 bits
Impedancia de salida	≤500 Ω máx.
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	3,0%/3,5% de rango máximo
Tiempo de estabilización	2 ms
Longitud de cable (metros)	100 m, par trenzado apantallado

[1] En una situación de desbordamiento, las salidas analógicas se comportarán según los valores de las propiedades de configuración del dispositivo. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

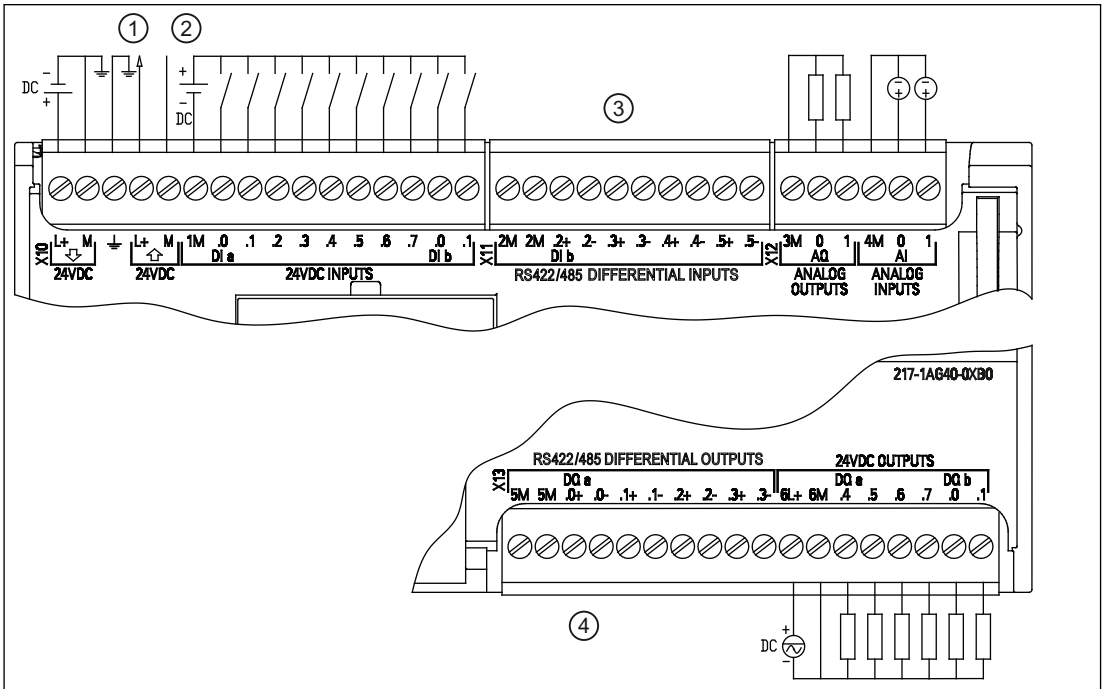
Tabla A-103 Representación de salidas analógicas para intensidad (CPU 1215C y CPU 1217C)

Sistema		Rango de salida de intensidad	
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	
32511	7EFF	23,52 mA	Rango de sobreimpulso
27649	6C01		
27648	6C00	20 mA	Rango nominal
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Valores negativos		Los valores negativos no se soportan	

[1] En una situación de rebase por exceso, las salidas analógicas se comportarán según los valores de las propiedades de configuración del dispositivo. En el parámetro "Reacción a STOP de la CPU", seleccione: "Aplicar valor sustitutivo" o "Mantener último valor".

A.8.5 Diagramas de cableado de la CPU 1217C

Tabla A-104 CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)



①	Alimentación de sensores 24 V DC Para una inmunidad a interferencias adicional, conecte "M" a masa incluso si no se utiliza la alimentación del sensor.
②	Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

③	Véase CPU 1217C Detalle y ejemplo de aplicación de entrada diferencial (DI) (Página 1321).
④	Véase CPU 1217C Detalle y ejemplo de aplicación de salida diferencial (DQ) (Página 1322).
Nota 1: Los conectores X12 deben ser de oro. Consulte el anexo C, Repuestos (Página 1439), para ver la referencia.	
Nota 2: Consulte Configuración de dispositivos (Página 137) para obtener información acerca del puerto Ethernet de la CPU.	

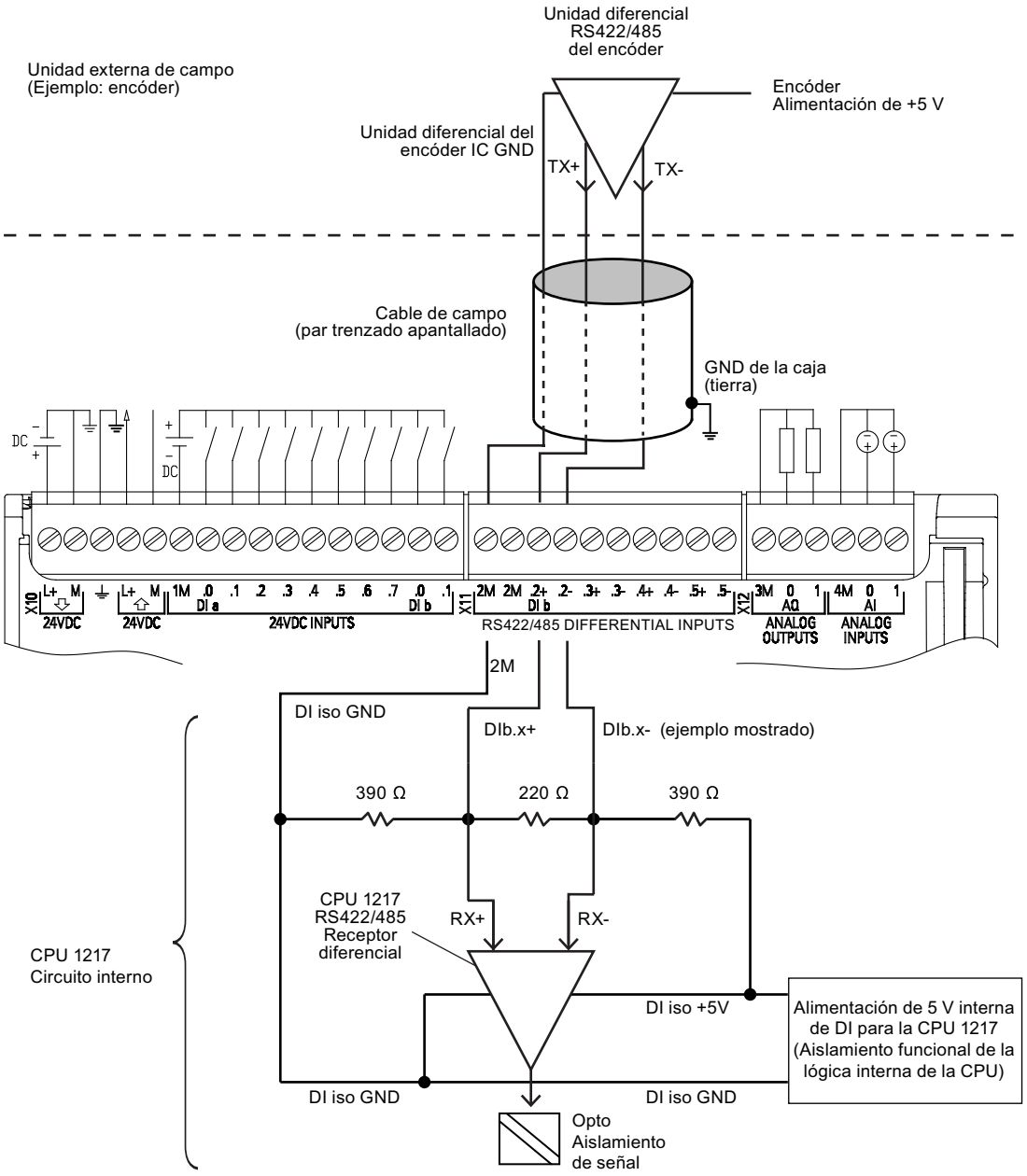
Tabla A-105 Asignación de pines de conectores para CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)

Pin	X10	X11	X12 (oro)	X13
1	L+ / 24 V DC	2M	3M	5M
2	M / 24 V DC	2M	AQ 0	5M
3	Tierra funcional	DI b.2+	AQ 1	DQ a.0+
4	L+ / salida de sensor 24 V DC	DI b.2-	4M	DQ a.0-
5	M / salida de sensor 24 V DC	DI b.3+	AI 0	DQ a.1+
6	1M	DI b.3-	AI 1	DQ a.1-
7	DI a.0	DI b.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

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

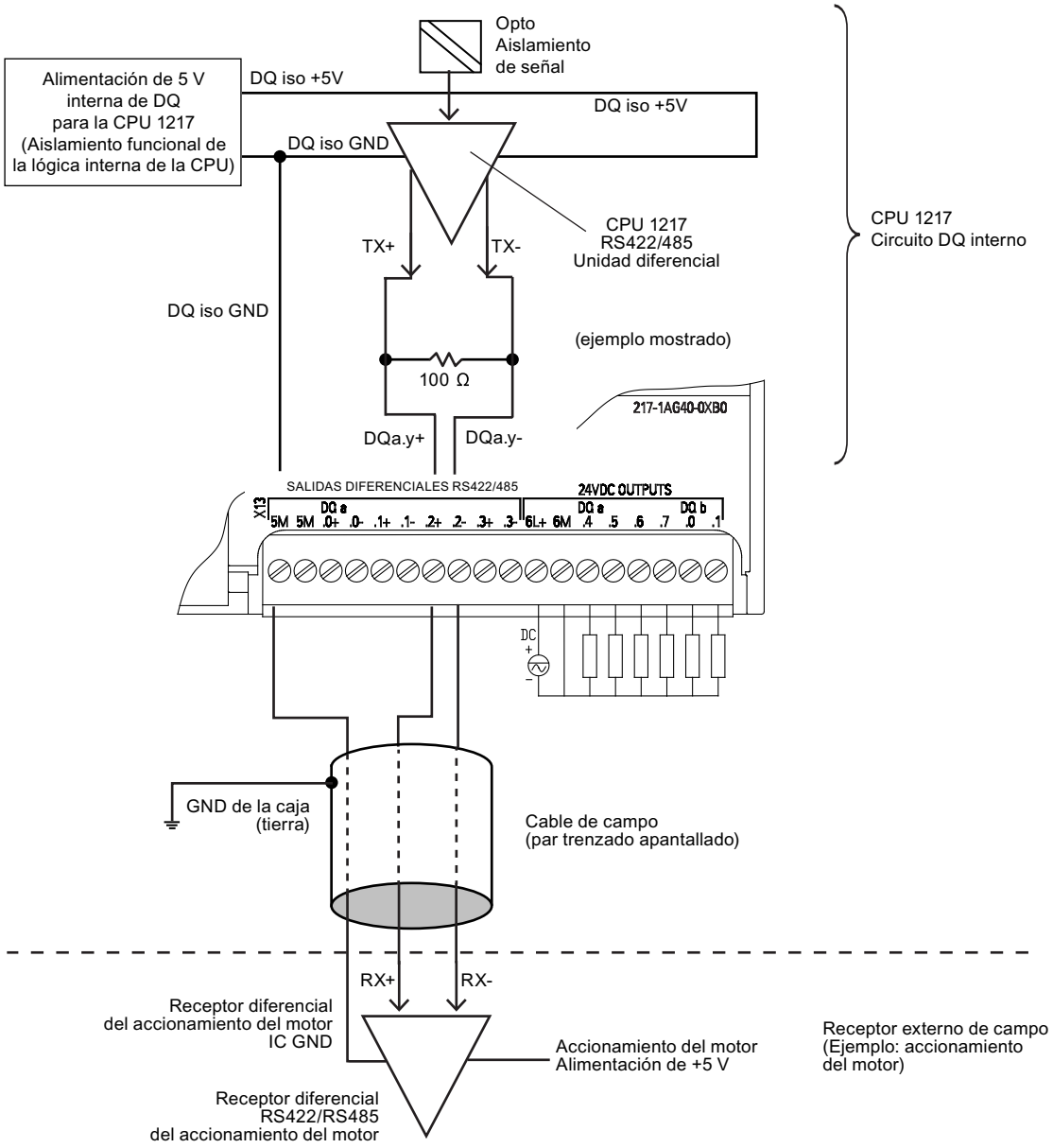
A.8.6 CPU 1217C Detalle y ejemplo de aplicación de entrada diferencial (DI)



Notas

- Cada DI diferencial está sin polarización cuando los tornillos del bloque de bornes tienen el circuito abierto.
- Terminación y polarización de la DI integrada = 100 Ω de impedancia equivalente.
- Los resistores de cierre y polarización de la DI integrada limitan el rango de tensión continua en modo común. Consulte los datos eléctricos para más detalles.

A.8.7 CPU 1217C Detalle y ejemplo de aplicación de salida diferencial (DQ)



Nota

- El resistor de cierre de la DQ integrada limita el rango de tensión continua en modo común. Consulte los datos eléctricos para más detalles.

A.9 Módulos de señales digitales (SMs)

A.9.1 Datos técnicos del módulo de entradas digitales SM 1221

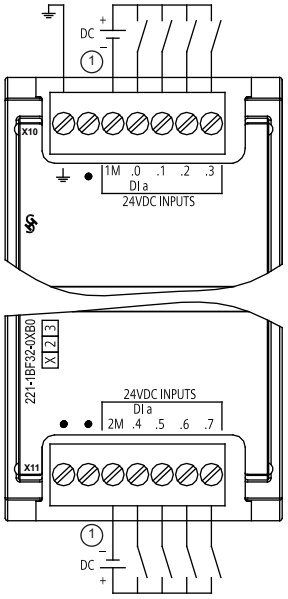
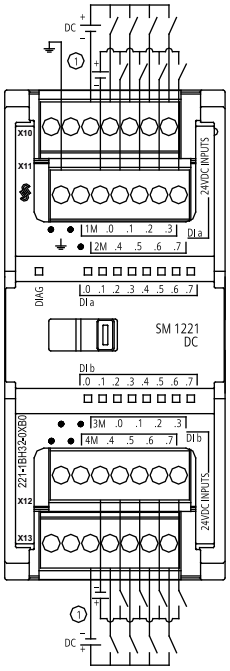
Tabla A-106 Especificaciones generales

Modelo	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Referencia	6ES7221-1BF32-0XB0	6ES7221-1BH32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	
Peso	170 gramos	210 gramos
Disipación de potencia	1,5 W	2,5 W
Consumo de corriente (bus SM)	105 mA	130 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada	

Tabla A-107 Entradas digitales

Modelo	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Número de entradas	8	16
Tipo	Sumidero/fuente (tipo 1 IEC sumidero)	
Tensión nominal	24 V DC a 4 mA, nominal	
Tensión continua admisible	30 V DC, máx.	
Sobretensión transitoria	35 V DC durante 0,5 s	
Señal 1 lógica (mín.)	15 V DC a 2,5 mA	
Señal 0 lógica (máx.)	5 V DC a 1 mA	
Aislamiento (de campo a lógica)	707 V DC (ensayo de tipo)	
Grupos de aislamiento	2	4
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms (seleccionable en grupos de 4)
Número de entradas ON simultáneamente	8	16
Longitud de cable (metros)	500 m apantallado, 300 no apantallado	

Tabla A-108 Diagramas de cableado de los SM de entradas digitales

SM 1221 DI 8 x 24 V DC (6ES7221-1BF32-0XB0)	SM 1221 DI 16 x 24 V DC (6ES7221-1BH32-0XB0)
	

① Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

Tabla A-109 Asignación de pines de conectores para SM 1221 DI 8 x 24 V DC (6ES7221-1BF32-0XB0)

Pin	X10	X11
1	Tierra funcional	Sin conexión
2	Sin conexión	Sin conexión
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

Tabla A-110 Asignación de pines de conectores para SM 1221 DI 16 x 24 V DC (6ES7221-1BH32-0XB0)

Pin	X10	X11	X12	X13
1	Sin conexión	Tierra funcional	Sin conexión	Sin conexión
2	Sin conexión	Sin conexión	Sin conexión	Sin conexión
3	1M	2M	3 M	4 M

Pin	X10	X11	X12	X13
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 Datos técnicos del módulo de salidas digitales SM 1222 de 8 salidas

Tabla A-111 Datos técnicos generales

Modelo	SM 1222 DQ 8 x Relay	SM 1222 DQ 8 Relay Changeover	SM 1222 DQ 8 x 24 V DC
Referencia	6ES7222-1HF32-0XB0	6ES7222-1XF32-0XB0	6ES7222-1BF32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Peso	190 gramos	310 gramos	180 gramos
Disipación de potencia	4,5 W	5 W	1,5 W
Consumo de corriente (bus SM)	120 mA	140 mA	120 mA
Consumo de corriente (24 V DC)	11 mA/bobina de relé utilizada	16,7 mA/bobina de relé utilizada	50 mA

Tabla A-112 Salidas digitales

Modelo	SM 1222 DQ 8 x Relay	SM 1222 DQ 8 Relay Changeover	SM 1222 DQ 8 x 24 V DC
Número de salidas	8	8	8
Tipo	Relé mecánico	Contacto inversor de relé	Estado sólido MOSFET (fuente)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	--	20 V DC mín.
Señal 0 lógica con carga de 10K Ω	--	--	0,1 V DC máx.
Intensidad (máx.)	2,0 A	2,0 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	30 W DC / 200 W AC	5 W
Resistencia de contacto en estado ON	Máx. 0,2 Ω (si son nuevas)	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.
Corriente de fuga por salida	--	--	10 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	7 A si están cerrados los contactos	8 A durante máx. 100 ms
Protección contra sobrecargas	No	No	
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y lógica)	1500 V AC (entre bobina y contacto)	707 V DC (examen de tipo)
Grupos de aislamiento	2	8	1
Intensidad por neutro (máx.)	10 A	2 A	4 A

Modelo	SM 1222 DQ 8 x Relay	SM 1222 DQ 8 Relay Changeover	SM 1222 DQ 8 x 24 V DC
Tensión de bloqueo inductiva	--	--	L+ menos 48 V, disipación de 1 W
Retardo de conmutación	10 ms máx.	10 ms máx.	50 µs máx. OFF a ON 200 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	1 Hz	--
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	10 000 000 ciclos apertura/cierre	--
Vida útil de los contactos bajo carga nominal (contacto NA)	100 000 ciclos apertura/cierre	100 000 ciclos apertura/cierre	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí		
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)		
Salidas paralelas para carga aumentada	No		
Número de salidas ON simultáneamente	8	• 4 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 8 a 55 °C en horizontal o 45 °C en vertical	8
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado

A.9.3 Datos técnicos del módulo de salidas digitales SM 1222 de 16 salidas

Tabla A-113 Datos técnicos generales

Modelo	SM 1222 DQ 16 x Relay	SM 1222 DQ 16 x 24 V DC	SM 1222 DQ 16 x 24 V DC sumidero
Referencia	6ES7222-1HH32-0XB0	6ES7222-1BH32-0XB0	6ES7222-1BH32-1XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Peso	260 gramos	220 gramos	220 gramos
Disipación de potencia	8,5 W	2,5 W	2,5 W
Consumo de corriente (bus SM)	135 mA	140 mA	140 mA
Consumo de corriente (24 V DC)	11 mA/bobina de relé utilizada	100 mA	40 mA

Tabla A-114 Salidas digitales

Modelo	**SM 1222 DQ 16 x Relay**	**SM 1222 DQ 16 x 24 V DC**	**SM 1222 DQ 16 x 24 V DC sumidero**
Número de salidas	16	16	16
Tipo	Relé mecánico	Estado sólido MOSFET (fuente)	Estado sólido - MOSFET (sinking)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	-	20 V DC mín.	0,5 V DC
Señal 0 lógica con carga de 10K Ω	-	0,1 V DC máx.	24 V (típico) menos 0,75 V DC
Intensidad (máx.)	2,0 A	0,5 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	5 W	5 W
Resistencia de contacto en estado ON	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.	0,5 Ω máx.
Corriente de fuga por salida	--	10 µA máx.	75 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	8 A durante máx. 100 ms	8 A durante máx. 100 ms
Protección contra sobrecargas	No		Sí, rango protegido con límite de intensidad De 1 A a 3,5 A
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y lógica)	707 V DC (examen de tipo)	707 V DC (examen de tipo)
Grupos de aislamiento	4	1	1
Intensidad por neutro (máx.)	10 A	8 A	Protegido con límite de intensidad
Tensión de bloqueo inductiva	-	L+ menos 48 V, disipación de 1 W	45 V
Retardo de conmutación	10 ms máx.	50 µs máx. OFF a ON 200 µs máx. ON a OFF	20 µs máx. OFF a ON 350 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	-	-
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	-	-
Vida útil de los contactos bajo carga nominal (contacto NA)	100 000 ciclos apertura/cierre	-	-
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí	Sí	Sí (entrada en fuente)
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)		
Salidas paralelas para carga aumentada	No		

Modelo	SM 1222 DQ 16 x Relay	SM 1222 DQ 16 x 24 V DC	SM 1222 DQ 16 x 24 V DC sumidero
Número de salidas ON simultánea-mente	• 8 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 16 a 55 °C en horizontal o 45 °C en vertical	16	16
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado		

Consulte también

Nuevas funciones (Página 35)

Tabla A-115 Diagramas de cableado de los SM de 8 salidas digitales

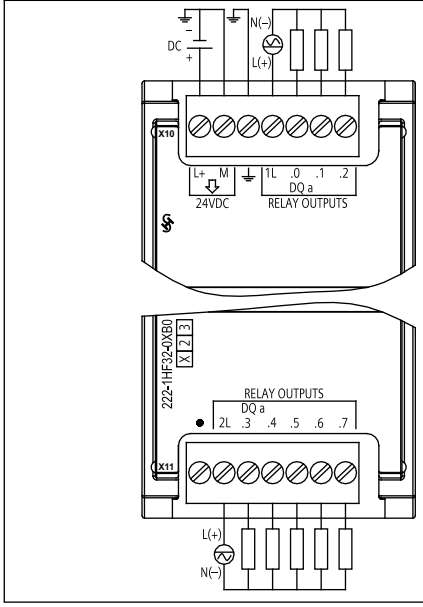
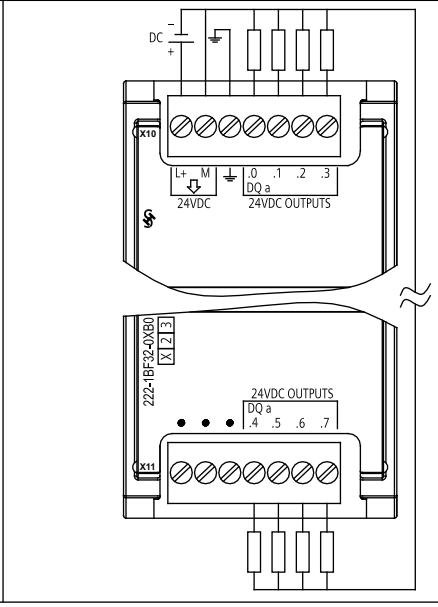
SM 1222 DQ 8 x Relay (6ES7222-1HF32-0XB0)	SM 1222 DQ 8 x 24 V DC (6ES7222-1BF32-0XB0)
	

Tabla A-116 Asignación de pines del conector para SM 1222 DQ 8 x Relay (6ES7222-1HF32-0XB0)

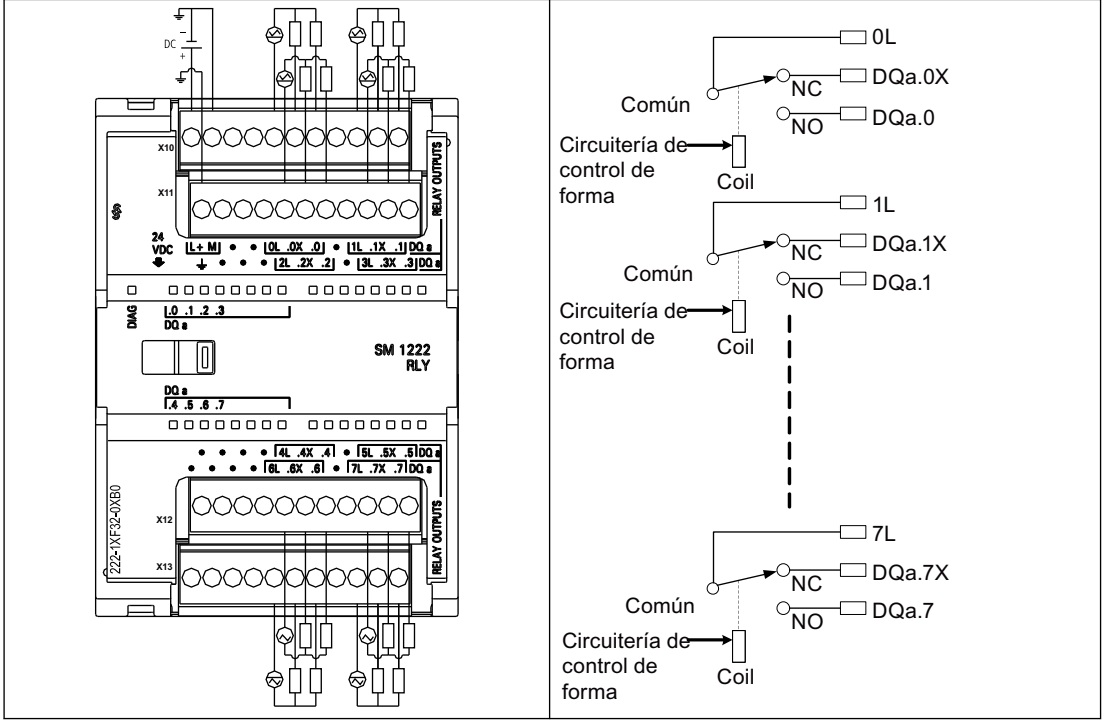
Pin	X10	X11
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	2L
3	Tierra funcional	DQ a.3
4	1L	DQ a.4
5	DQ a.0	DQ a.5
6	DQ a.1	DQ a.6
7	DQ a.2	DQ a.7

Tabla A-117 Asignación de pines del conector para SM 1222 DQ 8 x 24 V DC (6ES7222-1BF32-0XB0)

Pin	X10	X11
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	Sin conexión
3	Tierra funcional	Sin conexión
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

Tabla A-118 Diagrama de cableado del SM inversor de relé de 8 salidas digitales

SM 1222 DQ 8 x Relay Changeover (6ES7222-1XF32-0XB0)



Una salida de relé inversor controla dos circuitos mediante un borne común: un contacto normalmente cerrado y un contacto normalmente abierto. Si usamos "0" a modo de ejemplo, cuando la salida está desactivada, el (0L) común está conectado a un contacto normalmente cerrado (.0X) y está desconectado del contacto normalmente abierto (.0). Cuando la salida está activada, el (0L) común está desconectado del contacto normalmente cerrado (.0X) y está conectado al contacto normalmente abierto (.0).

Tabla A-119 Asignación de pines del conector para SM 1222 DQ 8 x Relay Changeover (6ES7222-1XF32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión

Pin	X10	X11	X12	X13
3	Sin conexión	Sin conexión	Sin conexión	Sin conexión
4	Sin conexión	Sin conexión	Sin conexión	Sin conexión
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	Sin conexión	Sin conexión	Sin conexión	Sin conexión
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

Tabla A-120 Diagramas de cableado de los SM de 16 salidas digitales

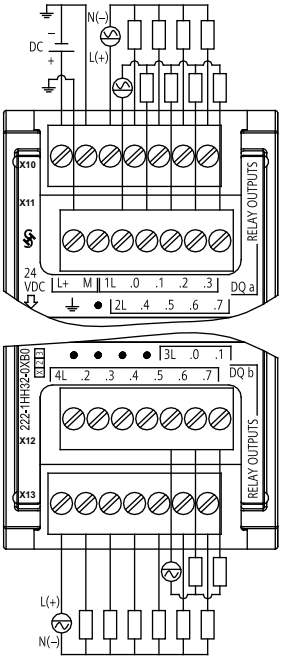
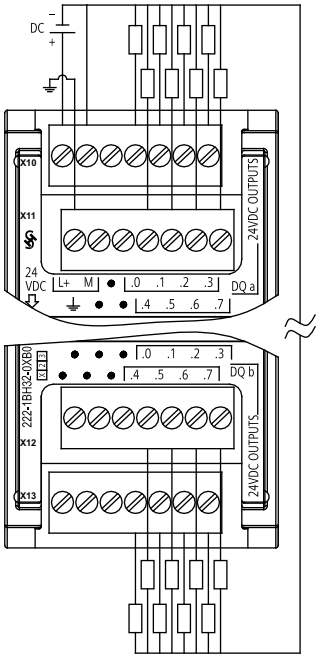
SM 1222 DQ 16 x Relay (6ES7222-1HH32-0XB0)	SM 1222 DQ 16 x 24 V DC (6ES7222-1BH32-0XB0)
	

Tabla A-121 Asignación de pines del conector para SM 1222 DQ 16 x Relay (6ES7222-1HH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	4L
2	M / 24 V DC	Sin conexión	Sin conexión	DQ b.2
3	1L	2L	Sin conexión	DQ b.3
4	DQ a.0	DQ a.4	Sin conexión	DQ b.4
5	DQ a.1	DQ a.5	3L	DQ b.5

Pin	X10	X11	X12	X13
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

Tabla A-122 Asignación de pines del conector para SM 1222 DQ 16 x 24 V DC (6ES7222-1BH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Sin conexión	Sin conexión	Sin conexión	Sin conexión
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

Tabla A-123 Diagramas de cableado de los SM de 16 salidas digitales en sumidero de 24 V DC

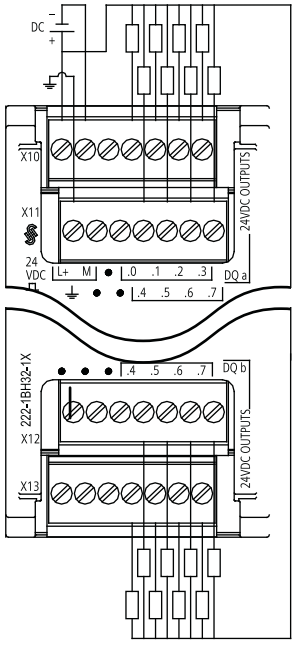
SM 1222 DQ 16 x 24 V DC sinking (6ES7222-1BH32-1XB0)	
	

Tabla A-124 Asignación de pines del conector para SM 1222 DQ 16 x 24 V DC sinking (6ES7222-1BH32-1XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Sin conexión	Sin conexión	Sin conexión	Sin conexión

Pin	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

A.9.4 Especificaciones de la SM 1223 entrada/salida digital V DC

Tabla A-125 Datos técnicos generales

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Referencia	6ES7223-1PH32-0XB0	6ES7223-1PL32-0XB0	6ES7223-1BH32-0XB0	6ES7223-1BL32-0XB0	6ES7223-1BL32-1XB0
Dimensiones A x A x P (mm)	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
Peso	230 gramos	350 gramos	210 gramos	310 gramos	310 gramos
Disipación de potencia	5,5 W	10 W	2,5 W	4,5 W	4,5 W
Consumo de corriente (bus SM)	145 mA	180 mA	145 mA	185 mA	185 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada 11 mA/bobina de relé utilizada		150 mA	200 mA	40 mA

Tabla A-126 Entradas digitales

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Número de entradas	8	16	8	16	16
Tipo	Sumidero/fuente (IEC tipo 1, sumidero)	Sumidero/fuente (IEC tipo 1, sumidero)	Sumidero/fuente (IEC tipo 1, sumidero)	Sumidero/fuente (IEC tipo 1, sumidero)	Sumidero/fuente (IEC tipo 1, sumidero)
Tensión nominal	24 V DC a 4 mA, nominal	24 V DC a 4 mA, nominal	24 V DC a 4 mA, nominal	24 V DC a 4 mA, nominal	24 V DC a 4 mA, nominal
Tensión continua admisible	30 V DC máx.	30 V DC máx.	30 V DC máx.	30 V DC máx.	30 V DC máx.
Sobretensión transitoria	35 V DC durante 0,5 s	35 V DC durante 0,5 s	35 V DC durante 0,5 s	35 V DC durante 0,5 s	35 V DC durante 0,5 s
Señal 1 lógica (mín.)	15 V DC a 2,5 mA	15 V DC a 2,5 mA	15 V DC a 2,5 mA	15 V DC a 2,5 mA	15 V DC a 2,5 mA
Señal 0 lógica (máx.)	5 V DC a 1 mA	5 V DC a 1 mA	5 V DC a 1 mA	5 V DC a 1 mA	5 V DC a 1 mA
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)	707 V DC (examen de tipo)	707 V DC (examen de tipo)	707 V DC (examen de tipo)	707 V DC (examen de tipo)
Grupos de aislamiento	2	2	2	2	2

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Tiempos de filtro	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 y 12,8 ms, seleccionable en grupos de 4
Número de entradas ON simultáneamente	8	• 8 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 16 a 55 °C en horizontal o 45 °C en vertical	8	16	16
Longitud de cable (metros)	500 m apantallado, 300 m no apantallado	500 m apantallado, 300 m no apantallado	500 m apantallado, 300 m no apantallado	500 m apantallado, 300 m no apantallado	500 m apantallado, 300 m no apantallado

Tabla A-127 Salidas digitales

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Número de salidas	8	16	8	16	16
Tipo	Relé mecánico	Relé mecánico	Estado sólido MOSFET (fuente)	Estado sólido MOSFET (fuente)	Estado sólido - MOSFET (sinking)
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC	De 5 a 30 V DC o de 5 a 250 V AC	De 20,4 a 28,8 V DC	De 20,4 a 28,8 V DC	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	--	--	20 V DC, mín.	20 V DC, mín.	0,5 V DC
Señal 0 lógica con carga de 10 KΩ	--	--	0,1 V DC, máx.	0,1 V DC, máx.	24 V (típico) menos 0,75 V DC
Intensidad (máx.)	2,0 A	2,0 A	0,5 A	0,5 A	0,5 A
Carga de lámparas	30 W DC / 200 W AC	30 W DC / 200 W AC	5 W	5 W	5 W
Resistencia de contacto en estado ON	Máx. 0,2 Ω (si son nuevas)	Máx. 0,2 Ω (si son nuevas)	0,6 Ω máx.	0,6 Ω máx.	0,5 Ω máx.
Corriente de fuga por salida	--	--	10 µA máx.	10 µA máx.	75 µA máx.
Extracorriente de conexión	7 A si están cerrados los contactos	7 A si están cerrados los contactos	8 A durante máx. 100 ms	8 A durante máx. 100 ms	Protegido con límite de intensidad

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Protección contra sobrecargas	No	No	No	No	Sí, rango protegido con límite de intensidad De 1 A a 3,5 A
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y lógica)	707 V DC (examen de tipo)	707 V DC (examen de tipo)	707 V DC (examen de tipo)
Grupos de aislamiento	2	4	1	1	1
Intensidad por neutro	10A	8 A	4 A	8 A	8 A
Tensión de bloqueo inductiva	--	--	L+ menos 48 V, disipación de 1 W	L+ menos 48 V, disipación de 1 W	45 V
Retardo de conmutación	10 ms máx.	10 ms máx.	50 µs máx. OFF a ON 200 µs máx. ON a OFF	50 µs máx. OFF a ON 200 µs máx. ON a OFF	20 µs máx. OFF a ON 350 µs máx. ON a OFF
Frecuencia máxima de conmutación de relé	1 Hz	1 Hz	--	--	--
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre	10 000 000 ciclos apertura/cierre	--	--	--
Vida útil de los contactos bajo carga nominal (contacto NA)	100 000 ciclos apertura/cierre	100 000 ciclos apertura/cierre	--	--	--
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí	Sí	Sí	Sí	Sí (entrada en fuente)
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)	Sí (con el mismo neutro)	Sí (con el mismo neutro)	Sí (con el mismo neutro)	Sí (con el mismo neutro)
Salidas paralelas para carga aumentada	No	No	No	No	No

Modelo	**SM 1223 DI 8 x 24 V DC, DQ 8 x Relay**	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay**	**SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC**	**SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking**
Número de salidas ON simultáneamente	8	• 8 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 16 a 55 °C en horizontal o 45 °C en vertical	8	16	16
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado	500 m apantallado, 150 m no apantallado

Consulte también

Nuevas funciones (Página 35)

Tabla A-128 Diagramas de cableado de SM entrada digital V DC/salida de relé

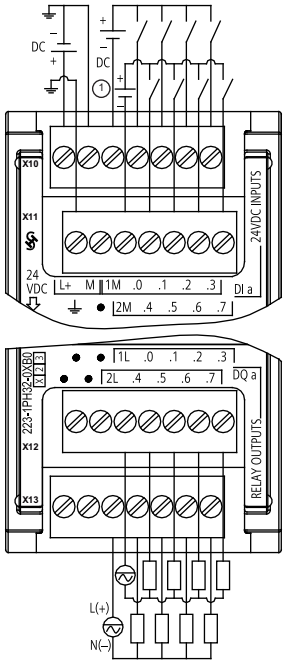
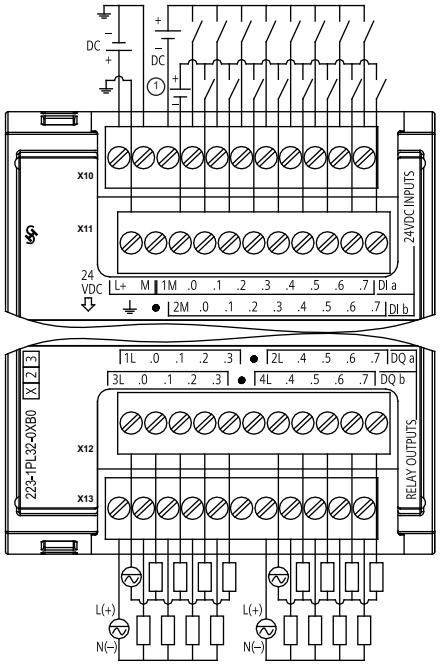
SM 1223 DI 8 x 24 V DC, DQ 8 x Relay (6ES7223-1PH32-0XB0)	**SM 1223 DI 16 x 24 V DC, DQ 16 x Relay (6ES7223-1PL32-0XB0)**	**Notas**
		1 Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

Tabla A-129 Asignación de pines del conector para SM 1223 DI 8 x 24 V DC, DQ 8 x Relay (6ES7223-1PH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
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

Tabla A-130 Asignación de pines del conector para SM 1223 DI 16 x 24 V DC, DQ 16 x Relay (6ES7223-1PL32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	1L	3L
2	M / 24 V DC	Sin conexión	DQ a.0	DQ b.0
3	1M	2M	DQ a.1	DQ b.1
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	Sin conexión	Sin conexión
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

Tabla A-131 Diagramas de cableado de SM de entradas digitales V DC/salida

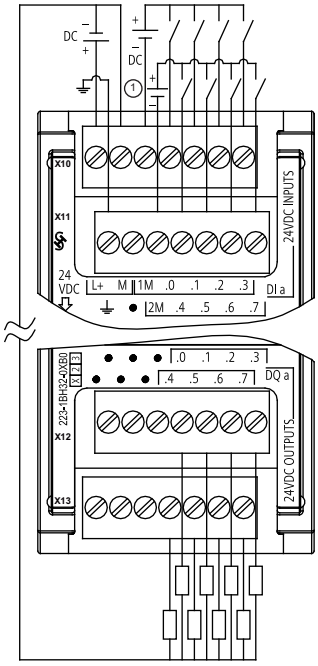
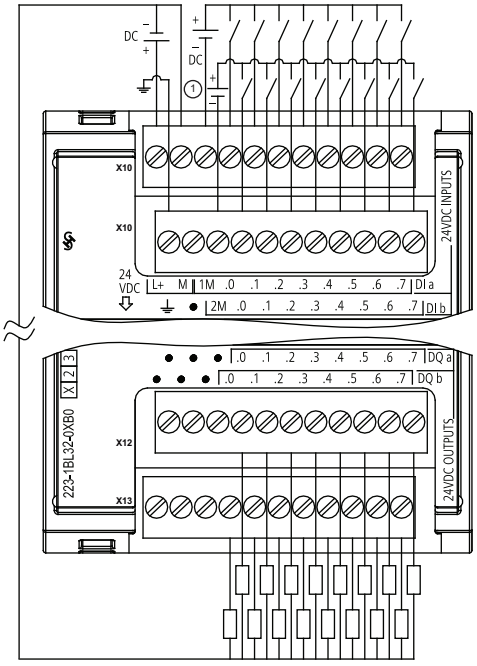
SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC (6ES7223-1BH32-0XB0)	SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC (6ES7223-1BL32-0XB0)	Notas
		(1) Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

Tabla A-132 Asignación de pines del conector para SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC (6ES7223-1BH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	1M	2M	Sin conexión	Sin conexión
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

Tabla A-133 Asignación de pines del conector para SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC (6ES7223-1BL32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	1M	2M	Sin conexión	Sin conexión
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

Pin	X10	X11	X12	X13
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

Tabla A-134 Diagrama de cableado de SM de entradas digitales V DC/salida

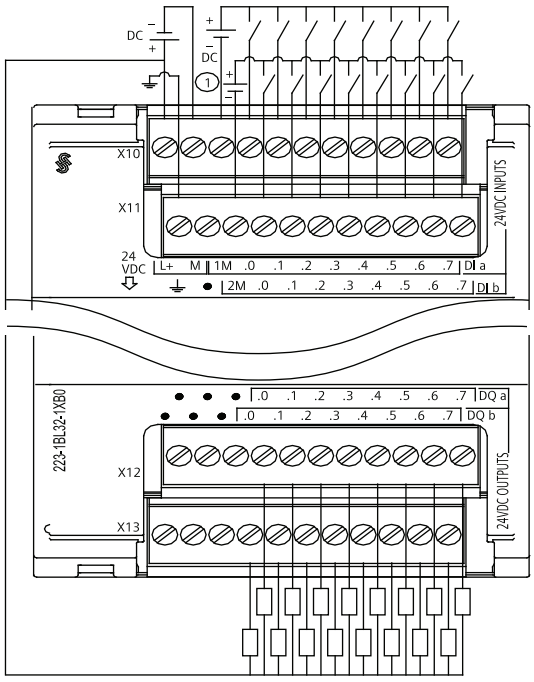
SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking (6ES7223-1BL32-1XB0)	Notas
	(1) Para entradas en sumidero, conecte "-" a "M" (como se indica). Para entradas en fuente, conecte "+" a "M".

Tabla A-135 Asignación de pines del conector para SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking (6ES7223-1BL32-1XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	1M	2M	Sin conexión	Sin conexión
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

Pin	X10	X11	X12	X13
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 Especificaciones de la SM 1223 entrada/salida digital V AC

Tabla A-136 Datos técnicos generales

Modelo	SM 1223 DI 8 x120/230 V AC / DQ 8 x relé
Referencia	6ES7223-1QH32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75 mm
Peso	190 gramos
Disipación de potencia	7,5 W
Consumo de corriente (bus SM)	120 mA
Consumo de corriente (24 V DC)	11 mA por salida en estado ON

Tabla A-137 Entradas digitales

Modelo	SM 1223 DI 8 x 120/230 V AC / DQ 8 x relé
Número de entradas	8
Tipo	Tipo 1 IEC
Tensión nominal	120 V AC a 6 mA, 230 V AC a 9 mA
Tensión continua admisible	264 V AC
Sobretensión transitoria	--
Señal 1 lógica (mín.)	79 V AC a 2,5 mA
Señal 0 lógica (máx.)	20 V AC a 1 mA
Corriente de fuga (máx.)	1 mA
Aislamiento (de campo a lógica)	1500 V AC
Grupos de aislamiento[1]	4
Tiempos de retardo de las entradas	Típ. 0,2 a 12,8 ms, seleccionable por el usuario Máximo: -
Conexión de sensor de proximidad de 2 hilos (Bero) (máx.)	1 mA
Longitud del cable	No apantallado: 300 metros Apantallado: 500 metros
Número de entradas ON simultáneamente	8

[1] Los canales dentro de un grupo deben ser de la misma fase.

Tabla A-138 Salidas digitales

Modelo	**SM 1223 DI 8 x 120/230 V AC / DQ 8 x relé**
Número de salidas	8
Tipo	Relé mecánico
Rango de tensión	De 5 a 30 V DC o de 5 a 250 V AC
Señal 1 lógica a intensidad máx.	--
Señal 0 lógica con carga de 10K Ω	--
Intensidad (máx.)	2,0 A
Carga de lámparas	30 W DC / 200 W AC
Resistencia de contacto en estado ON	Máx. 0,2 Ω (si son nuevas)
Corriente de fuga por salida	--
Extracorriente de conexión	7 A si están cerrados los contactos
Protección contra sobrecargas	No
Aislamiento (de campo a lógica)	1500 V AC (entre bobina y contacto) Ninguno (entre bobina y lógica)
Grupos de aislamiento	2
Intensidad por neutro (máx.)	10 A
Tensión de bloqueo inductiva	--
Retardo de conmutación (máx.)	10 ms
Frecuencia máxima de conmutación de relé	1 Hz
Vida útil mecánica (sin carga)	10 000 000 ciclos apertura/cierre
Vida útil de los contactos bajo carga nominal	100 000 ciclos apertura/cierre
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí
Salidas paralelas para control de carga redundante	Sí (con el mismo neutro)
Salidas paralelas para carga aumentada	No
Número de salidas ON simultáneamente	• 4 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 8 a 55 °C en horizontal o 45 °C en vertical
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado

Tabla A-139 SM 1223 DI 8 x 120/230 V AC, DQ 8 x relé (6ES7223-1QH32-0XB0)

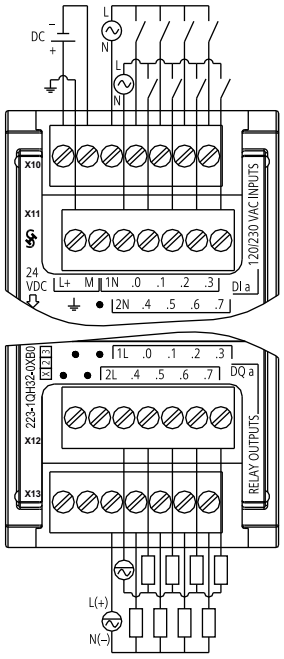


Tabla A-140 Asignación de pines de conectores para SM 1223 DI 8 x 120/240 V AC, DQ 8 x relé (6ES7223-1QH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Tierra funcional	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
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 Módulos de señales analógicos (SMs)

A.10.1 Datos técnicos del módulo de entradas analógicas SM 1231

Tabla A-141 Especificaciones generales

Modelo	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Referencia	6ES7231-4HD32-0XB0	6ES7231-4HF32-0XB0	6ES7231-5ND32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75		

Modelo	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Peso	180 gramos		
Disipación de potencia	2,2 W	2,3 W	2,0 W
Consumo de corriente (bus SM)	80 mA	90 mA	80 mA
Consumo de corriente (24 V DC)	45 mA		65 mA

Tabla A-142 Entradas analógicas

Modelo	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Número de entradas	4	8	4
Tipo	Tensión o intensidad (diferencial): Seleccionable en grupos de 2		Tensión o intensidad (diferencial)
Rango	±10 V, ±5 V, ±2,5 V, de 0 a 20 mA o de 4 mA a 20 mA		±10 V, ±5 V, ±2,5 V, ±1,25 V, 0 a 20 mA o 4 mA a 20 mA
Rango total (palabra de datos)	Tensión de -27.648 a 27.648 / intensidad de 0 a 27.648		
Rango de sobreimpulso/subimpulso (palabra de datos) Consulte el apartado en que aparecen los rangos de entrada analógica de tensión e intensidad (Página 1352).	Tensión: de 32.511 a 27.649 /de -27.649 a -32.512 Intensidad: de 32.511 a 27.649 / de 0 a -4.864		
Rebase por exceso/defecto (palabra de datos) Consulte el apartado en que aparecen los rangos de entrada de tensión e intensidad (Página 1352).	Tensión: de 32.767 a 32.512 / de -32.513 a -32.768 Intensidad de 0 a 20 mA: de 32.767 a 32.512 / de -4.865 a -32.768 Intensidad de 4 a 20 mA: de 32.767 a 32.512 (los valores por debajo de -4.864 indican rotura de hilo)		
Resolución1	12 bits + bit de signo		15 bits + bit de signo
Tensión/intensidad soportada máxima	±35 V / ±40 mA		
Filtrado	Ninguno, débil, medio o fuerte Consulte el apartado en que aparecen los tiempos de respuesta a un escalón (Página 1351).		
Supresión de ruido	400, 60, 50 o 10 Hz Consulte el apartado en que aparecen las frecuencias de muestreo (Página 1351).		
Impedancia de entrada Antes de parametrización Tensión Intensidad	 ≥ 1 MΩ ≥ 9 MΩ, FS 06 y superior ≥ 1 MΩ ≥ 270 Ω, < 290 Ω		 ≥ 1 MΩ ≥ 1 MΩ < 315 Ω, > 280 Ω
Aislamiento De campo a lógica De lógica a 24 V DC De campo a 24 V DC a lógica Entre canales	Ninguno		 707 V DC (ensayo de tipo) 707 V DC (ensayo de tipo) 500 V DC (ensayo de tipo) Ninguno
Precisión (25 °C / de -20 a 60 °C)	±0,1% / ±0,2% de rango máximo		±0,1%/±0,3% de rango máximo
Principio de medición	Conversión de valor real		
Rechazo en modo común	40 dB, DC a 60 Hz		

Modelo	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Rango de señales operativo[1]	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V		
Longitud de cable (metros)	100 m, trenzado y apantallado		

[1] La aplicación de tensiones que estén fuera del rango operativo a un canal puede causar interferencias en los demás canales.

Tabla A-143 Diagnóstico

Modelo	SM 1231 AI 4 x 13 bits	SM 1231 AI 8 x 13 bits	SM 1231 AI 4 x 16 bits
Rebase por exceso/defecto	Sí		
24 V DC baja tensión	Sí		
Rotura de hilo	Solo en el rango 4 a 20 mA (si la entrada es inferior a -4864; 1,185 mA)		

Medición de intensidad SM 1231

Se puede implementar la medición de intensidad con un transductor de 2 hilos o uno de 4 hilos de la forma representada en la siguiente figura:

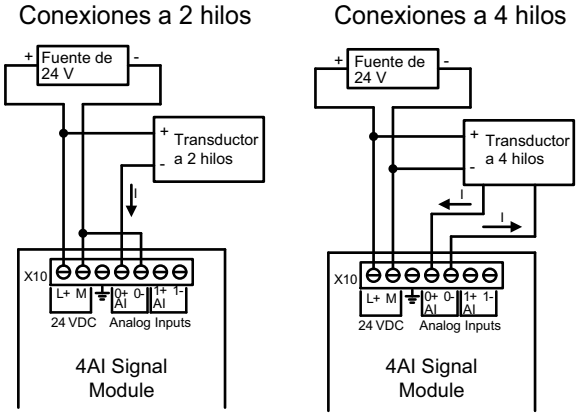


Tabla A-144 Diagramas de cableado de los SM de entradas analógicas

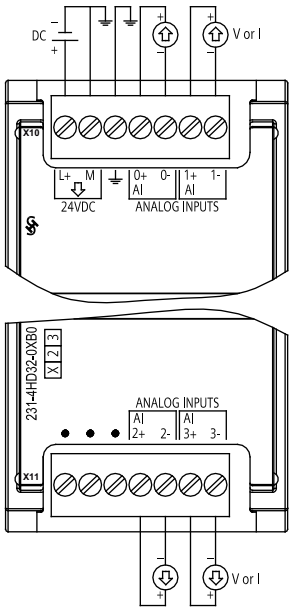
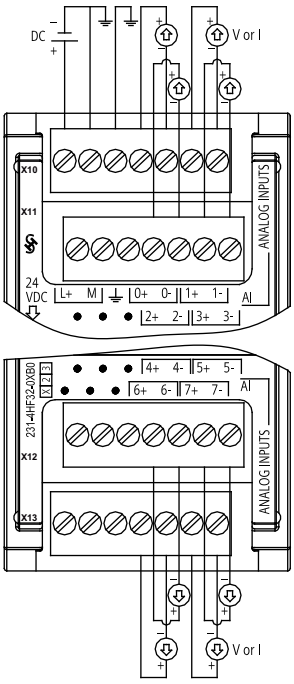
SM 1231 AI 4 x 13 bits (6ES7231-4HD32-0XB0)	SM 1231 AI 8 x 13 bits (6ES7231-4HF32-0XB0)
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-145 Asignación de pines de conectores para SM 1231 AI 4 x 13 bits (6ES7231-4HD32-0XB0)

Pin	X10 (oro)	X11 (oro)
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	Sin conexión
3	Tierra funcional	Sin conexión
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Tabla A-146 Asignación de pines de conectores para SM 1231 AI 8 x 13 bits (6ES7231-4HF32-0XB0)

Pin	X10 (oro)	X11 (oro)	X12 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión	Sin conexión
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-

Tabla A-147 Diagrama de cableado del SM de entrada analógica

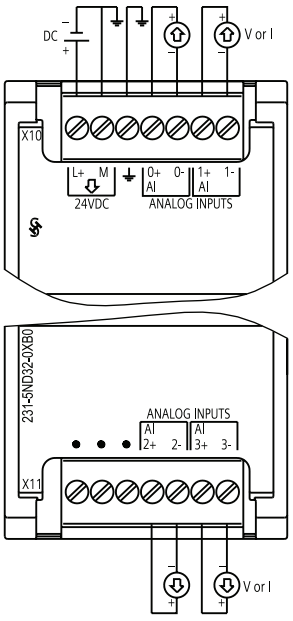
SM 1231 AI 4 x 16 bits (6ES7231-5ND32-0XB0)	
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-148 Asignación de pines de conectores para SM 1231 AI 4 x 16 bits (6ES7231-5ND32-0XB0)

Pin	X10 (oro)	X11 (oro)
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	Sin conexión
3	Tierra funcional	Sin conexión
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Nota

Los canales de entrada de tensión que no se utilicen deben cortocircuitarse.

Los canales de entrada de intensidad que no se utilicen deben ponerse a entre 0 y 20 mA y/o debe deshabilitarse la notificación de error por rotura de hilo.

Las entradas configuradas para el modo de intensidad no conducirán corriente en bucle a no ser que el módulo esté configurado y reciba alimentación.

Los canales de entrada de intensidad no funcionarán a no ser que el transmisor reciba alimentación externa.

A.10.2 Datos técnicos del módulo de salidas analógicas SM 1232

Tabla A-149 Especificaciones generales

Datos técnicos	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Referencia	6ES7232-4HB32-0XB0	6ES7232-4HD32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	
Peso	180 gramos	
Disipación de potencia	1,8 W	2,0 W
Consumo de corriente (bus SM)	80 mA	
Consumo de corriente (24 V DC)	45 mA (sin carga)	

Tabla A-150 Salidas analógicas

Datos técnicos	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Número de salidas	2	4
Tipo	Tensión o intensidad	
Rango	±10 V, de 0 a 20 mA o de 4 mA a 20 mA	
Resolución	Tensión: 14 bits Intensidad: 13 bits	
Rango total (palabra de datos)	Tensión: -27.648 a 27.648; intensidad: 0 a 27.648 Consulte los rangos de salida de tensión e intensidad (Página 1353).	
Precisión (25 °C / de -20 a 60 °C)	±0,3% / ±0,6% de rango máximo	
Tiempo de estabilización (95% del nuevo valor)	Tensión: 300 µs (R), 750 µs (1 uF) Intensidad: 600 µs (1 mH), 2 ms (10 mH)	
Impedancia de carga	Tensión: ≥ 1000 Ω Intensidad: ≤ 600 Ω	
Corriente de cortocircuito máxima en la salida	Modo de tensión: ≤ 24 mA Modo de intensidad: ≥ 38,5 mA	
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	
Aislamiento (de campo a lógica)	Ninguno	
Aislamiento (de 24 V a salida)	Ninguno	
Longitud de cable (metros)	100 m, trenzado y apantallado	

Tabla A-151 Diagnóstico

Datos técnicos	SM 1232 AQ 2 x 14 bit	SM 1232 AQ 4 x 14 bit
Rebase por exceso/defecto	Sí	
Cortocircuito a tierra (solo en modo de tensión)	Sí	
Rotura de hilo (solo en modo de intensidad)[1]	Sí	
24 V DC baja tensión[2]	Sí	

[1] La detección de cortocircuito solo es posible si la tensión de salida es inferior a -0,5 V o superior a +0,5 V.

[2] La detección de rotura de hilo solo es posible si la intensidad de salida es superior a 1 mA.

Tabla A-152 Diagramas de cableado de los SM de salidas analógicas

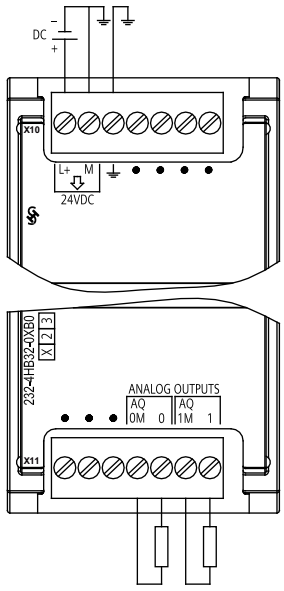
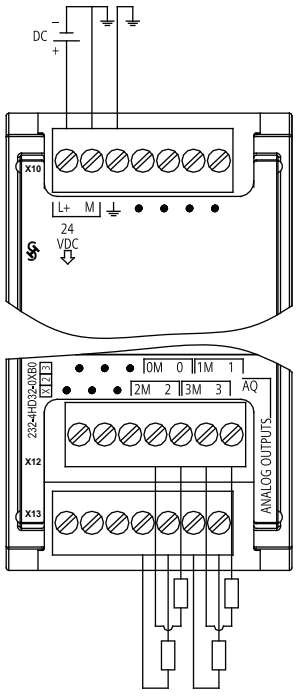
SM 1232 AQ 2 x 14 bits (6ES7232-4HB32-0XB0)	**SM 1232 AQ 4 x 14 bits (6ES7232-4HD32-0XB0)**
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-153 Asignación de pines de conectores para SM 1232 AQ 2 x 14 bits (6ES7232-4HB32-0XB0)

Pin	X10 (oro)	X11 (oro)
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	Sin conexión
3	Tierra funcional	Sin conexión
4	Sin conexión	AQ 0M
5	Sin conexión	AQ 0
6	Sin conexión	AQ 1M
7	Sin conexión	AQ 1

Tabla A-154 Asignación de pines de conectores para SM 1232 AQ 4 x 14 bits (6ES7232-4HD32-0XB0)

Pin	X10 (oro)	X12 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión
4	Sin conexión	AQ 0M	AQ 2M
5	Sin conexión	AQ 0	AQ 2

Pin	X10 (oro)	X12 (oro)	X13 (oro)
6	Sin conexión	AQ 1M	AQ 3M
7	Sin conexión	AQ 1	AIQ 3

A.10.3 Datos técnicos del módulo de entradas/salidas analógicas SM 1234

Tabla A-155 Especificaciones generales

Datos técnicos	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Referencia	6ES7234-4HE32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75
Peso	220 gramos
Disipación de potencia	2,4 W
Consumo de corriente (bus SM)	80 mA
Consumo de corriente (24 V DC)	60 mA (sin carga)

Tabla A-156 Entradas analógicas

Modelo	SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits
Número de entradas	4
Tipo	Tensión o intensidad (diferencial): Seleccionable en grupos de 2
Rango	±10 V, ±5 V, ±2,5 V, de 0 a 20 mA o de 4 mA a 20 mA
Rango total (palabra de datos)	De -27.648 a 27.648
Rango de sobreimpulso/subimpulso (palabra de datos)	Tensión: de 32.511 a 27.649 /de 27.649 a -32512 Intensidad: de 32.511 a 27.649 / de 0 a -4.864 Consulte el apartado en que aparecen los rangos de entrada de tensión e intensidad (Página 1352).
Rebase por exceso/defecto (palabra de datos)	Tensión: de 32.767 a 32.512 / de -32.513 a -32.768 Intensidad: de 32.767 a 32.512 / de -4.865 a -32.768 Consulte el apartado en que aparecen los rangos de entrada de tensión e intensidad (Página 1352).
Resolución	12 bits + bit de signo
Tensión/intensidad soportada máxima	±35 V / ±40 mA
Filtrado	Ninguno, débil, medio o fuerte Consulte el apartado en que aparecen los tiempos de respuesta a un escalón (Página 1351).
Supresión de ruido	400, 60, 50 o 10 Hz Consulte el apartado en que aparecen las frecuencias de muestreo (Página 1351).
Impedancia de entrada	≥ 9 MΩ, FS 07 y superior ≥ 1 MΩ (tensión) / ≥ 270 Ω, < 290 Ω (intensidad)
Aislamiento (de campo a lógica)	Ninguno
Precisión (25 °C / de -20 a 60 °C)	±0,1% / ±0,2% de rango máximo
Tiempo de conversión analógica-digital	625 µs (rechazo de 400 Hz)

Modelo	**SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits**
Rechazo en modo común	40 dB, DC a 60 Hz
Rango de señales operativo[1]	La tensión de señal más la tensión en modo común debe ser menor que +12 V y mayor que -12 V
Longitud de cable (metros)	100 m, trenzado y apantallado

[1] La aplicación de tensiones que estén fuera del rango operativo a un canal puede causar interferencias en los demás canales.

Tabla A-157 Salidas analógicas

Datos técnicos	**SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits**
Número de salidas	2
Tipo	Tensión o intensidad
Rango	±10 V, de 0 a 20 mA o de 4 mA a 20 mA
Resolución	Tensión: 14 bits; intensidad: 13 bits
Rango total (palabra de datos)	Tensión: -27648 a 27648; intensidad: de 0 a 27.648 Consulte el apartado en que aparecen los rangos de salida de tensión e intensidad (Página 1353).
Precisión (25 °C / de -20 a 60 °C)	±0,3% / ±0,6% de rango máximo
Tiempo de estabilización (95% del nuevo valor)	Tensión: 300 µs (R), 750 µs (1 uF) Intensidad: 600 µs (1 mH), 2 ms (10 mH)
Impedancia de carga	Tensión: ≥ 1000 Ω Intensidad: ≤ 600 Ω
Corriente de cortocircuito máxima en la salida	Modo de tensión: ≤ 24 mA Modo de intensidad: ≥ 38,5 mA
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Aislamiento (de campo a lógica)	Ninguno
Aislamiento (de 24 V a salida)	Ninguno
Longitud de cable (metros)	100 m, trenzado y apantallado

Tabla A-158 Diagnóstico

Modelo	**SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits**
Rebase por exceso/defecto	Sí
Cortocircuito a tierra (solo en modo de tensión)[1]	Sí en las salidas
Rotura de hilo (solo en modo de intensidad)[2]	Sí en las salidas
24 V DC baja tensión	Sí

[1] La detección de cortocircuito solo es posible si la tensión de salida es inferior a -0,5 V o superior a +0,5 V.

[2] La detección de rotura de hilo solo es posible si la intensidad de salida es superior a 1 mA.

Medición de intensidad SM 1234

Se puede implementar la medición de intensidad con un transductor de 2 hilos o uno de 4 hilos de la forma representada en la siguiente figura:

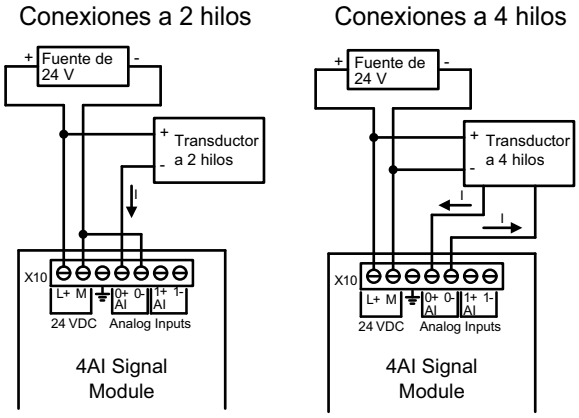


Tabla A-159 Diagramas de cableado de los SM de entradas/salidas analógicas

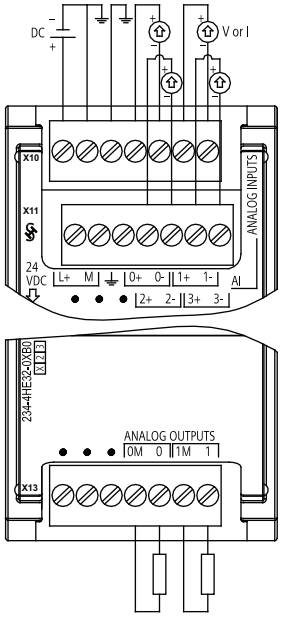
SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits (6ES7234-4HE32-0XB0)	
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-160 Asignación de pines de conectores para SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits (6ES7234-4HE32-0XB0)

Pin	X10 (oro)	X11 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión

Pin	X10 (oro)	X11 (oro)	X13 (oro)
3	Tierra funcional	Sin conexión	Sin conexión
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

Nota

Los canales de entrada de tensión que no se utilicen deben cortocircuitarse.

Los canales de entrada de intensidad que no se utilicen deben ponerse a entre 0 y 20 mA y/o debe deshabilitarse la notificación de error por rotura de hilo.

Las entradas configuradas para el modo de intensidad no conducirán corriente en bucle a no ser que el módulo esté configurado y reciba alimentación.

Los canales de entrada de intensidad no funcionarán a no ser que el transmisor reciba alimentación externa.

A.10.4 Respuesta a un escalón de las entradas analógicas

Tabla A-161 Respuesta a un escalón (ms), de 0 a rango máximo, medido al 95%

Selección de filtrado (valor medio de muestreo)	Reducción de ruido/supresión de frecuencias (selección del tiempo de integración)			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Ninguno (1 ciclo): Sin media	4 ms	18 ms	22 ms	100 ms
Débil (4 ciclos): 4 muestreos	9 ms	52 ms	63 ms	320 ms
Medio (16 ciclos): 16 muestreos	32 ms	203 ms	241 ms	1200 ms
Fuerte (32 ciclos): 32 muestreos	61 ms	400 ms	483 ms	2410 ms

A.10.5 Tiempo de muestreo y tiempos de actualización para entradas analógicas

Tabla A-162 Tiempos de muestreo y de actualización del módulo para todos los canales

Supresión de frecuencias (tiempo de integración)	Tiempos de muestreo y de actualización del módulo para todos los canales			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
SM de 4 canales x 13 bits	0,625 ms	4,17 ms	5 ms	25 ms
SM de 8 canales x 13 bits	1,25 ms	4,17 ms	5 ms	25 ms
SM de 4 canales x 16 bits	0,417 ms	0,397 ms	0,400 ms	0,400 ms

A.10.6 Rangos de medida de entradas analógicas de tensión e intensidad (SB y SM)

Tabla A-163 Representación de entradas analógicas de tensión (SB y SM)

Sistema		**Rango de medida de tensión**				
Decimal	**Hexadecimal**	**±10 V**	**±5 V**	**±2,5 V**	**±1,25 V**	
32767	7FFF[1]	11,851 V	5,926 V	2,963 V	1,481 V	Rebase por exceso
32512	7F00					
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Rango de sobreimpulso
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Rango nominal
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V	
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V	
-27649	93FF					Rango de subimpulso
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	
-32513	80FF					Rebase por defecto
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

[1] 7FFF puede ser devuelto por una de las siguiente razones: desbordamiento (tal como se indica en la tabla), antes de que los valores válidos estén disponibles (p. ej. justo después de un arranque) o si se detecta una rotura de hilo.

Tabla A-164 Representación de entradas analógicas de intensidad (SB y SM)

Sistema		**Rango de medida de intensidad**		
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	**De 4 mA a 20 mA**	
32767	7FFF	> 23,52 mA	> 22,81 mA	Rebase por exceso
32511	7EFF	23,52 mA	22,81 mA	Rango de sobreimpulso
27649	6C01			
27648	6C00	20 mA	20 mA	Rango nominal
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Rango de subimpulso
-4864	ED00	-3,52 mA	1,185 mA	
32767[1]	7FFF		< 1,185 mA	Rotura de hilo (4 a 20 mA)
-32768	8000	< -3,52 mA		Rebase por defecto (0 a 20 mA)

[1] El valor de rotura de hilo de 32767 (16#7FFF) se devuelve siempre, independientemente del estado de la alarma de rotura de hilo.

Consulte también

Determinar el tipo de condición de rotura de hilo en un módulo SM 1231 (Página 1236)

A.10.7 Rangos de medida de salidas analógicas de tensión e intensidad (SB y SM)

Tabla A-165 Representación de salidas analógicas para tensión (SB y SM)

Sistema		Rango de salida de tensión	
Decimal	**Hexadecimal**	**±10 V**	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	
32511	7EFF	11,76 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
1	1	361,7 µ V	
0	0	0 V	
-1	FFFF	-361,7 µ V	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Rango de subimpulso
-32512	8100	-11,76 V	
-32513	80FF	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	

[1] En condición de rebase por exceso o defecto, las salidas analógicas adoptarán el valor sustitutivo del estado STOP.

Tabla A-166 Representación de salidas analógicas para intensidad (SB y SM)

Sistema		Rango de salida de intensidad		
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	**De 4 mA a 20 mA**	
32767	7FFF	V. nota 1	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	V. nota 1	
32511	7EFF	23,52 mA	22,81 mA	Rango de sobreimpulso
27649	6C01			
27648	6C00	20 mA	20 mA	Rango nominal
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4mA	
-1	FFFF		De 4 mA a 578,7 nA	Rango de subimpulso
-6912	E500		0 mA	
-6913	E4FF			No es posible. Valor de salida limitado a 0 mA.
-32512	8100			
-32513	80FF	V. nota 1	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	V. nota 1	

[1] En condición de rebase por exceso o defecto, las salidas analógicas adoptarán el valor sustitutivo del estado STOP.

A.11 Módulos de señales RTD y de termopar (SMs)

A.11.1 SM 1231 de termopar

Tabla A-167 Especificaciones generales

Modelo	**SM 1231 AI 4 x 16 bits TC**	**SM 1231 AI 8 x 16 bits TC**
Referencia	6ES7231-5QD32-0XB0	6ES7231-5QF32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	
Peso	180 gramos	190 gramos
Disipación de potencia	1,5 W	
Consumo de corriente (bus SM)	80 mA	
Consumo de corriente (24 V DC) [1]	40 mA	

[1] De 20,4 a 28,8 V DC (clase 2, potencia limitada o alimentación de sensor por PLC)

Tabla A-168 Entradas analógicas

Modelo		**SM 1231 AI 4 x 16 bits TC**	**SM 1231 AI 8 x 16 bits TC**
Número de entradas		4	8
Rango Rango nominal (palabra de datos) Rango de saturación máximo/mínimo (palabra de datos) Rebase por exceso/defecto (palabra de datos)		Véase la tabla de selección de termopares (Página 1357).	
Resolución	Temperatura	0,1 °C/0,1 °F	
	Tensión	Signo más (+) de 15 bits	
Tensión soportada máxima		± 35 V	
Supresión de ruido		85 dB para el ajuste de filtro seleccionado (10 Hz, 50 Hz, 60 Hz o 400 Hz)	
Rechazo en modo común		> 120 dB a 120 V AC	
Impedancia		≥ 10 MΩ	
Aislamiento	De campo a lógica	707 V DC (ensayo de tipo)	
	De campo a 24 V DC	707 V DC (ensayo de tipo)	
	De 24 V DC a lógica	707 V DC (ensayo de tipo)	
Entre canales		120 V AC	
Precisión		Véase la tabla de selección de termopares (Página 1357).	
Repetitividad		±0,05% FS	
Principio de medición		Integrador	
Tiempo de actualización del módulo		Véase la tabla de selección de reducción de ruido (Página 1357).	
Error de unión fría		±1,5 °C	
Longitud de cable (metros)		100 metros hasta el sensor (máx.)	
Resistencia del cable		100 Ω máx.	

Tabla A-169 Diagnóstico

Modelo	SM 1231 AI 4 x 16 bits TC	SM 1231 AI 8 x 16 bits TC
Rebase por exceso/defecto[1]	Sí	
Rotura de hilo [2, 3]	Sí	
24 V DC baja tensión [1]	Sí	

[1] La información de las alarmas de diagnóstico de baja tensión y de rebase por exceso/defecto se indicará en los valores de datos analógicos aunque las alarmas estén deshabilitadas en la configuración del módulo.

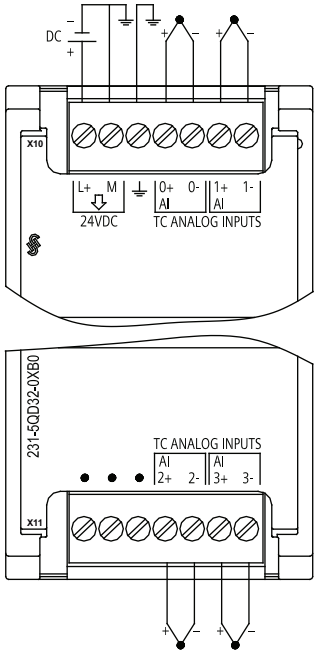
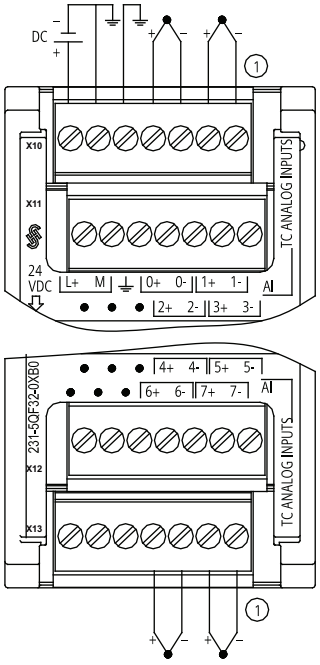
[2] Si la alarma de rotura de hilo está deshabilitada y se presenta una condición de rotura de hilo en la línea del sensor, el módulo puede señalizar valores aleatorios.

[3] El módulo comprueba cada 6 segundos si hay una rotura de hilo, con lo que el tiempo de actualización se prolonga en 9 ms para cada canal habilitado una vez cada 6 segundos.

El módulo de señales analógicas de termopar SM 1231 (TC) mide el valor de la tensión conectada a las entradas del módulo. El tipo de medición de temperatura puede ser "termopar" o "tensión".

- "Termopar": el valor se expresará en grados multiplicados por diez (p. ej. 25,3 grados se expresarán como 253 decimales).
- "Tensión": el valor máximo del rango nominal serán 27648 decimales.

Tabla A-170 Diagramas de cableado de los SM de termopar

SM 1231 AI 4 x TC 16 bits (6ES7231-5QD32-0XB0)	SM 1231 AI 8 x TC bits (6ES7231-5QF32-0XB0)
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

① Para mejorar la claridad, TC 2, 3, 4 y 5 no se muestran conectados.

Tabla A-171 Asignación de pines de conectores para SM 1231 AI 4 x TC 16 bits (6ES7231-5QD32-0XB0)

Pin	X10 (oro)	X11 (oro)
1	L+ / 24 V DC	Sin conexión
2	M / 24 V DC	Sin conexión
3	Tierra funcional	Sin conexión
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

Tabla A-172 Asignación de pines de conectores para SM 1231 AI 8 x TC bits (6ES7231-5QF32-0XB0)

Pin	X10 (oro)	X11 (oro)	X12 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión	Sin conexión
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

Nota

Las entradas analógicas que no se utilicen deben cortocircuitarse.

Los canales de termopar que no se utilizan pueden desactivarse. No se producirá ningún error si se desactiva un canal no utilizado.

A.11.1.1 Funcionamiento básico de un termopar

Los termopares se forman por la unión de dos metales diferentes que se conectan eléctricamente produciendo una tensión. La tensión generada es proporcional a la temperatura de la unión. Se trata de una tensión pequeña; un microvoltio puede representar varios grados. La medición de temperatura con termopares consiste en medir la tensión de un termopar, compensar las uniones adicionales y linealizar posteriormente el resultado.

Cuando se conecta un termopar al módulo de termopar SM 1231, los dos hilos de metales distintos se unen al conector de señales del módulo. El punto en el que los dos hilos diferentes se unen el uno con el otro constituye el termopar del sensor.

Dos termopares adicionales se forman donde los dos hilos diferentes se unen al conector de señales. La temperatura del conector genera una tensión que se suma a la del termopar del sensor. Si no se corrige esta tensión, la temperatura indicada será diferente de la temperatura del sensor.

La compensación de unión fría se utiliza para compensar el termopar del conector. Las tablas de termopares se basan en una temperatura de referencia que, por lo general, es de cero grados centígrados. La compensación de unión fría compensa el conector a cero grados centígrados. La

compensación restablece la tensión sumada por los termopares del conector. La temperatura del módulo se mide internamente y se convierte luego a un valor que se suma a la conversión del sensor. La conversión del sensor corregida se linealiza entonces utilizando las tablas de termopares.

Para optimizar el funcionamiento de la compensación de unión fría es necesario colocar el módulo de termopar en un entorno térmicamente estable. Una variación lenta (inferior a 0,1 °C/ minuto) del módulo a temperatura ambiente se compensa correctamente dentro de las especificaciones del módulo. Si hay corriente de aire a través del módulo también se producirán errores de compensación de unión fría.

Si se requiere una mejor compensación del error de unión fría, se puede utilizar un bloque de terminales isotérmico. El módulo de termopar permite utilizar un bloque de terminales con una referencia de 0 °C o 50 °C.

A.11.1.2 Tablas de selección de termopares para el SM 1231

Los rangos y la exactitud de los diferentes tipos de termopares soportados por el módulo de señales de termopar SM 1231 se indican en la tabla siguiente.

Tabla A-173 Tabla de selección de termopares

Tipo	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión en rango normal[3, 4] a 25 °C	Precisión en rango normal[1, 2, 6] de -20 °C a 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

Tipo	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión en rango normal[3, 4] a 25 °C	Precisión en rango normal[1, 2, 6] de -20 °C a 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
Tensión	-32512	-27648 -80mV	27648 80mV	32511	±0,05%	±0,1%

1 Los valores de termopar inferiores al valor del rango de saturación mínimo se notifican como -32768.

2 Los valores de termopar por encima del valor máximo del rango superior se notifican como 32767.

3 El error de unión fría interna es de ±1,5 °C para todos los rangos. Este debe agregarse a los valores de error de la tabla. El módulo requiere un tiempo mínimo de calentamiento de 30 minutos para cumplir esta especificación. Para temperaturas ambiente en el entorno del módulo inferiores a -10 °C, el error de unión fría interna puede ser superior a 1,5 °C.

4 En presencia de radiofrecuencia radiada de 970 MHz a 990 MHz, la precisión del SM 1231 AI 4 x 16 bits TC puede verse reducida.

5 Límite inferior de 3276,6 con indicación °F

6 El error de compensación de unión fría interna no se ha caracterizado para temperaturas ambiente del módulo inferiores a 0 °C y puede que exceda el valor especificado.

Nota

Canal de termopar

Cada canal del módulo de señales del termopar puede configurarse con un tipo de termopar diferente (seleccionable en el software durante la configuración del módulo).

Tabla A-174 Reducción de ruido y tiempos de actualización para el SM 1231 termopar

Selección de supresión de frecuencias	Tiempo de integración	Tiempo de actualización de módulo de 4 canales (segundos)	Tiempo de actualización de módulo de 8 canales (segundos)
400 Hz (2,5 ms)	10 ms[1]	0,143	0,285
60 Hz (16,6 ms)	16,67 ms	0,223	0,445
50 Hz (20 ms)	20 ms	0,263	0,525
10 Hz (100 ms)	100 ms	1,225	2,450

1 Para mantener la resolución y precisión del módulo con supresión de 400 Hz seleccionada, el tiempo de integración es de 10 ms. Esta selección también suprime el ruido de 100 Hz y 200 Hz.

Para medir termopares se recomienda utilizar un tiempo de integración de 100 ms. El uso de tiempos de integración inferiores aumentará el error de repetibilidad de las lecturas de temperatura.

Nota

Después de aplicar tensión al módulo, este lleva a cabo una calibración interna del conversor de analógico a digital. Durante este tiempo, el módulo indica un valor de 32767 en cada canal hasta que haya información válida disponible en el canal respectivo. Es posible que el programa de usuario deba autorizar este tiempo de inicialización. Dado que la configuración del módulo puede variar la duración del tiempo de inicialización, conviene verificar el comportamiento o el módulo en la configuración. Si es necesario, se puede incluir lógica en el programa de usuario para adaptarse al tiempo de inicialización del módulo.
Esta lógica puede implementarse utilizando una lectura de sondeo en el "OB de arranque", lo que bloquea la operación hasta que la inicialización se haya completado. Es necesario implementar la lectura de sondeo con un acceso inmediato. Si el valor de la lectura de sondeo de termopar es 32767, habrá que repetir la lectura hasta que el valor cambie. Para cada módulo el sondeo solo tiene que ejecutarse para el punto de entrada con el número más alto utilizado (las entradas del módulo se inicializan en orden de 0 a 7).

Representación de los valores analógicos de termopar tipo J

En la tabla siguiente se muestra una representación de los valores analógicos de termopares tipo J.

Tabla A-175 Representación de los valores analógicos de termopares del tipo J

Tipo J en °C	Unidades		Tipo J en °F	Unidades		Rango
	Decimal	Hexadecimal		Decimal	Hexadecimal	
> 1450,0	32767	7FFF	> 2642,0	32767	7FFF	Desborda-miento
1450,0 : 1200,1	14500 : 12001	38A4 : 2EE1	2642,0 : 2192,2	26420 : 21922	6734 : 55A2	Rango de saturación por exceso
1200,0 : -150	12000 : 1500	2EE0 : FA24	2192,0 : -238,0	21920 : -2380	55A0 : F6B4	Rango nominal
-150,1 : -210	-1501 : -2100	FA23 : F7CC	-238,1 : -346,0	-2381 : -3460	F6B3 : F27C	Rango de saturación por defecto
< -210,0	-32768	8000	< -346,0	-32768	8000	Rebase por defecto[1]

[1] Un cableado defectuoso (por ejemplo, inversión de polaridad o entradas abiertas) o error del sensor en el rango negativo (por ejemplo, tipo erróneo de termopar) pueden provocar que el módulo de termopar señale un rebase por defecto.

A.11.2 SM 1231 RTD

Datos técnicos del SM 1231 RTD

Tabla A-176 Especificaciones generales

Datos técnicos	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x 16 bits
Referencia	6ES7231-5PD32-0XB0	6ES7231-5PF32-0XB0
Dimensiones A x A x P (mm)	45 x 100 x 75	70 x 100 x 75
Peso	220 gramos	270 gramos
Disipación de potencia	1,5 W	
Consumo de corriente (bus SM)	80 mA	90 mA
Consumo de corriente (24 V DC) [1]	40 mA	

[1] De 20,4 a 28,8 V DC (clase 2, potencia limitada o alimentación de sensor por CPU)

Tabla A-177 Entradas analógicas

Datos técnicos		SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x16 bits
Número de entradas		4	8
Tipo		RTD y Ω indicado por el módulo	
Rango Rango nominal (palabra de datos) Rango de sobreimpulso/subimpulso (palabra de datos) Rebase por exceso/defecto (palabra de datos)		Véase la tabla de selección de sensores RTD (Página 1363).	
Resolución	Temperatura	0,1 °C/0,1 °F	
	Resistencia	Signo más (+) de 15 bits	
Tensión soportada máxima		± 35 V	
Supresión de ruido		85 dB para la reducción de ruido seleccionada (10 Hz, 50 Hz, 60 Hz o 400 Hz)	
Rechazo en modo común		> 120dB	
Impedancia		≥ 10 MΩ	
Aislamiento	De campo a lógica	707 V DC (ensayo de tipo)	
	De campo a 24 V DC	707 V DC (ensayo de tipo)	
	De 24 V DC a lógica	707 V DC (ensayo de tipo)	
Aislamiento entre canales		Ninguno	
Precisión		Véase la tabla de selección de sensores RTD (Página 1363).	
Repetitividad		±0,05% FS	
Disipación máxima del sensor		0,5 mW	
Principio de medición		Integrador	
Tiempo de actualización del módulo		Véase la tabla de selección de reducción de ruido (Página 1363).	
Longitud de cable (metros)		100 metros hasta el sensor (máx.)	
Resistencia del cable		20 Ω, 2,7 Ω para 10 Ω RTD máx.	

Tabla A-178 Diagnóstico

Datos técnicos	SM 1231 AI 4 x RTD x 16 bits	SM 1231 AI 8 x RTD x16 bits
Rebase por exceso/defecto[1,2]	Sí	
Rotura de hilo [3]	Sí	
24 V DC baja tensión [1]	Sí	

[1] La información de las alarmas de diagnóstico de baja tensión y de rebase por exceso/defecto se indicará en los valores de datos analógicos aunque las alarmas estén deshabilitadas en la configuración del módulo.

[2] La detección de rebase por defecto nunca está habilitada para los rangos de resistencia.

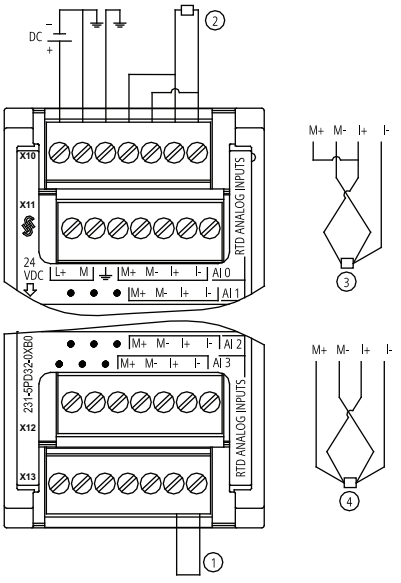
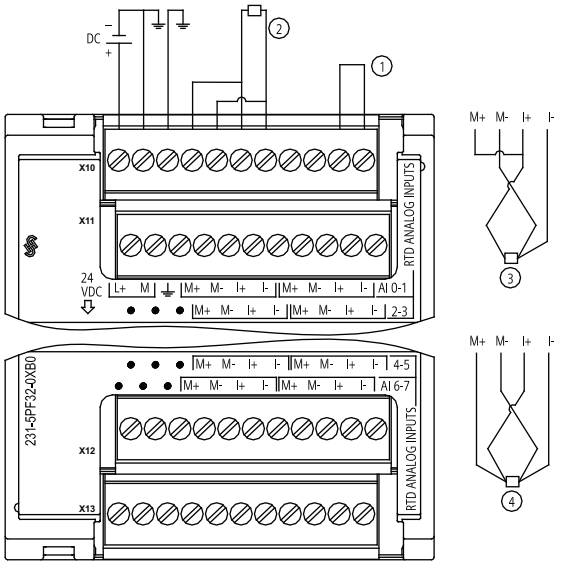
[3] Si la alarma de rotura de hilo está deshabilitada y se presenta una condición de rotura de hilo en la línea del sensor, el módulo puede señalizar valores aleatorios.

El módulo de señales analógico SM 1231 RTD mide el valor de la resistencia conectada a las entradas del módulo. El tipo de medición puede elegirse entre "resistor" y "resistor térmico".

- "Resistor": el valor máximo del rango nominal serán 27648 decimales.
- "Resistor térmico": el valor se expresará en grados multiplicados por diez (p. ej. 25,3 grados se expresarán como 253 decimales). Los valores del rango climático se expresarán en grados multiplicados por cien (p. ej. 25,34 grados se expresarán como 2534 decimales).

El módulo SM 1231 RTD soporta mediciones con base en conexiones a 2, 3 y 4 hilos que van al sensor de resistencia.

Tabla A-179 Diagramas de cableado de SM RTD

SM 1231 RTD 4 x 16 bits (6ES7231-5PD32-0XB0)	SM 1231 RTD 8 x 16 bits (6ES7231-5PF32-0XB0)
	

① Entradas de bucle RTD no utilizadas
② RTD a 2 hilos ③ RTD a 3 hilos ④ RTD a 4 hilos

NOTA: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.

Tabla A-180 Asignación de pines de conectores para SM 1231 RTD 4 x 16 bits (6ES7231-5PD32-0XB0)

Pin	X10 (oro)	X11 (oro)	X12 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión	Sin conexión
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

Tabla A-181 Asignación de pines de conectores para SM 1231 RTD 8 x 16 bits (6ES7231-5PF32-0XB0)

Pin	X10 (oro)	X11 (oro)	X12 (oro)	X13 (oro)
1	L+ / 24 V DC	Sin conexión	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión	Sin conexión
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

Nota

Los canales RTD que no se utilizan pueden desactivarse. No se producirá ningún error si se desactiva un canal no utilizado.

El módulo RTD necesita un bucle de corriente continuo para eliminar el tiempo de estabilización suplementario que se agrega automáticamente a un canal no utilizado que no se ha desactivado. El módulo RTD debe tener conectado un resistor para fines de coherencia (como la conexión RTD a 2 hilos).

A.11.2.1 Tablas de selección para el SM 1231 RTD

Tabla A-182 Rangos y precisión de los diferentes sensores soportados por los módulos RTD

Coeficiente de temperatura	Tipo de RTD	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión de rango normal a 25 °C	Precisión de rango normal entre -20 °C y 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 climatiz.	-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						
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] Los valores de RTD inferiores al valor del rango de saturación mínimo se notifican como -32768.

[2] Los valores de RTD superiores al valor del rango de saturación máximo se notifican como +32767.

Tabla A-183 Resistencia

Rango	Rango de saturación mínimo	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[1]	Precisión de rango normal a 25 °C	Precisión de rango normal entre -20 °C y 60 °C
150 Ω	n/a	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0,05 %	±0,1 %
300 Ω	n/a	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0,05 %	±0,1 %
600 Ω	n/a	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0,05 %	±0,1 %

[1] Los valores de resistencia superiores al valor del rango de saturación máximo se notifican como +32767.

Nota

El módulo reporta 32767 en todo canal que esté activado y que no tenga conectado ningún sensor. Si la detección de rotura de hilo también está habilitada, en el módulo se encenderán de forma intermitente los LEDs rojos correspondientes.

En caso de utilizar rangos de 500 Ω y 1000 Ω del RTD con otros resistores de menor valor, el error puede aumentar al doble del error especificado.

La exactitud más elevada se alcanza para los rangos de 10 Ω del RTD con conexiones a 4 hilos.

La resistencia de los hilos en el modo a 2 hilos dará lugar a un error de lectura en el sensor y por ello no se garantiza la exactitud.

Tabla A-184 Reducción de ruido y tiempos de actualización para los módulos RTD

Selección de supresión de frecuencias	Tiempo de integración	Tiempo de actualización (segundos)	
		Módulo de 4 canales	Módulo de 8 canales
400 Hz (2,5 ms)	10 ms[1]	4/2 hilos: 0,142 3 hilos: 0,285	4/2 hilos: 0,285 3 hilos: 0,525
60 Hz (16,6 ms)	16,67 ms	4/2 hilos: 0,222 3 hilos: 0,445	4/2 hilos: 0,445 3 hilos: 0,845
50 Hz (20 ms)	20 ms	4/2 hilos: 0,262 3 hilos: 0.505	4/2 hilos: 0,524 3 hilos: 1,015
10 Hz (100 ms)	100 ms	4/2 hilos: 1,222 3 hilos: 2,445	4/2 hilos: 2,425 3 hilos: 4,845

[1] Para mantener la resolución y precisión del módulo con el filtro 400 Hz seleccionado, el tiempo de integración es de 10 ms. Esta selección también suprime el ruido de 100 Hz y 200 Hz.

Nota

Después de aplicar tensión al módulo, este lleva a cabo una calibración interna del conversor de analógico a digital. Durante este tiempo, el módulo indica un valor de 32767 en cada canal hasta que haya información válida disponible en el canal respectivo. Es posible que el programa de usuario deba autorizar este tiempo de inicialización. Como la configuración del módulo puede modificar la longitud del tiempo de inicialización, es conveniente verificar el comportamiento o el módulo en la configuración. Si es necesario, se puede incluir lógica en el programa de usuario para adaptarse al tiempo de inicialización del módulo.
Esta lógica puede implementarse utilizando una lectura de sondeo en el "OB de arranque", lo que bloquea la operación hasta que la inicialización se haya completado. Es necesario implementar la lectura de sondeo con un acceso inmediato. Si el valor de la lectura de sondeo de RTD es 32767, habrá que repetir la lectura hasta que el valor cambie. Para cada módulo el sondeo solo tiene que ejecutarse para el punto de entrada con el número más alto utilizado (las entradas del módulo se inicializan en orden de 0 a 7).

Representación de valores analógicos para RTD

En las tablas siguientes se muestra una representación del valor medido digitalizado para los sensores de rango de temperatura estándar RTD.

Tabla A-185 Representación de los valores analógicos para termorresistencias PT 100, 200, 500, 1000 y PT 10, 50, 100, 500 GOST (0,003850) estándar

Pt x00 estándar en °C (1 dígito = 0,1 °C)	Unidades		Pt x00 estándar en °F (1 dígito = 0,1 °F)	Unidades		Rango
	Decimal	Hexadecimal		Decimal	Hexadecimal	
> 1000,0	32767	7FFF	> 1832,0	32767	7FFF	Desbordamiento
1000,0 : 850,1	10000 : 8501	2710 : 2135	1832,0 : 1562,1	18320 : 15621	4790 : 3D05	Sobre rango
850,0 : -200,0	8500 : -2000	2134 : F830	1562,0 : -328,0	15620 : -3280	3D04 : F330	Rango nominal
-200,1 : -243,0	-2001 : -2430	F82F : F682	-328,1 : -405,4	-3281 : -4054	F32F : F02A	Bajo rango
< -243,0	-32768	8000	< -405,4	-32768	8000	Rebase por defecto

A.12 Módulos tecnológicos

A.12.1 Maestro SM 1278 4xIO-Link

Tabla A-186 Especificaciones generales

Datos técnicos		**Módulo de señales SM 1278 4xIO-Link Master**
Referencia		6ES7278-4BD32-0XB0
Dimensiones An. x Al. x P. (mm)		45 x 100 x 75
Peso		150 gramos
Información general		
	Datos I&M	Sí; de IM0 a IM3
Tensión de alimentación		
	Tensión nominal (DC)	24 V DC
	Límite inferior del rango válido (DC)	19,2 V; 20,5 V si se utiliza IO-Link (la tensión de alimentación para los dispositivos IO-Link en el maestro debe ser como mínimo de 20 V)
	Límite superior del rango válido (DC)	28,8 V DC
	Protección contra inversión de polaridad	Sí
Intensidad de entrada		
	Consumo de corriente	65 mA; sin carga
Alimentación de encoders		
	Número de salidas	4
	Intensidad de salida, valor nominal	200 mA por canal
Disipación		
	Disipación, típ.	1 W, excluida carga de puerto
Entradas/salidas digitales		
	Longitud de cable (metros)	20 m, sin apantallar, máx.
SDLC		
	Longitud de cable (metros)	20 m, sin apantallar, máx.
IO-Link		
	Número de puertos	4
	Número de puertos que pueden controlarse simultánea-mente	4
	Protocolo IO-Link 1.0	Sí
	Protocolo IO-Link 1.1	Sí
Estado operativo		
	IO-Link	Sí
	DI	Sí
	DQ	Sí; máx. 100 mA
Conexión de dispositivos IO-Link		

Datos técnicos			**Módulo de señales SM 1278 4xIO-Link Master**
	Puerto tipo A		Sí
		Velocidad de transferencia	4,8 kBd (COM1)
			38,4 kBd (COM2)
			230,4 kBd (COM3)
	Tiempo de ciclo, mín.		2 ms; dinámico, depende de la longitud de los datos de usuario
	Tamaño de datos de proceso, entrada por puerto		32 bytes; máx.
	Tamaño de datos de proceso, entrada por módulo		32 bytes
	Tamaño de datos de proceso, salida por puerto		32 bytes; máx.
	Tamaño de datos de proceso, salida por módulo		32 bytes
	Memoria para parámetros de dispositivo		2 KB
	Longitud del cable sin apantallar, máx. (metros)		20 m
Información de alarmas/diagnóstico/estado			
	Indicador de estado		Sí
Alarmas			
	Alarma de diagnóstico		Sí; el diagnóstico de puerto solo está disponible en el modo IO-Link
Alarmas de diagnóstico			
	Diagnóstico		
		Vigilancia de la tensión de alimentación	Sí
		Cortocircuito	Sí
LED indicador de diagnóstico			
	Vigilancia de la tensión de alimentación		Sí; LED DIAG intermitente en rojo
	Indicador de estado de canal		Sí; un LED verde por canal para estado de canal Qn (modo SIO) y estado de PORT Cn (modo IO-Link)
	Para diagnóstico de canal		Sí; LED Fn rojo
	Para diagnóstico de módulo		Sí; LED DIAG verde/rojo
Aislamiento galvánico			
	Aislamiento galvánico canales		
		Entre canales	No
		Entre los canales y el bus de fondo	Sí
Aislamiento			
	Aislamiento ensayado con		707 V AC (ensayo de tipo)
Condiciones ambientales			
	Temperatura de servicio		
		Mín.	-20 °C
		Máx.	60 °C
		Montaje horizontal, mín.	-20 °C
		Montaje horizontal, máx.	60 °C
		Montaje vertical, mín.	-20 °C
		Montaje vertical, máx.	50 °C

Sinopsis del tiempo de respuesta

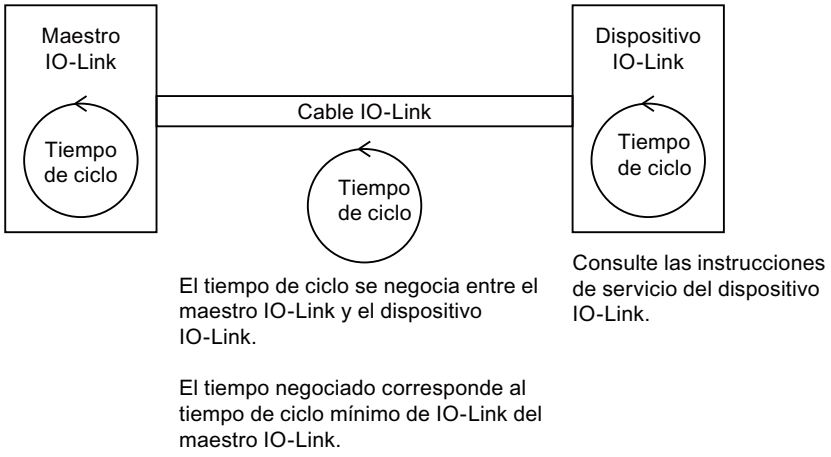


Tabla A-187 Diagrama de cableado del maestro SM 1278 IO-Link

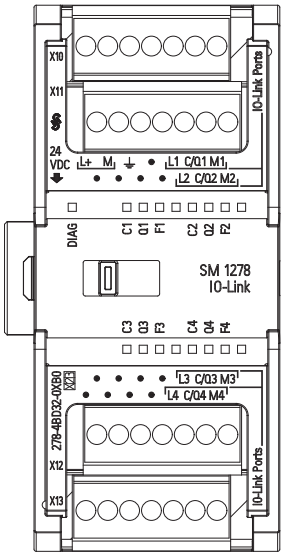
SM 1278 maestro IO-Link (6ES7278-4BD32-0XB0)	
	

Tabla A-188 Asignación de pines de conectores para SM 1278 maestro IO-Link (6ES7278-4BD32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Sin conexión	Sin conexión	Sin conexión
2	M / 24 V DC	Sin conexión	Sin conexión	Sin conexión
3	Tierra funcional	Sin conexión	Sin conexión	Sin conexión
4	Sin conexión	Sin conexión	Sin conexión	Sin conexión
5	L_1	L_2	L_3	L_4
6	C/Q_1	C/Q_2	C/Q_3	C/Q_4
7	M_1	M_2	M_3	M_4

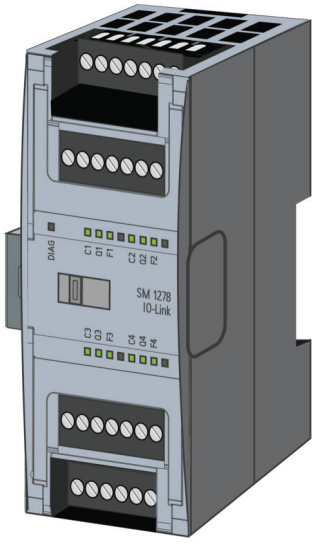
A.12.1.1 Vista general del maestro SM 1278 4xIO-Link

El maestro SM 1278 4xIO-Link es un módulo de 4 puertos que funciona como módulo de señales y módulo de comunicación. Cada puerto puede funcionar en el modo IO-Link, entrada digital de 24 V DC o salida digital de 24 V DC únicas.

El maestro IO-Link programa la comunicación acíclica con un dispositivo IO-Link mediante el bloque de función IO_LINK_DEVICE (FB) en el programa STEP 7 del controlador S7-1200. El FB IO_LINK_DEVICE indica qué maestro IO-Link utiliza el programa y qué puertos utiliza el maestro para el intercambio de datos.

Visite el sitio web Siemens Industry Online Support (https://support.industry.siemens.com/cs/ww/es/) para conocer más detalles sobre cómo trabajar con la librería IO-Link . Introduzca "IO-Link" en el campo de búsqueda de la web para acceder a información sobre los productos IO-Link y su uso.

Vista del módulo



Propiedades

Propiedades técnicas

- El maestro IO-Link según la especificación de IO-Link V1.1 (consulte el sitio web de IO-Link Consortium (http://io-link.com/en/index.php) para obtener más información).
- Módulo de comunicación serie con cuatro puertos (canales)
- Velocidad de transferencia de datos COM1 (4.8 kbaud), COM2 (38.4 kbaud), COM3 (230.4 kbaud)
- Modo SIO (modo IO estándar)
- Conexión de hasta cuatro dispositivos IO-Link (conexión a 3 hilos) o cuatro actuadores estándar o encóders estándar
- Función de diagnóstico programable por puerto

Funciones admitidas

- Datos de identificación de instalación y mantenimiento
- Actualización de firmware
- Parametrización de IO-Link mediante la herramienta de configuración de puertos S7-PCT, STEP 7 Professional y una CPU. S7-1200 V4.0 o posterior. En STEP 7 Professional, V15 (utilizando V2.1 HSP o posterior), la parametrización de IO-Link puede realizarse utilizando el TIA Portal con funcionalidad limitada.
- Bits Port Qualifier Information (PQI)
- Copia de seguridad y restauración con FB de librería IO-Link

IO-Link es una conexión punto a punto entre un maestro y un dispositivo. Los actuadores/sensores convencionales e inteligentes se pueden usar como dispositivos en el IO-Link con cables estándar no apantallados mediante tecnología de 3 hilos contrastada. IO-Link es compatible de forma retroactiva con actuadores y sensores digitales convencionales anteriores. El estado del circuito y el canal de datos están diseñados en la acreditada tecnología de 24 V DC.

Encontrará información adicional sobre la tecnología SIMATIC-IO-Link en el "manual de funciones del sistema IO-Link", en la página web de Siemens Industry Online Support (http://support.automation.siemens.com).

Nota

Datos de parámetros de IO-Link

Cuando sustituya el maestro SM 4xIO-Link, los datos de parámetros no se le asignarán automáticamente.

⚠ PRECAUCIÓN

Extracción e inserción

Si inserta el maestro SM 4xIO-Link con la carga conectada, se pueden generar situaciones peligrosas en la instalación.

A consecuencia de ello, se pueden producir daños físicos en el sistema de automatización de S7-1200.

Extraiga o inserte el maestro SM 4xIO-Link solo cuando la carga esté apagada.

Efectos del restablecimiento a la configuración de fábrica

Use la función "Restablecer configuración de fábrica" para restaurar las asignaciones de parámetros que ha realizado con S7-PCT al estado de suministro.

Después de aplicar "Restablecer configuración de fábrica", los parámetros del módulo SM 1278 4xIO-Link se asignan del siguiente modo:

- Los puertos están en modo DI.
- Los puertos se mapean en las direcciones relativas de 0.0 a 0.3.
- El PortQualifier está deshabilitado.
- Los datos de mantenimiento de 1 a 3 se borran.

Nota

Cuando restablezca la configuración de fábrica, se borrarán los parámetros de dispositivo y se restaurará el estado de suministro.

Si extrae un módulo de señales SM 1278 4xIO-Link, restablézcalo a la configuración de fábrica antes de almacenarlo.

Procedimiento

Para aplicar "Restablecer configuración de fábrica", proceda como se describe en la ayuda online de S7-PCT en "Master Configuration > 'Commands" (Configuración maestra > ficha Comandos).

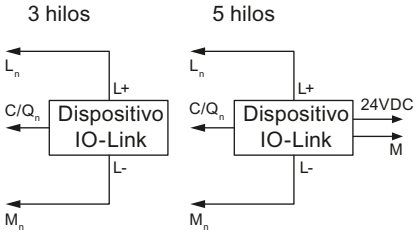
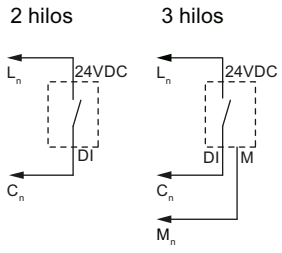
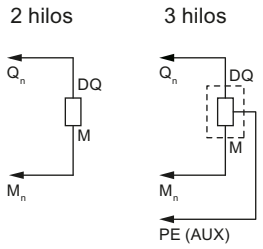
A.12.1.2 Conexión

Para consultar más detalles sobre la asignación de pines véase la tabla Asignación de pines de conectores para SM 1278 I/O-Link Master (6ES 278-4BD32-0XB0). (Página 1366)

En la siguiente tabla se muestran las asignaciones de los bornes para el maestro SM 1278 4xIO-Link:

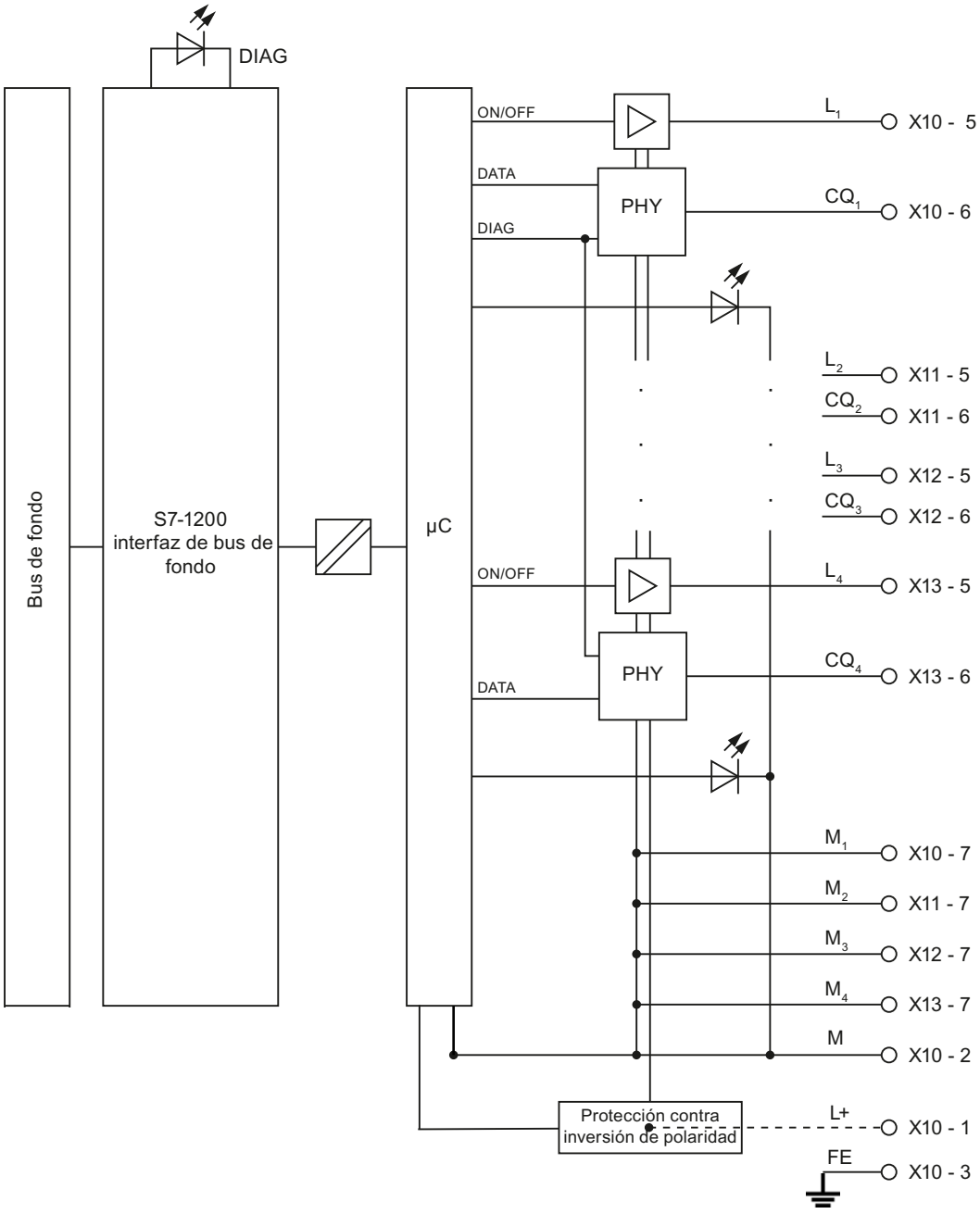
Pin	X10	X11	X12	X13	Notas	BaseUnits
7	M_1	M_2	M_3	M_4	• M_n: tierra a esclavo • C/Q_n: SDLC, DI o DQ • L_n: De 24 V DC a esclavo • M: tierra • L+: De 24 V DC a maestro • RES: reservado; puede no estar asignado	A1
6	C/Q_1	C/Q_2	C/Q_3	C/Q_4		
5	L_1	L_2	L_3	L_4		
4	RES	RES	RES	RES		
3	⏚ (tierra funcional)	RES	RES	RES		
2	M	RES	RES	RES		
1	L+	RES	RES	RES		

En la tabla siguiente se incluyen ilustraciones de ejemplos de conexión, en los que n = número de puerto:

Estado operativo IO-Link	Estado operativo DI	Estado operativo DQ
		

Nota

Los sensores conectados deben usar la alimentación de dispositivo proporcionada por la conexión L_n del módulo maestro


DIAG
Bus de fondo
S7-1200
interfaz de bus de
fondo
µC
ON/OFF
DATA
DIAG
PHY
L1
X10 - 5
CQ1
X10 - 6
L2
X11 - 5
CQ2
X11 - 6
L3
X12 - 5
CQ3
X12 - 6
ON/OFF
DATA
PHY
L4
X13 - 5
CQ4
X13 - 6
M1
X10 - 7
M2
X11 - 7
M3
X12 - 7
M4
X13 - 7
M
X10 - 2
Protección contra
inversión de polaridad
L+
X10 - 1
FE
X10 - 3

A.12.1.3 Parámetros/espacio de direcciones

Configuración del maestro SM 1278 4xIO-Link

Para la integración de módulos, la asignación de parámetros y la puesta en marcha se requiere STEP 7 V13 o posterior. Para algunas funciones también se requiere S7-PCT (Port Configuration Tool).

La tabla siguiente muestra las condiciones en las que se necesita S7-PCT:

	Maestro SM 1278 V2.0 4xIO-Link	**Maestro SM 1278 V2.1 4xIO-Link**
STEP7 V13.x y STEP V14.x	Se requiere S7-PCT	Se requiere S7-PCT
STEP 7 15.x	Se requiere S7-PCT	No se requiere S7-PCT para funciones básicas Se requiere S7-PCT para funciones avanzadas

Para más información, consulte el Manual de sistema SIMATIC IO-Link (https://support.industry.siemens.com/cs/ww/es/view/65949252).

En la siguiente tabla se muestran los parámetros para el maestro SM 1278 4xIO-Link:

Parámetros	**Intervalo de valores**	**Valor predeterminado**	**Configuración en RUN**	**Rango de eficiencia**
Puerto de diagnóstico 1	• Inhibir • Habilitar	Inhibir	Sí	Puerto (canal)
Puerto de diagnóstico 2	• Inhibir • Habilitar	Inhibir	Sí	Puerto (canal)
Puerto de diagnóstico 3	• Inhibir • Habilitar	Inhibir	Sí	Puerto (canal)
Puerto de diagnóstico 4	• Inhibir • Habilitar	Inhibir	Sí	Puerto (canal)

Habilitación de diagnósticos para el parámetro del puerto 1 al puerto 4

Este parámetro permite habilitar los diagnósticos para los puertos específicos de los cuatro puertos IO-Link.

Las asignaciones de puertos son los siguientes:

Puerto 1 → canal 1

Puerto 2 → canal 2

Puerto 3 → canal 3

Puerto 4 → canal 4

El tamaño máximo de las direcciones de entrada y salida de SM 4xIO-Link Master es de 32 bytes en cada caso. Los espacios de direcciones se asignan utilizando la herramienta de configuración de puertos S7-PCT, o, a partir de V15 V2.1 HSP o posterior, con la configuración hardware de TIA Portal.

Registro de parámetros

Asignación de parámetros en el programa de usuario

Puede configurar el dispositivo en tiempo de ejecución.

Cambio de parámetros en tiempo de ejecución

Los parámetros de módulo se incluyen en el registro 128. Puede transmitir los parámetros modificables en el módulo con la instrucción WRREC.

Cuando inicializa (conecta y desconecta) la CPU, la CPU sobrescribe los parámetros que se han enviado al módulo con la instrucción WRREC durante el proceso de parametrización.

Instrucción para la asignación de parámetros

La siguiente instrucción se proporciona para asignar parámetros al módulo IO del programa de usuario:

Instrucción	Aplicación
SFB 53 WRREC	Transfiera los parámetros modificables al módulo.

Mensaje de error

El siguiente valor de retorno se notifica en caso de error:

Código de error	Significado
$80B1_H$	Error de longitud de datos
$80E0_H$	Error de información de encabezado
$80E1_H$	Error de parametrización

Estructura del registro

En la siguiente tabla se muestran los parámetros de IO-Link:

Offset	Etiqueta	Tipo	Valor predeterminado	Descripción
0	Versión	1 byte	0x02	Muestra la estructura del registro 0x02 del maestro IO-Link conforme a IO-Link V1.1.
1	Longitud del parámetro	1 byte	0x02	Longitud del parámetro (2 bytes + 2 encabezamientos)
Parámetros de arranque IO-Link				
2	Diagnóstico de puerto (puerto1 1 a n)	1 byte	0x00	Activación del diagnóstico para el puerto 1 a n
3	Propiedades de IOL	1 byte	0x00	Propiedades de módulo

En la siguiente tabla se muestra la versión del registro:

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reservado		Versión principal (00)		Versión secundaria (0010)			

En la siguiente tabla se muestra el diagnóstico de puerto del registro:

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reservado				EN_Port4	EN_Port3	EN_Port2	EN_Port1

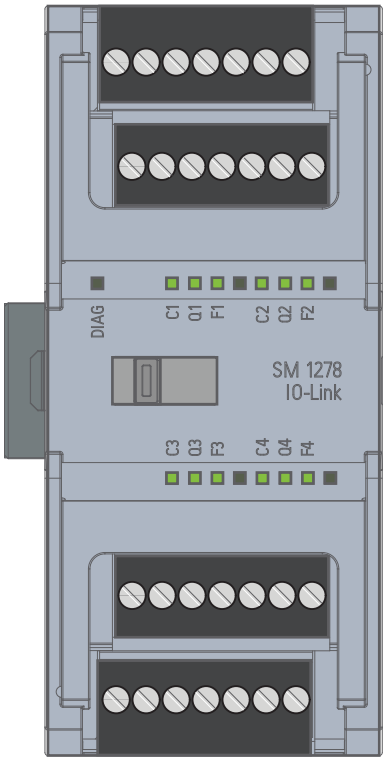
EN_Portx:
0 = Diagnóstico desactivado
1 = Diagnóstico activado

En la siguiente tabla se muestran las propiedades IOL del registro:

Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
Reservado							

A.12.1.4 Alarmas, alarmas de error y de sistema

Indicador LED



Significado de los indicadores LED

En la siguiente tabla se explica el significado de los indicadores de estado y error. En el apartado "Alarmas de diagnóstico" encontrará medidas correctoras para las alarmas de diagnóstico.

LED DIAG

DIAG	Significado
☐ Off	El suministro de bus de fondo de S7-1200 no es correcto.
☼ Parpadea	El módulo no se ha configurado.
■ On	Módulo parametrizado y sin diagnóstico de módulo
☼ Parpadea	El módulo parametrizado y sin diagnóstico de módulo. O L+ alimentación sin conectar

Estado de puerto de LED

Válido para el puerto IO-Link que se encuentra en el modo de puerto IO-Link.

COM/1 ... COM/4	Significado
□ Off	Puerto desactivado
☼ Parpadea	Puerto activado, dispositivo no conectado o el puerto no está conectado al dispositivo configurado.
■ On	Puerto activado, dispositivo conectado

Estado de canal de LED

Válido para el puerto IO-Link que se encuentra en modo DI/Q.

DI/Q1 ... DI/Q4	Significado
□ Off	Señal de proceso = 0
■ On	Señal de proceso = 1

LED de error de puerto

F1...F4	Significado
□ Off	No hay error
■ On	Error

Los errores de módulo se indican como diagnósticos (estados de módulo) solo en modo IO-Link.

Alarma de diagnóstico	Código de error (decimal)	ESTADO (W#16#...)	Significado (código de error de IO-Link)	Maestro IO-Link	Dispositivo IO-Link
Cortocircuito	1	1804	Cortocircuito en los cables de proceso del dispositivo IO-Link	X	
		7710	Cortocircuito en el dispositivo IO		X
Subtensión	2	5111 5112	Tensión de alimentación demasiado baja		X
Sobretensión	3	5110	Tensión de alimentación demasiado alta		X
Sobrecalentamiento	5	1805	Temperatura excesiva en el maestro	X	
		4000 4210	Temperatura excesiva en el dispositivo		X

Alarma de diagnóstico	Código de error (decimal)	ESTADO (W#16#...)	Significado (código de error de IO-Link)	Maestro IO-Link	Dispositivo IO-Link
Rotura de hilo	6	1800	• No hay ningún dispositivo IO-Link conectado. • Hay una rotura en la línea de señales para el dispositivo IO-Link. • El dispositivo IO-Link no se puede comunicar debido a un error distinto.	X	
Rebase por exceso	7	8C10 8C20	Rango de etiqueta de proceso excedido		X
		8C20	Rango de medición excedido		
Rebase por defecto	8	8C30	Rango de etiqueta de proceso demasiado bajo		X
Error	9	---	Todos los códigos de error de IO-Link que no estén enumerados aquí se mapean a este error de PROFIBUS DP.		X
Error de asignación de parámetro	16	1882 1883	No se ha podido configurar el maestro IO-Link.	X	
		1802	Dispositivo incorrecto		
		1886	Error de almacenamiento		
		6320 6321 6350	El dispositivo no se ha configurado correctamente.		X
Falta tensión de alimentación	17	1806	Falta L+ tensión de alimentación para el dispositivo.	X	
		1807	L+ tensión de alimentación para el dispositivo es insuficiente(<20 V).		
Fusible defectuoso	18	5101	El fusible en el dispositivo es defectuoso.		X
Desconexión de seguridad	25	1880	Error grave (se debe sustituir el maestro)	X	
Fallo externo	26	1809 180A 180B 180C 180D	Error de almacenamiento de datos	X	
		1808	Hay más de 6 errores pendientes a la vez en el dispositivo IO-Link.		

A.13 Signal Boards digitales (SBs)

A.13.1 Datos técnicos de la SB 1221 de entradas digitales 200 kHz

Tabla A-189 Especificaciones generales

Datos técnicos	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Referencia	6ES7221-3BD30-0XB0	6ES7221-3AD30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21	
Peso	35 gramos	
Disipación de potencia	1,5 W	1,0 W
Consumo de corriente (bus SM)	40 mA	
Consumo de corriente (24 V DC)	7 mA / entrada + 20 mA	15 mA / entrada + 15 mA

Tabla A-190 Entradas digitales

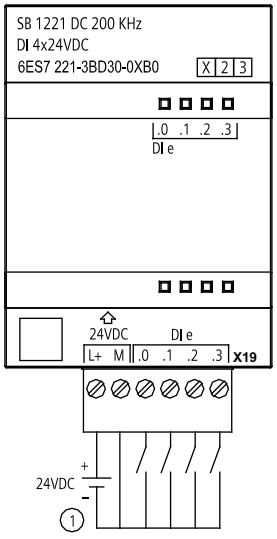
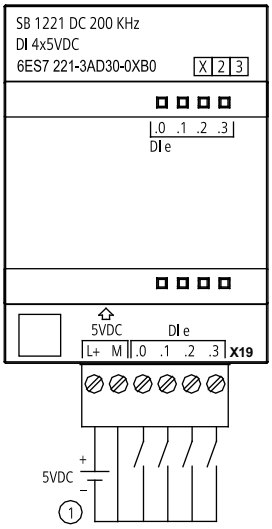
Datos técnicos		SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Número de entradas		4	
Tipo		Fuente	
Tensión nominal		24 V DC a 7 mA, nominal	5 V DC a 15 mA, nominal
Tensión continua admisible		28,8 V DC	6 V DC
Sobretensión transitoria		35 V DC durante 0,5 s	6 V
Señal 1 lógica		De 0 V (10 mA) a L+ menos 10 V (2,9 mA)	De 0 V (20 mA) a L+ menos 2,0 V (5,1 mA)
Señal 0 lógica		De L+ menos 5 V (1,4 mA) a L+ (0 mA)	De L+ menos 1,0 V (2,2 mA) a L + (0 mA)
Frecuencias de entrada de reloj HSC (máx.)		Fase simple: 200 kHz Fase en cuadratura: 160 kHz	
Aislamiento (de campo a lógica)		707 V DC (ensayo de tipo)	
Grupos de aislamiento		1	
Tiempos de filtro	Configuración en us	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
	Configuración en ms	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
Número de entradas ON simultáneamente		• 2 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 4 a 55 °C en horizontal o 45 °C en vertical	4
Longitud de cable (metros)		50, par trenzado apantallado	

Nota

En caso de conmutar frecuencias superiores a 20 kHz, es importante que las entradas digitales reciban una forma de onda cuadrada. Tenga en cuenta las siguientes posibilidades para mejorar la calidad de señal hacia las entradas:

- Minimizar la longitud del cable
- Cambiar el driver tipo sumidero NPN por uno tipo sumidero NPN y fuente PNP
- Cambiar el cable por uno de mayor calidad
- Reducir el circuito/los componentes de 24 V a 5 V
- Agregar una carga externa a la entrada

Tabla A-191 Diagramas de cableado de las SB de entradas digitales 200 kHz

SB 1221 DI 4 x 24 V DC, 200 kHz (6ES7221-3BD30-0XB0)	**SB 1221 DI 4 x 5 V DC, 200 kHz (6ES7221-3AD30-0XB0)**
	

① Admite únicamente entradas en fuente

Tabla A-192 Asignación de pines de conectores para SB 1221 DI 4 x 24 V DC, 200 kHz (6ES7221-3BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

Tabla A-193 Asignación de pines de conectores para SB 1221 DI 4 x 5 V DC, 200 kHz (6ES7221-3AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DI e.0
4	DI e.1
5	DI e.2
6	DI e.3

A.13.2 Datos técnicos de la SB 1222 de salidas digitales 200 kHz

Tabla A-194 Datos técnicos generales

Datos técnicos	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Referencia	6ES7222-1BD30-0XB0	6ES7222-1AD30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21	
Peso	35 gramos	
Disipación de potencia	0,5 W	
Consumo de corriente (bus SM)	35 mA	
Consumo de corriente (24 V DC)	15 mA	

Tabla A-195 Salidas digitales

Datos técnicos	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Número de salidas	4	
Tipo de salida	Estado sólido, MOSFET (sumidero y fuente)[1]	
Rango de tensión	De 20,4 a 28,8 V DC	De 4,25 a 6,0 V DC
Señal 1 lógica a intensidad máx.	L+ menos 1,5 V	L+ menos 0,7 V
Señal 0 lógica a intensidad máx.	1,0 V DC, máx.	0,2 V DC, máx.
Intensidad (máx.)	0,1 A	
Carga de lámparas	--	
Resistencia en estado ON (contactos)	11 Ω máx.	7 Ω máx.
Resistencia en estado desact.	6 Ω máx.	0,2 Ω máx.
Corriente de fuga por salida	--	
Frecuencia de tren de impulsos	200 kHz máx., 2 Hz mín.	
Extracorriente de conexión	0,11 A	
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)	
Grupos de aislamiento	1	
Intensidad por neutro	0,4 A	
Tensión de bloqueo inductiva	Ninguna	

Datos técnicos	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Retardo de conmutación	1,5 µs + 300 ns asc. 1,5 µs + 300 ns desc.	200 ns + 300 ns asc. 200 ns + 300 ns desc.
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)	
Control de una entrada digital	Sí	
Salidas paralelas para control de carga redundante	No	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	• 2 (no adyacentes) a 60 °C en horizontal o 50 °C en vertical • 4 a 55 °C en horizontal o 45 °C en vertical	4
Longitud de cable (metros)	50, par trenzado apantallado	

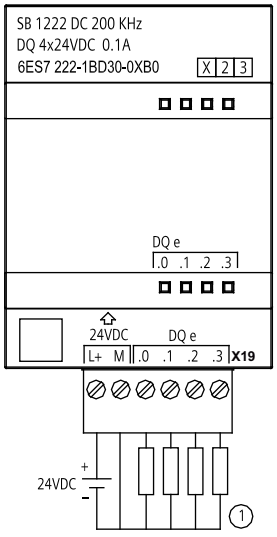
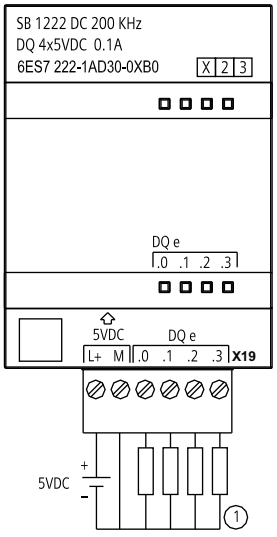
[1] Dado que la misma circuitería admite configuraciones en sumidero y en fuente, el estado activo de una carga en fuente es opuesto al de una carga en sumidero. Una salida de fuente es de lógica positiva (el bit Q y los LED están en ON cuando por la carga pasa corriente); una salida de sumidero es de lógica negativa (el bit Q y los LED están en OFF cuando por la carga pasa corriente). Si el módulo está conectado sin programa de usuario, de forma predeterminada estará a 0 V para este módulo, lo que significa que una carga en sumidero estará ON.

Nota

En caso de conmutar frecuencias superiores a 20 kHz, es importante que las entradas digitales reciban una forma de onda cuadrada. Tenga en cuenta las siguientes posibilidades para mejorar la calidad de señal hacia las entradas:

- Minimizar la longitud del cable
- Cambiar el driver solo sumidero por uno tipo sumidero y fuente
- Cambiar el cable por uno de mayor calidad
- Reducir el circuito/los componentes de 24 V a 5 V
- Agregar una carga externa a la entrada

Tabla A-196 Diagramas de cableado de las SB de salidas digitales 200 kHz

SB 1222 DQ 4 x 24 V DC, 200 kHz (6ES7222-1BD30-0XB0)	**SB 1222 DQ 4 x 5 V DC, 200 kHz (6ES7222-1AD30-0XB0)**
	

① Para salidas en fuente, conectar "Carga" a "-" (como se indica). Para salidas en sumidero, conecte "Carga" a "+". Dado que la misma circuitería admite configuraciones en sumidero y en fuente, el estado activo de una carga en fuente es opuesto al de una carga en sumidero. Una salida de fuente es de lógica positiva (el bit Q y los LED están en ON cuando por la carga pasa corriente); una salida de sumidero es de lógica negativa (el bit Q y los LED están en OFF cuando por la carga pasa corriente). Si el módulo está conectado sin programa de usuario, de forma predeterminada estará a 0 V para este módulo, lo que significa que una carga en sumidero estará ON.

Nota

Asegúrese de que el cable de conexión M está firmemente puesto a tierra. La pérdida de la conexión del cable de tierra a una SB DQ rápida puede hacer que haya suficiente corriente de fuga para activar una carga DC. Si las salidas se utilizan para aplicaciones de carga DC críticas, deben tomarse medidas de precaución adicionales cuando se usa un cable de tierra redundante hacia la SB.

Tabla A-197 Asignación de pines de conectores para SB 1222 DQ 4 x 24 V DC, 200 kHz (6ES7222-1BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

Tabla A-198 Asignación de pines de conectores para SB 1222 DQ 4 x 5 V DC, 200 kHz (6ES7222-1AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DQ e.0
4	DQ e.1
5	DQ e.2
6	DQ e.3

A.13.3 Datos técnicos de la SB 1223 de entradas/salidas digitales 200 kHz

Tabla A-199 Datos técnicos generales

Datos técnicos	SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC, 200 kHz	SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz
Referencia	6ES7223-3BD30-0XB0	6ES7223-3AD30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21	
Peso	35 gramos	
Disipación de potencia	1,0 W	0,5 W
Consumo de corriente (bus SM)	35 mA	
Consumo de corriente (24 V DC)	7 mA / entrada + 30 mA	15 mA / entrada + 15 mA

Tabla A-200 Entradas digitales

Datos técnicos		SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC, 200 kHz	SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz
Número de entradas		2	
Tipo		Fuente	
Tensión nominal		24 V DC a 7 mA, nominal	5 V DC a 15 mA, nominal
Tensión continua admisible		28,8 V DC	6 V DC
Sobretensión transitoria		35 V DC durante 0,5 s	6 V
Señal 1 lógica		De 0 V (10 mA) a L+ menos 10 V (2,9 mA)	De 0 V (20 mA) a L+ menos 2,0 V (5,1 mA)
Señal 0 lógica		De L+ menos 5 V (1,4 mA) a L+ (0 mA)	De L+ menos 1,0 V (2,2 mA) a L + (0 mA)
Frecuencias de entrada de reloj HSC (máx.)		Fase simple: 200 kHz Fase en cuadratura: 160 kHz	
Aislamiento (de campo a lógica)		707 V DC (examen de tipo)	
Grupos de aislamiento		1 (no aislado hacia salidas)	
Tiempos de filtro	Configuración en us	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	
	Configuración en ms	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0	

Datos técnicos	**SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC, 200 kHz**	**SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz**
Número de entradas ON simultáneamente	2	
Longitud de cable (metros)	50, par trenzado apantallado	

Tabla A-201 Salidas digitales

Datos técnicos	**SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC, 200 kHz**	**SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz**
Número de salidas	2	
Tipo de salida	Estado sólido, MOSFET (sumidero y fuente)[1]	
Rango de tensión	De 20,4 a 28,8 V DC	De 4,25 a 6,0 V DC
Valor nominal	24 V DC	5 V DC
Señal 1 lógica a intensidad máx.	L+ menos 1,5 V	L+ menos 0,7 V
Señal 0 lógica a intensidad máx.	1,0 V DC, máx.	0,2 V DC, máx.
Intensidad (máx.)	0,1 A	
Carga de lámparas	--	
Resistencia en estado ON (contactos)	11 Ω máx.	7 Ω máx.
Resistencia en estado desact.	6 Ω máx.	0,2 Ω máx.
Corriente de fuga por salida	--	
Frecuencia de tren de impulsos	200 kHz máx., 2 Hz mín.	
Extracorriente de conexión	0,11 A	
Protección contra sobrecargas	No	
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)	
Grupos de aislamiento	1 (no aislado hacia entradas)	
Intensidad por neutro	0,2 A	
Tensión de bloqueo inductiva	Ninguna	
Retardo de conmutación	1,5 µs + 300 ns asc. 1,5 µs + 300 ns desc.	200 ns + 300 ns asc. 200 ns + 300 ns desc.
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado 0)	
Control de una entrada digital	Sí	
Salidas paralelas para control de carga redundante	No	
Salidas paralelas para carga aumentada	No	
Número de salidas ON simultáneamente	2	
Longitud de cable (metros)	50, par trenzado apantallado	

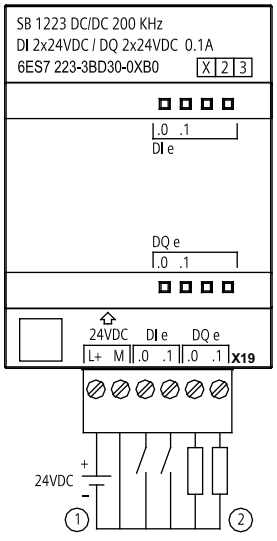
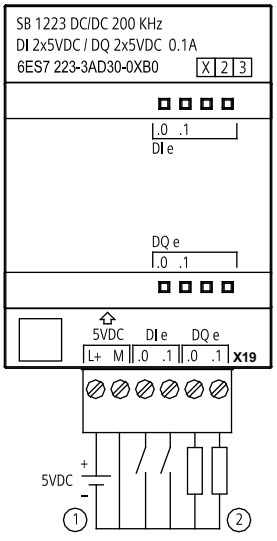
[1] Dado que la misma circuitería admite configuraciones en sumidero y en fuente, el estado activo de una carga en fuente es opuesto al de una carga en sumidero. Una salida de fuente es de lógica positiva (el bit Q y los LED están en ON cuando por la carga pasa corriente); una salida de sumidero es de lógica negativa (el bit Q y los LED están en OFF cuando por la carga pasa corriente). Si el módulo está conectado sin programa de usuario, de forma predeterminada estará a 0 V para este módulo, lo que significa que una carga en sumidero estará ON.

Nota

En caso de conmutar frecuencias superiores a 20 kHz, es importante que las entradas digitales reciban una forma de onda cuadrada. Tenga en cuenta las siguientes posibilidades para mejorar la calidad de señal hacia las entradas:

- Minimizar la longitud del cable
- Cambiar el driver solo sumidero por uno tipo sumidero y fuente
- Cambiar el cable por uno de mayor calidad
- Reducir el circuito/los componentes de 24 V a 5 V
- Agregar una carga externa a la entrada

Tabla A-202 Diagramas de cableado de las SB de entradas/salidas digitales 200 kHz

SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz (6ES7223-3BD30-0XB0)	**SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz (6ES7223-3AD30-0XB0)**
	

① Admite únicamente entradas en fuente

② Para salidas en fuente, conectar "Carga" a "-" (como se indica). Para salidas en sumidero, conecte "Carga" a "+". [1] Dado que la misma circuitería admite configuraciones en sumidero y en fuente, el estado activo de una carga en fuente es opuesto al de una carga en sumidero. Una salida de fuente es de lógica positiva (el bit Q y los LED están en ON cuando por la carga pasa corriente); una salida de sumidero es de lógica negativa (el bit Q y los LED están en OFF cuando por la carga pasa corriente). Si el módulo está conectado sin programa de usuario, de forma predeterminada estará a 0 V para este módulo, lo que significa que una carga en sumidero estará ON.

Nota

Asegúrese de que el cable de conexión M está firmemente puesto a tierra. La pérdida de la conexión del cable de tierra a una SB DQ rápida puede hacer que haya suficiente corriente de fuga para activar una carga DC. Si las salidas se utilizan para aplicaciones de carga DC críticas, deben tomarse medidas de precaución adicionales cuando se usa un cable de tierra redundante hacia la SB.

Tabla A-203 Asignación de pines de conectores para SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz (6ES7223-3BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

Tabla A-204 Asignación de pines de conectores para SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC, 200 kHz (6ES7223-3AD30-0XB0)

Pin	X19
1	L+ / 5 V DC
2	M / 5 V DC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.13.4 Especificaciones de la SB 1223 2 entradas de 24 V DC / 2 salidas de 24 V DC

Tabla A-205 Datos técnicos generales

Datos técnicos	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Referencia	6ES7223-0BD30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	40 gramos
Disipación de potencia	1,0 W
Consumo de corriente (bus SM)	50 mA
Consumo de corriente (24 V DC)	4 mA/entrada utilizada

Tabla A-206 Entradas digitales

Datos técnicos		**SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC**
Número de entradas		2
Tipo		Tipo 1 IEC sumidero
Tensión nominal		24 V DC a 4 mA, nominal
Tensión continua admisible		30 V DC, máx.
Sobretensión transitoria		35 V DC durante 0,5 s
Señal 1 lógica (mín.)		15 V DC a 2,5 mA
Señal 0 lógica (máx.)		5 V DC a 1 mA
Frecuencias de entrada de reloj HSC (máx.)		Fase simple: 30 kHz (de 15 a 26 V DC) Fase en cuadratura: 20 kHz (de 15 a 26 V DC)
Aislamiento (de campo a lógica)		707 V DC (examen de tipo)
Grupos de aislamiento		1
Tiempos de filtro	Configuración en us	0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
	Configuración en ms	0,05, 0,1, 0,2, 0,4, 0,8, 1,6, 3,2, 6,4, 10,0, 12,8, 20,0
Número de entradas ON simultáneamente		2
Longitud de cable (metros)		500 apantallado, 300 no apantallado

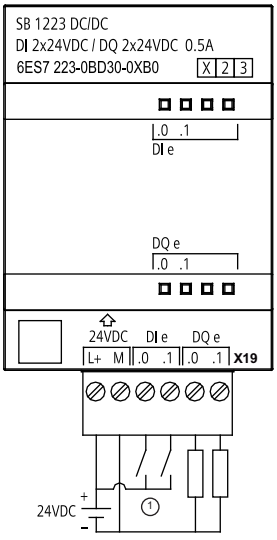
Tabla A-207 Salidas digitales

Datos técnicos	**SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC**
Número de salidas	2
Tipo de salida	Estado sólido MOSFET (fuente)
Rango de tensión	De 20,4 a 28,8 V DC
Señal 1 lógica a intensidad máx.	20 V DC mín.
Señal 0 lógica con carga de 10K Ω	0,1 V DC máx.
Intensidad (máx.)	0,5 A
Carga de lámparas	5 W
Resistencia en estado ON (contactos)	0,6 Ω máx.
Corriente de fuga por salida	10 µA máx.
Frecuencia de tren de impulsos (PTO)	20 kHz máx., 2 Hz mín.[1]
Extracorriente de conexión	5 A durante máx. 100 ms
Protección contra sobrecargas	No
Aislamiento (de campo a lógica)	707 V DC (examen de tipo)
Grupos de aislamiento	1
Intensidad por neutro	1 A
Tensión de bloqueo inductiva	L+ menos 48 V, disipación de 1 W
Retardo de conmutación	2 µs máx. OFF a ON 10 µs máx. ON a OFF
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Control de una entrada digital	Sí
Salidas paralelas para control de carga redundante	No
Salidas paralelas para carga aumentada	No

Datos técnicos	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Número de salidas ON simultáneamente	2
Longitud de cable (metros)	500 m apantallado, 150 m no apantallado

[1] En función del receptor de impulsos y del cable, un resistor de carga adicional (al menos 10% de la intensidad nominal) puede mejorar la calidad de la señal de los impulsos y la inmunidad a interferencias.

Tabla A-208 Diagrama de cableado de la SB de entradas/salidas digitales

SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC (6ES7223-0BD30-0XB0)	
	

(1) Soporta únicamente entradas NPN

Tabla A-209 Asignación de pines de conectores para SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC (6ES7223-0BD30-0XB0)

Pin	X19
1	L+ / 24 V DC
2	M / 24 V DC
3	DI e.0
4	DI e.1
5	DQ e.0
6	DQ e.1

A.14 Signal Boards analógicas (SBs)

A.14.1 Datos técnicos de la SB 1231 de 1 entrada analógica

Nota

Para utilizar esta SB el firmware de la CPU debe tener la versión 2.0 o superior.

Tabla A-210 Especificaciones generales

Datos técnicos	SB 1231 AI 1 x 12 bits
Referencia	6ES7231-4HA30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	35 gramos
Disipación de potencia	0,4 W
Consumo de corriente (bus SM)	55 mA
Consumo de corriente (24 V DC)	Ninguno

Tabla A-211 Entradas analógicas

Datos técnicos	SB 1231 AI 1x12 bits
Número de entradas	1
Tipo	Tensión o intensidad (diferencial)
Rango	±10 V, ±5 V, ±2,5 o de 0 a 20 mA
Resolución	11 bits + bit de signo
Rango total (palabra de datos)	De -27.648 a 27.648
Rango de saturación máximo/mínimo (palabra de datos)	Tensión: de 32.511 a 27.649 / de -27.649 a -32.512 Intensidad: de 32.511 a 27.649 / de 0 a -4.864 (Consulte Representación de entradas analógicas para tensión y representación de entradas analógicas para intensidad (Página 1395).)
Rebase por exceso/defecto (palabra de datos)	Tensión: de 32.767 a 32.512 / de -32.513 a -32.768 Intensidad: de 32.767 a 32.512 / de -4.865 a -32.768 (Consulte Representación de entradas analógicas para tensión y representación de entradas analógicas para intensidad (Página 1395).)
Tensión/intensidad soportada máxima	±35 V / ±40 mA
Filtrado	Ninguno, débil, medio o fuerte (consulte los tiempos de respuesta de las entradas analógicas para más detalles sobre el tiempo de respuesta a un escalón (Página 1395).)
Supresión de ruido	400, 60, 50 o 10 Hz (consulte las frecuencias de muestreo en Tiempos de respuesta de las entradas analógicas (Página 1395).)
Precisión (25 °C / de -20 a 60 °C)	±0,3% / ±0,6% de rango máximo
Impedancia de entrada	Tensión: 150 kΩ; intensidad: 250 Ω
Principio de medición	Conversión de valor real

Datos técnicos	SB 1231 AI 1x12 bits
Rechazo en modo común	40 dB, DC a 60 Hz
Rango de señales operativo	La tensión de señal más la tensión en modo común debe ser menor que +35 V y mayor que -35 V
Aislamiento (de campo a lógica)	Ninguno
Longitud de cable (metros)	100 m, trenzado y apantallado

Tabla A-212 Diagnóstico

Datos técnicos	SB 1231 AI 1 x 12 bits
Rebase por exceso/defecto	Sí
24 V DC baja tensión	No

Transductores de corriente SB 1231

Los transductores de corriente están disponibles como transductores a 2 y a 4 hilos, tal como se muestra abajo.

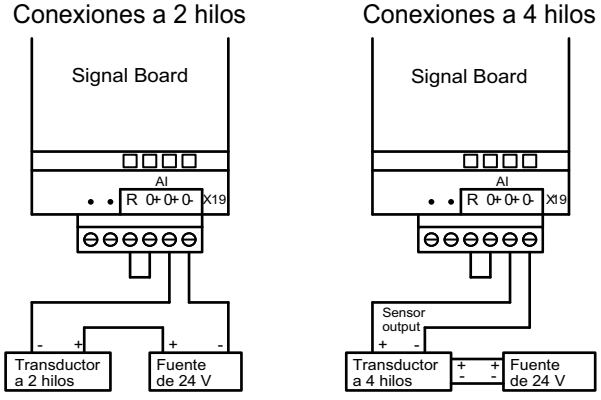


Tabla A-213 Diagrama de cableado de la SB de entrada analógica

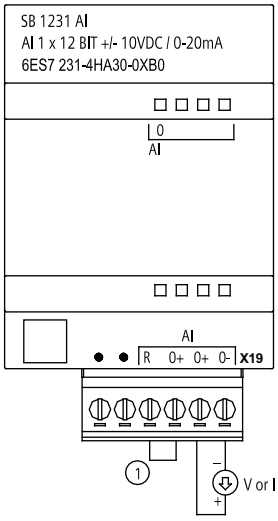
SB 1231 AI x 12 bits (6ES7231-4HA30-0XB0)	
	① Conecte "R" y "0+" para la intensidad. Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.

Tabla A-214 Asignación de pines de conectores para SB 1231 AI x 12 bits (6ES7231-4HA30-0XB0)

Pin	X19 (oro)
1	Sin conexión
2	Sin conexión
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

A.14.2 Datos técnicos de la SB 1232 de 1 salida analógica

Tabla A-215 Especificaciones generales

Datos técnicos	SB 1232 AQ 1 x 12 bits
Referencia	6ES7232-4HA30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	40 gramos
Disipación de potencia	1,5 W
Consumo de corriente (bus SM)	15 mA
Consumo de corriente (24 V DC)	40 mA (sin carga)

Tabla A-216 Salidas analógicas

Datos técnicos	**SB 1232 AQ 1 x 12 bits**
Número de salidas	1
Tipo	Tensión o intensidad
Rango	±10 V o 0 a 20 mA
Resolución	Tensión: 12 bits Intensidad: 11 bits
Rango total (palabra de datos) Consulte los rangos de salida de tensión e intensidad (Página 1396).	Tensión: -27648 a 27648 Intensidad: de 0 a 27.648
Precisión (25 °C / de -20 a 60 °C)	±0,5% / ±1% de rango máximo
Tiempo de estabilización (95% del nuevo valor)	Tensión: 300 µs (R), 750 µs (1 uF) Intensidad: 600 µs (1 mH), 2 ms (10 mH)
Impedancia de carga	Tensión: ≥ 1000 Ω Intensidad: ≤ 600 Ω
Reacción al cambiar de RUN a STOP	Último valor o valor sustitutivo (valor predeterminado: 0)
Aislamiento (de campo a lógica)	Ninguno
Longitud de cable (metros)	100 m, trenzado y apantallado

Tabla A-217 Diagnóstico

Datos técnicos	**SB 1232 AQ 1 x 12 bits**
Rebase por exceso/defecto	Sí
Cortocircuito a tierra (solo en modo de tensión)	Sí
Rotura de hilo (solo en modo de intensidad)	Sí

Tabla A-218 Diagrama de cableado de la SB 1232 AQ 1 x 12 bits

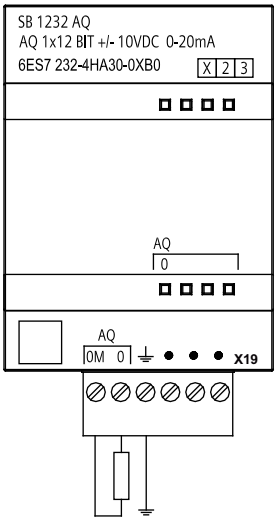
SB 1232 AQ 1 x 12 bits (6ES7232-4HA30-0XB0)	
	
Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-219 Asignación de pines de conectores para SB 1232 AQ 1 x 12 bits (6ES7232-4HA30-0XB0)

Pin	X19 (oro)
1	AQ 0M
2	AQ 0
3	Tierra funcional
4	Sin conexión
5	Sin conexión
6	Sin conexión

A.14.3 Rangos de medida para entradas y salidas analógicas

A.14.3.1 Respuesta a un escalón de las entradas analógicas

Tabla A-220 Respuesta a un escalón (ms), de 0 V a 10 V medidos al 95%

Selección de filtrado (valor medio de muestreo)	Selección del tiempo de integración			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Ninguno (1 ciclo): Sin media	4,5 ms	18,7 ms	22,0 ms	102 ms
Débil (4 ciclos): 4 muestreos	10,6 ms	59,3 ms	70,8 ms	346 ms
Medio (16 ciclos): 16 muestreos	33,0 ms	208 ms	250 ms	1240 ms
Fuerte (32 ciclos): 32 muestreos	63,0 ms	408 ms	490 ms	2440 ms
Tiempo de muestreo	**0,156 ms**	**1,042 ms**	**1,250 ms**	**6,250 ms**

A.14.3.2 Tiempo de muestreo y tiempos de actualización para entradas analógicas

Tabla A-221 Tiempo de muestreo y tiempo de actualización

Selección	Tiempo de muestreo	Tiempo de actualización de la SB
400 Hz (2,5 ms)	0,156 ms	0,156 ms
60 Hz (16,6 ms)	1,042 ms	1,042 ms
50 Hz (20 ms)	1,250 ms	1,25 ms
10 Hz (100 ms)	6,250 ms	6,25 ms

A.14.3.3 Rangos de medida de entradas analógicas de tensión e intensidad (SB y SM)

Tabla A-222 Representación de entradas analógicas de tensión (SB y SM)

Sistema		Rango de medida de tensión				
Decimal	Hexadecimal	±10 V	±5 V	±2,5 V	±1,25 V	
32767	7FFF[1]	11,851 V	5,926 V	2,963 V	1,481 V	Rebase por exceso
32512	7F00					

Sistema		Rango de medida de tensión				
Decimal	**Hexadecimal**	**±10 V**	**±5 V**	**±2,5 V**	**±1,25 V**	
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Rango de sobreimpulso
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Rango nominal
20736	5100	7,5 V	3,75 V	1,875 V	0,938 V	
1	1	361,7 µV	180,8 µV	90,4 µV	45,2 µV	
0	0	0 V	0 V	0 V	0 V	
-1	FFFF					
-20736	AF00	-7,5 V	-3,75 V	-1,875 V	-0,938 V	
-27648	9400	-10 V	-5 V	-2,5 V	-1,250 V	
-27649	93FF					Rango de subimpulso
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	
-32513	80FF					Rebase por defecto
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

[1] 7FFF puede ser devuelto por una de las siguiente razones: desbordamiento (tal como se indica en la tabla), antes de que los valores válidos estén disponibles (p. ej. justo después de un arranque) o si se detecta una rotura de hilo.

Tabla A-223 Representación de entradas analógicas de intensidad (SB y SM)

Sistema		Rango de medida de intensidad		
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	**De 4 mA a 20 mA**	
32767	7FFF	> 23,52 mA	> 22,81 mA	Rebase por exceso
32511	7EFF	23,52 mA	22,81 mA	Rango de sobreimpulso
27649	6C01			
27648	6C00	20 mA	20 mA	Rango nominal
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Rango de subimpulso
-4864	ED00	-3,52 mA	1,185 mA	
32767[1]	7FFF		< 1,185 mA	Rotura de hilo (4 a 20 mA)
-32768	8000	< -3,52 mA		Rebase por defecto (0 a 20 mA)

[1] El valor de rotura de hilo de 32767 (16#7FFF) se devuelve siempre, independientemente del estado de la alarma de rotura de hilo.

A.14.3.4 Rangos de medida de salidas analógicas de tensión e intensidad (SB y SM)

Tabla A-224 Representación de salidas analógicas para tensión (SB y SM)

Sistema		Rango de salida de tensión	
Decimal	**Hexadecimal**	**±10 V**	
32767	7FFF	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	

Sistema		Rango de salida de tensión	
Decimal	**Hexadecimal**	**±10 V**	
32511	7EFF	11,76 V	Rango de sobreimpulso
27649	6C01		
27648	6C00	10 V	Rango nominal
20736	5100	7,5 V	
1	1	361,7 µ V	
0	0	0 V	
-1	FFFF	-361,7 µ V	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Rango de subimpulso
-32512	8100	-11,76 V	
-32513	80FF	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	

[1] En condición de rebase por exceso o defecto, las salidas analógicas adoptarán el valor sustitutivo del estado STOP.

Tabla A-225 Representación de salidas analógicas para intensidad (SB y SM)

Sistema		Rango de salida de intensidad		
Decimal	**Hexadecimal**	**De 0 mA a 20 mA**	**De 4 mA a 20 mA**	
32767	7FFF	V. nota 1	V. nota 1	Rebase por exceso
32512	7F00	V. nota 1	V. nota 1	
32511	7EFF	23,52 mA	22,81 mA	Rango de sobreimpulso
27649	6C01			
27648	6C00	20 mA	20 mA	Rango nominal
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4mA	
-1	FFFF		De 4 mA a 578,7 nA	Rango de subimpulso
-6912	E500		0 mA	
-6913	E4FF			No es posible. Valor de salida limitado a 0 mA.
-32512	8100			
-32513	80FF	V. nota 1	V. nota 1	Rebase por defecto
-32768	8000	V. nota 1	V. nota 1	

[1] En condición de rebase por exceso o defecto, las salidas analógicas adoptarán el valor sustitutivo del estado STOP.

A.14.4 Signal Boards de termopar (SB)

A.14.4.1 Datos técnicos de la SB 1231 de termopar de 1 entrada analógica

Nota

Para utilizar esta SB el firmware de la CPU debe tener la versión 2.0 o superior.

Tabla A-226 Especificaciones generales

Datos técnicos	**SB 1231 AI 1 x 16 bits de termopar**
Referencia	6ES7231-5QA30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	35 gramos
Disipación de potencia	0,5 W
Consumo de corriente (bus SM)	5 mA
Consumo de corriente (24 V DC)	20 mA

Tabla A-227 Entradas analógicas

Datos técnicos		**SB 1231 AI 1x16 bits de termopar**
Número de entradas		1
Tipo		TC flotante y mV
Rango • Rango nominal (palabra de datos) • Rango de saturación máximo/mínimo (palabra de datos) • Rebase por exceso/defecto (palabra de datos)		Véase la tabla de selección de filtros de termopar (Página 1399).
Resolución	Temperatura	0,1 °C / 0,1 °F
	Tensión	Signo más (+) de 15 bits
Tensión soportada máxima		±35 V
Supresión de ruido		85 dB para el ajuste de filtro seleccionado (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Rechazo en modo común		> 120 dB a 120 V AC
Impedancia		≥ 10 M Ω
Precisión		Véase la tabla de selección de termopares (Página 1399).
Repetitividad		±0,05% FS
Principio de medición		Integrador
Tiempo de actualización del módulo		Véase la tabla de selección de filtros de termopar (Página 1399).
Error de unión fría		±1,5 °C
Aislamiento (de campo a lógica)		707 V DC (ensayo de tipo)

Datos técnicos	SB 1231 AI 1x16 bits de termopar
Longitud de cable (metros)	100 m hasta el sensor (máx.)
Resistencia del cable	100 Ω máx.

Tabla A-228 Diagnóstico

Datos técnicos	SB 1231 AI 1 x 16 bits de termopar
Rebase por exceso/defecto[1]	Sí
Rotura de hilo[2, 3]	Sí

1 La información de las alarmas de rebase por exceso/defecto se indicará en los valores de datos analógicos aunque las alarmas estén deshabilitadas en la configuración del módulo.

2 Si la alarma de rotura de hilo está deshabilitada y se presenta una condición de rotura de hilo en la línea del sensor, el módulo puede señalizar valores aleatorios.

3 El módulo comprueba cada 6 segundos si hay una rotura de hilo, con lo que el tiempo de actualización se prolonga en 9 ms para cada canal habilitado una vez cada 6 segundos.

El módulo de señales analógicas de termopar SM 1231 (TC) mide el valor de la tensión conectada a las entradas del módulo.

La Signal Board analógica de termopar SB 1231 mide el valor de la tensión conectada a las entradas de la Signal Board. El tipo de medición de temperatura puede ser "termopar" o "tensión".

- "Termopar": el valor se expresará en grados multiplicados por diez (p. ej. 25,3 grados se expresarán como 253 decimales).
- "Tensión": el valor máximo del rango nominal serán 27648 decimales.

A.14.4.2 Funcionamiento básico de un termopar

Los termopares se forman por la unión de dos metales diferentes que se conectan eléctricamente produciendo una tensión. La tensión generada es proporcional a la temperatura de la unión. Se trata de una tensión pequeña; un microvoltio puede representar varios grados. La medición de temperatura con termopares consiste en medir la tensión de un termopar, compensar las uniones adicionales y linealizar posteriormente el resultado.

Cuando se conecta un termopar al módulo de termopar SM 1231, los dos hilos de metales distintos se unen al conector de señales del módulo. El punto en el que los dos hilos diferentes se unen el uno con el otro constituye el termopar del sensor.

Dos termopares adicionales se forman donde los dos hilos diferentes se unen al conector de señales. La temperatura del conector genera una tensión que se suma a la del termopar del sensor. Si no se corrige esta tensión, la temperatura indicada será diferente de la temperatura del sensor.

La compensación de unión fría se utiliza para compensar el termopar del conector. Las tablas de termopares se basan en una temperatura de referencia que, por lo general, es de cero grados centígrados. La compensación de unión fría compensa el conector a cero grados centígrados. La compensación restablece la tensión sumada por los termopares del conector. La temperatura del módulo se mide internamente y se convierte luego a un valor que se suma a la conversión del sensor. La conversión del sensor corregida se linealiza entonces utilizando las tablas de termopares.

Para optimizar el funcionamiento de la compensación de unión fría es necesario colocar el módulo de termopar en un entorno térmicamente estable. Una variación lenta (inferior a 0,1 °C/minuto) del módulo a temperatura ambiente se compensa correctamente dentro de las especificaciones del módulo. Si hay corriente de aire a través del módulo también se producirán errores de compensación de unión fría.

Si se requiere una mejor compensación del error de unión fría, se puede utilizar un bloque de terminales isotérmico. El módulo de termopar permite utilizar un bloque de terminales con una referencia de 0 °C o 50 °C.

Tabla de selección de termopares para la SB 1231

Los rangos y la exactitud de los diferentes tipos de termopares soportados por la Signal Board de termopar SB 1231 se indican en la tabla siguiente.

Tabla A-229 Tabla de selección de termopares

Tipo	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión en rango normal[3, 4] a 25 °C	Precisión en rango normal[1, 2, 6] de -20 °C a 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

Tipo	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión en rango normal[3, 4] a 25 °C	Precisión en rango normal[1, 2, 6] de -20 °C a 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
Tensión	-32512	-27648 -80mV	27648 80mV	32511	±0,05%	±0,1%

1 Los valores de termopar inferiores al valor del rango de saturación mínimo se notifican como -32768.

2 Los valores de termopar por encima del valor máximo del rango superior se notifican como 32767.

3 El error de la unión fría interna es de ±1.5 °C en todos los rangos. Esto debe añadirse al error en esta tabla. El módulo requiere un tiempo mínimo de calentamiento de 30 minutos para cumplir esta especificación.

4 En presencia de radiofrecuencia radiada de 970 MHz a 990 MHz, la precisión del SM 1231 AI 4 x 16 bits TC puede verse reducida.

5 Límite inferior de 3276,6 °F con indicación °F

6 El error de compensación de la unión fría interna no se ha caracterizado para temperaturas ambiente del módulo inferiores a 0 °C y puede que exceda el valor especificado.

Tabla A-230 Tabla de selección de filtros para el termopar de la SB 1231

Supresión de frecuencias (Hz)	Tiempo de integración (ms)	Tiempo de actualización de la Signal Board (segundos)
10	100	0,306
50	20	0,066
60	16,67	0,056
400[1]	10	0,036

1 Para mantener la resolución y precisión del módulo con supresión de 400 Hz seleccionada, el tiempo de integración es de 10 ms. Esta selección también suprime el ruido de 100 Hz y 200 Hz.

Para medir termopares se recomienda utilizar un tiempo de integración de 100 ms. El uso de tiempos de integración inferiores aumentará el error de repetibilidad de las lecturas de temperatura.

Nota

Tras aplicar la alimentación, el módulo realiza una calibración interna para el convertidor analógico-digital. Durante ese tiempo el módulo devuelve un valor de 32767 en cada canal hasta que haya datos válidos en ese canal. Es posible que el programa de usuario deba autorizar este tiempo de inicialización. Dado que la configuración del módulo puede variar la duración del tiempo de inicialización, conviene verificar el comportamiento o el módulo en la configuración. En caso necesario, se puede incluir lógica en el programa de usuario para acomodar el tiempo de inicialización del módulo.
Esta lógica puede implementarse utilizando una lectura de sondeo en el "OB de arranque", lo que bloquea la operación hasta que la inicialización se haya completado. Es necesario implementar la lectura de sondeo con un acceso inmediato. Si el valor de la lectura de sondeo de termopar es 32767, habrá que repetir la lectura hasta que el valor cambie. Para cada módulo el sondeo solo tiene que ejecutarse para el punto de entrada con el número más alto utilizado (las entradas del módulo se inicializan en orden de 0 a 7).

Tabla A-231 Diagrama de cableado de la SB 1231 AI 1 x 16 termopar

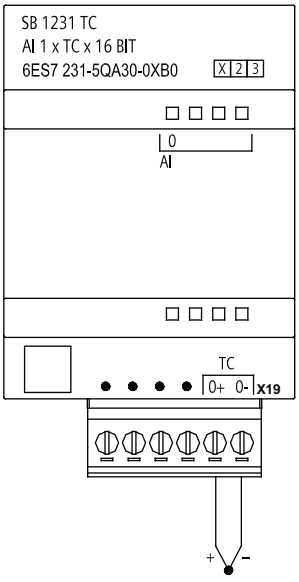
SB 1231 AI 1 x 16 bits termopar (6ES7231-5QA30-0XB0)	
	
Nota: los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.	

Tabla A-232 Asignación de pines de conectores para SB 1231 AI 1 x 16 bits termopar (6ES7231-5QA30-0XB0)

Pin	X19 (oro)
1	Sin conexión
2	Sin conexión
3	Sin conexión
4	Sin conexión
5	AI 0-/TC
6	AI 0+/TC

A.14.5 Signal Boards RTD (SB)

A.14.5.1 Datos técnicos de la SB 1231 de 1 entrada analógica RTD

Nota

Para utilizar esta SB el firmware de la CPU debe tener la versión 2.0 o superior.

Tabla A-233 Especificaciones generales

Datos técnicos	**SB 1231 AI 1 x 16 bits RTD**
Referencia	6ES7231-5PA30-0XB0
Dimensiones A x A x P (mm)	38 x 62 x 2
Peso	35 gramos
Disipación de potencia	0,7 W
Consumo de corriente (bus SM)	5 mA
Consumo de corriente (24 V DC)	25 mA

Tabla A-234 Entradas analógicas

Datos técnicos		**SB 1231 AI 1 x 16 bits RTD**
Número de entradas		1
Tipo		RTD y ohmios indicados por el módulo
Rango • Rango nominal (palabra de datos) • Rango de saturación máximo/mínimo (palabra de datos) • Rebase por exceso/defecto (palabra de datos)		Véanse las tablas de selección (Página 1405).
Resolución	Temperatura	0,1 °C/0,1 °F
	Tensión	Signo más (+) de 15 bits
Tensión soportada máxima		± 35 V
Supresión de ruido		85 dB (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Rechazo en modo común		> 120 dB
Impedancia		≥ 10 MΩ
Precisión		Véanse las tablas de selección (Página 1405).
Repetitividad		±0,05% FS
Disipación máxima del sensor		0,5 mW
Principio de medición		Integrador
Tiempo de actualización del módulo		Véase la tabla de selección (Página 1405).
Aislamiento (de campo a lógica)		707 V DC (ensayo de tipo)
Longitud de cable (metros)		100 m hasta el sensor (máx.)
Resistencia del cable		20 Ω, 2,7 para 10 Ω RTD máx.

Tabla A-235 Diagnóstico

Datos técnicos	**SB 1231 AI 1 x 16 bits RTD**
Rebase por exceso/defecto [1, 2]	Sí
Rotura de hilo [3]	Sí

[1] La información de las alarmas de rebase por exceso/defecto se indicará en los valores de datos analógicos aunque las alarmas estén deshabilitadas en la configuración del módulo.

[2] La detección de rebase por defecto nunca está habilitada para los rangos de resistencia.

[3] Si la alarma de rotura de hilo está deshabilitada y se presenta una condición de rotura de hilo en la línea del sensor, el módulo puede señalizar valores aleatorios.

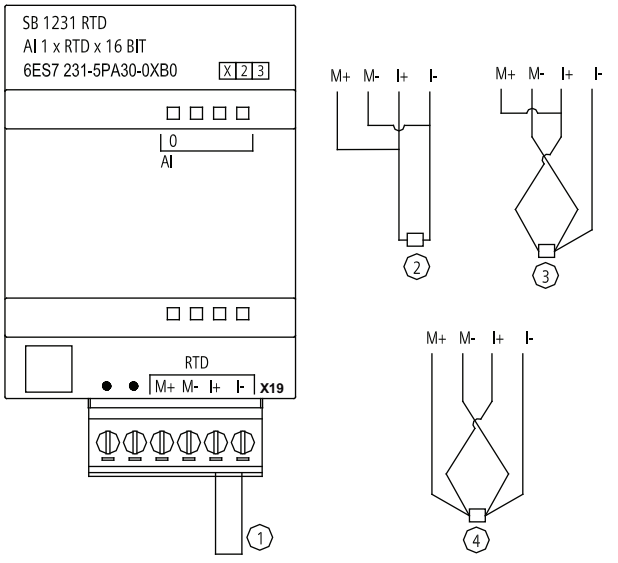
La Signal Board analógica SB 1231 RTD mide el valor de la resistencia conectada a las entradas de la Signal Board. El tipo de medición puede elegirse entre "resistor" y "resistor térmico".

- "Resistor": el valor máximo del rango nominal serán 27648 decimales.
- "Resistor térmico": el valor se expresará en grados multiplicados por diez (p. ej. 25,3 grados se expresarán como 253 decimales). Los valores del rango climático se expresarán en grados multiplicados por cien (p. ej. 25,34 grados se expresarán como 2534 decimales).

La Signal Board SB 1231 RTD soporta mediciones con base en conexiones de 2, 3 y 4 hilos que van al sensor de resistencia.

Tabla A-236 Diagrama de cableado de la SB 1231 AI 1 x 16 bits RTD

SB 1231 AI 1 x 16 bits RTD (6ES7231-5PA30-0XB0)



① Entrada de bucle RTD no utilizada
② RTD de dos hilos
③ RTD de tres hilos
④ RTD de cuatro hilos

Nota: Los conectores deben ser de oro. Consulte el anexo C, piezas de repuesto, para ver la referencia.

Tabla A-237 Asignación de pines de conectores para SB 1231 AI 1 x 16 bits RTD (6ES7231-5PA30-0XB0)

Pin	X19 (oro)
1	Sin conexión
2	Sin conexión
3	AI 0 M+/RTD
4	AI 0 M-/RTD

Pin	X19 (oro)
5	AI 0 I+/RTD
6	AI 0 I-/RTD

A.14.5.2 Tablas de selección para la SB 1231 RTD

Tabla A-238 Rangos y precisión de los diferentes sensores soportados por los módulos RTD

Coeficiente de temperatura	Tipo de RTD	Rango de saturación mínimo[1]	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[2]	Precisión de rango normal a 25 °C	Precisión de rango normal entre -20 °C y 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 climatiz.	-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						
	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 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 °	±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] Los valores de RTD inferiores al valor del rango de saturación mínimo se notifican como -32768.

[2] Los valores de RTD superiores al valor del rango de saturación máximo se notifican como +32768.

Tabla A-239 Resistencia

Rango	Rango de saturación mínimo	Rango nominal límite inferior	Rango nominal límite superior	Rango de saturación máximo[1]	Precisión de rango normal a 25 °C	Precisión de rango normal entre -20 °C y 60 °C
150 Ω	n/a	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0,05 %	±0,1 %
300 Ω	n/a	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0,05 %	±0,1 %
600 Ω	n/a	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0,05 %	±0,1 %

[1] Los valores de resistencia superiores al valor del rango de saturación máximo se notifican como 32767.

Nota

El módulo reporta 32767 en todo canal que esté activado y que no tenga conectado ningún sensor. Si la detección de rotura de hilo también está habilitada, en el módulo se encenderán de forma intermitente los LEDs rojos correspondientes.

La exactitud más elevada se alcanza para los rangos de 10 Ω del RTD con conexiones a 4 hilos.

La resistencia de los hilos en el modo a 2 hilos dará lugar a un error de lectura en el sensor y por ello no se garantiza la exactitud.

Tabla A-240 Reducción de ruido y tiempos de actualización para los módulos RTD

Selección de supresión de frecuencias	Tiempo de integración	Módulo de 4/2 hilos y 1 canal Tiempo de actualización (segundos)	Módulo de 3 hilos y 1 canal Tiempo de actualización (segundos)
400 Hz (2,5 ms)	10 ms[1]	0,036	0,071
60 Hz (16,6 ms)	16,67 ms	0,056	0,111
50 Hz (20 ms)	20 ms	0,066	1,086
10 Hz (100 ms)	100 ms	0,306	0,611

[1] Para mantener la resolución y precisión del módulo con el filtro 400 Hz seleccionado, el tiempo de integración es de 10 ms. Esta selección también suprime el ruido de 100 Hz y 200 Hz.

Nota

Después de aplicar tensión al módulo, este lleva a cabo una calibración interna del conversor de analógico a digital. Durante este tiempo, el módulo indica un valor de 32767 en cada canal hasta que haya información válida disponible en el canal respectivo. Es posible que el programa de usuario deba autorizar este tiempo de inicialización. Como la configuración del módulo puede modificar la longitud del tiempo de inicialización, es conveniente verificar el comportamiento o el módulo en la configuración. Si es necesario, se puede incluir lógica en el programa de usuario para adaptarse al tiempo de inicialización del módulo.
Esta lógica puede implementarse utilizando una lectura de sondeo en el "OB de arranque", lo que bloquea la operación hasta que la inicialización se haya completado. Es necesario implementar la lectura de sondeo con un acceso inmediato. Si el valor de la lectura de sondeo de RTD es 32767, habrá que repetir la lectura hasta que el valor cambie. Para cada módulo el sondeo solo tiene que ejecutarse para el punto de entrada con el número más alto utilizado (las entradas del módulo se inicializan en orden de 0 a 7).

A.15 BB 1297 Battery Board

BB 1297 Battery Board

La Battery Board (placa de batería) BB 1297 de S7-1200 está pensada para proporcionar respaldo a largo plazo del reloj de tiempo real. Se puede conectar en la ranura para Signal Board de la CPU S7-1200 (firmware 3.0 y posteriores). Debe agregar la BB 1297 a la configuración de dispositivo y descargar la configuración de hardware en la CPU para que la BB esté operativa.

La batería (tipo CR1025) no se suministra con la BB 1297 y debe adquirirla el usuario.

Nota

La BB 1297 tiene un diseño mecánico que permite que las CPU cuenten con el firmware 3.0 y versiones posteriores.

No utilice la BB 1297 con CPU de versiones anteriores, ya que el conector de la BB 1297 no se podrá enchufar a la CPU.

⚠ ADVERTENCIA

La instalación de una batería sin especificar en la BB 1297 o la conexión de cualquier otro modo de una batería sin especificar en el circuito puede provocar un incendio o causar daños materiales y un comportamiento imprevisto de la maquinaria.

Un incendio o el comportamiento imprevisto de la maquinaria podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Utilice solo la batería CR1025 especificada para el respaldo del reloj en tiempo real.

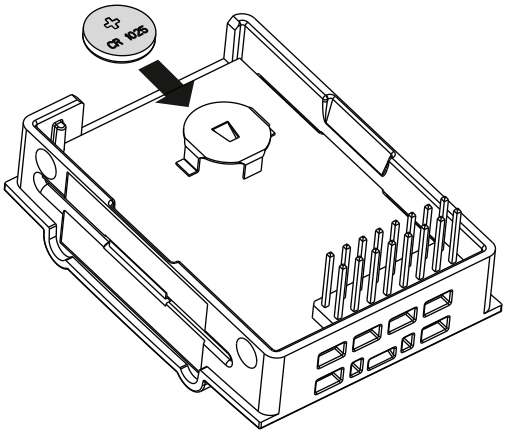
Tabla A-241 Especificaciones generales

Datos técnicos	BB 1297 Battery Board
Referencia	6ES7297-0AX30-0XA0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	28 gramos
Disipación de potencia	0,5 W
Consumo de corriente (bus SM)	11 mA
Consumo de corriente (24 V DC)	ninguno

Batería (no incluida)	BB 1297 Battery Board
Tiempo de retención	Aproximadamente 1 año
Tipo de batería	CR1025 Consulte Instalar o sustituir la batería en la BB 1297 (Página 52)
Tensión nominal	3 V
Capacidad nominal	30 mAh mínimo

Diagnóstico	BB 1297 Battery Board
Nivel crítico de batería	< 2,5 V
Diagnóstico de batería	Indicador de baja tensión: • La baja tensión de la batería hace que el LED MAINT de la CPU se encienda con una luz ámbar continua. • Evento de búfer de diagnóstico: 16#06:2700 "Mantenimiento solicitado de submódulo: Como mínimo hay una batería agotada (BATTF)"
Estado de la batería	Hay un bit de estado de la batería 0 = Batería OK 1 = Batería baja
Actualización del estado de la batería	El estado de la batería se actualiza durante el encendido y después, diariamente mientras la CPU está en estado operativo RUN.

Tabla A-242 Diagrama de inserción para la BB 1297

BB 1297 Battery Board (6ES7297-0AX30-0XA0)	
	

A.16 Interfaces de comunicación

A.16.1 PROFIBUS

A.16.1.1 CM 1242-5 ESCLAVO PROFIBUS DP

Tabla A-243 Datos técnicos del CM 1242-5

Datos técnicos	
Referencia	6GK7242-5DX30-0XE0
Interfaces	

Datos técnicos	
Conexión a PROFIBUS	Conector hembra sub-D de 9 pines
Consumo de corriente máxima en la interfaz PROFIBUS cuando se conectan componentes de red (por ejemplo, componentes de red ópticos)	15 mA a 5 V (solo para cierre del bus) *)
Condiciones ambientales admisibles	
Temperatura ambiente • durante el almacenamiento • durante el transporte • durante la operación con instalación vertical (perfil DIN simétrico horizontal) • durante la operación con instalación horizontal (perfil DIN simétrico vertical)	 • De -40 °C a 70 °C • De -40 °C a 70 °C • De 0 °C a 55 °C • De 0 °C a 45 °C
Humedad relativa a 25 °C durante el funcionamiento, sin condensación, máxima	95 %
Grado de protección	IP20
Fuente de alimentación, consumo de corriente y pérdidas	
Tipo de fuente de alimentación	DC
Alimentación eléctrica del bus de fondo	5 V
Consumo de corriente (típico)	150 mA
Disipación efectiva (típica)	0,75 W
Aislamiento galvánico • Interfaz PROFIBUS conectada a tierra • Interfaz PROFIBUS conectada a circuito interno	710 V DC durante 1 minuto
Dimensiones y peso	
• Ancho • Altura • Profundidad	• 30 mm • 100 mm • 75 mm
Peso • Peso neto • Peso con embalaje	 • 115 g • 152 g

*)La carga de corriente de un consumidor externo conectado entre VP (pin 6) y DGND (pin 5) no debe exceder un máximo de 15 mA (a prueba de cortocircuito) para el cierre del bus.

A.16.1.2 Asignación de pines del conector hembra Sub D del CM 1242-5

Interfaz PROFIBUS

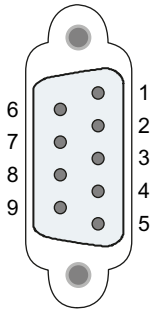


Tabla A-244 Asignación de pines del conector sub-D

Pin	Descripción	Pin	Descripción
1	- no utilizado -	6	P5V2: alimentación de +5V
2	- no utilizado -	7	- no utilizado -
3	RxD/TxD-P: línea de datos B	8	RxD/TxD-N: línea de datos A
4	RTS	9	- no utilizado -
5	M5V2: potencial de referencia de datos (tierra DGND)	Carcasa	Conector de tierra

A.16.1.3 CM 1243-5 maestro PROFIBUS DP

Tabla A-245 Datos técnicos del CM 1243-5

Datos técnicos	
Referencia	6GK7243-5DX30-0XE0
Interfaces	
Conexión a PROFIBUS	Conector hembra sub-D de 9 pines
Consumo de corriente máxima en la interfaz PROFIBUS cuando se conectan componentes de red (por ejemplo, componentes de red ópticos)	15 mA a 5 V (solo para cierre del bus) *)
Condiciones ambientales admisibles	
Temperatura ambiente	
• durante el almacenamiento	• De -40 °C a 70 °C
• durante el transporte	• De -40 °C a 70 °C
• durante la operación con instalación vertical (perfil DIN simétrico horizontal)	• De 0 °C a 55 °C
• durante la operación con instalación horizontal (perfil DIN simétrico vertical)	• De 0 °C a 45 °C

Datos técnicos	
Humedad relativa a 25 °C durante el funcionamiento, sin condensación, máxima	95 %
Grado de protección	IP20
Fuente de alimentación, consumo de corriente y pérdidas	
Tipo de fuente de alimentación	DC
Alimentación eléctrica / externa • mínimo • máximo	24 V • 19,2 V • 28,8 V
Consumo de corriente (típico) • de 24 V DC • del bus de fondo del S7-1200	 • 100 mA • 0 mA
Disipación efectiva (típica) • de 24 V DC • del bus de fondo del S7-1200	 • 2,4 W • 0 W
Alimentación de 24 V DC / externa • Sección de cable mín. • Sección de cable máx. • Par de apriete de los terminales atornillados	 • mín.: 0,14 mm^2 (AWG 25) • máx.: 1,5 mm^2 (AWG 15) • 0,45 Nm (4 lb-in)
Aislamiento galvánico • Interfaz PROFIBUS conectada a tierra • Interfaz PROFIBUS conectada a circuito interno	710 V DC durante 1 minuto
Dimensiones y peso	
• Ancho • Altura • Profundidad	• 30 mm • 100 mm • 75 mm
Peso • Peso neto • Peso con embalaje	 • 134 g • 171 g

*)La carga de corriente de un consumidor externo conectado entre VP (pin 6) y DGND (pin 5) no debe exceder un máximo de 15 mA (a prueba de cortocircuito) para el cierre del bus.

Nota

El CM 1243-5 (módulo maestro PROFIBUS) necesita la alimentación de sensores de 24 V DC de la CPU.

A.16.1.4 Asignación de pines del conector sub-D del CM 1243-5

Interfaz PROFIBUS

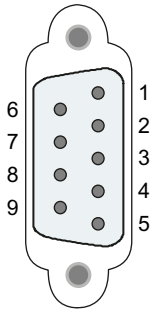


Tabla A-246 Asignación de pines del conector sub-D

Pin	Descripción	Pin	Descripción
1	- no utilizado -	6	VP: alimentación de +5 V solo para resistencias terminadoras del bus; no para suministrar dispositivos externos
2	- no utilizado -	7	- no utilizado -
3	RxD/TxD-P: línea de datos B	8	RxD/TxD-N: línea de datos A
4	CNTR-P: RTS	9	- no utilizado -
5	DGND: tierra para señales de datos y VP	Carcasa	Conector de tierra

Cable PROFIBUS

Nota

Conectar la pantalla del cable PROFIBUS

La pantalla del cable PROFIBUS debe conectarse.

Para ello, pele el aislamiento del extremo del cable PROFIBUS y conecte la pantalla a la tierra funcional.

A.16.2 CP 1242-7

Nota

El CP 1242-7 no está aprobado para aplicaciones marítimas

El CP 1242-7 no dispone de homologación marítima

Nota

Para utilizar estos módulos el firmware de la CPU debe tener la versión 2.0 o superior.

A.16.2.1 CP 1242-7 GPRS

Tabla A-247 Datos técnicos del CP 1242-7 GPRS V2

Datos técnicos	
Referencia	6GK7242-7KX3-0XE0
Interfaz inalámbrica	
Conector de antena	Conector SMA
Impedancia nominal	50 ohmios
Conexión inalámbrica	
Potencia de transmisión máxima	• GSM 850, clase 4: +33 dBm ±2dBm • GSM 900, clase 4: +33 dBm ±2dBm • GSM 1800, clase 1: +30 dBm ±2dBm • GSM 1900, clase 1: +30 dBm ±2dBm
GPRS	Multislot clase 10 dispositivo clase B esquema de codificación 1...4 (GMSK)
SMS	Modo de salida: MO Servicio: punto a punto
Condiciones ambientales admisibles	
Temperatura ambiente	
• durante el almacenamiento	• De -40 °C a 70 °C
• durante el transporte	• De -40 °C a 70 °C
• durante la operación con instalación vertical (perfil DIN simétrico horizontal)	• De 0 °C a 55 °C
• durante la operación con instalación horizontal (perfil DIN simétrico vertical)	• De 0 °C a 45 °C
Humedad relativa a 25 °C durante el funcionamiento, sin condensación, máxima	95 %
Grado de protección	IP20
Fuente de alimentación, consumo de corriente y pérdidas	
Tipo de fuente de alimentación	DC
Alimentación eléctrica / externa	24 V
• mínimo	• 19,2 V
• máximo	• 28,8 V
Consumo de corriente (típico)	
• de 24 V DC	• 100 mA
• del bus de fondo del S7-1200	• 0 mA

Datos técnicos	
Disipación efectiva (típica) • de 24 V DC • del bus de fondo del S7-1200	 • 2,4 W • 0 W
Alimentación de 24 V DC • Sección de cable mín. • Sección de cable máx. • Par de apriete de los terminales atornillados	 • mín.: 0,14 mm^2 (AWG 25) • máx.: 1,5 mm^2 (AWG 15) • 0,45 Nm (4 lb-in)
Aislamiento galvánico Unidad de alimentación a circuito interno	710 V DC durante 1 minuto
Dimensiones y peso	
• Ancho • Altura • Profundidad	• 30 mm • 100 mm • 75 mm
Peso • Peso neto • Peso con embalaje	 • 133 g • 170 g

Nota

Evitar interferencias en la CPU provenientes de las antenas

Es posible que se produzcan interferencias en la CPU si hay una antena demasiado cerca o si no se utilizan las antenas recomendadas. Encontrará las antenas recomendadas en Antenna ANT794-4MR for LTE/UMTS/GSM Compact Operating Instructions (https://support.industry.siemens.com/cs/ww/es/view/23119005/en) (solo disponibles en inglés y alemán)

A.16.2.2 Antena GSM/GPRS ANT794-4MR

Datos técnicos de la antena ANT794-4MR GSM/GPRS

ANT794-4MR	
Referencia	6NH9860-1AA00
Redes inalámbricas móviles	GSM/GPRS
Rangos de frecuencia	• 824 a 960 MHz (GSM 850, 900) • 1 710 a 1 880 MHz (GSM 1 800) • 1 900 a 2 200 MHz (GSM / UMTS)
Características	Omnidireccional
Ganancia de la antena	0 dB
Impedancia	50 ohmios
Relación de onda estacionaria (SWR)	< 2,0
Potencia máx.	20 W
Polaridad	Lineal vertical

ANT794-4MR	
Conector	SMA
Longitud del cable de la entena	5 m
Material externo	PVC duro, resistente a los rayos UV
Grado de protección	IP20
Condiciones ambientales admisibles • Temperatura de servicio • Temperatura de transporte/almacenamiento • Humedad relativa	 • De -40 °C a +70 °C • De -40 °C a +70 °C • 100 %
Material externo	PVC duro, resistente a los rayos UV
Construcción	Antena con 5 m de cable fijo y un conector macho SMA
Dimensiones (d x al) en mm	25 x 193
Peso • Antena incluido el cable • Boquillas	 • 310 g • 54 g
Instalación	Con soporte suministrado

A.16.2.3 Antena plana ANT794-3M

Datos técnicos de la antena plana ANT794-3M

ANT794-3M		
Referencia	6NH9870-1AA00	
Redes inalámbricas móviles	**GSM 900**	**GSM 1800/1900**
Rangos de frecuencia	890 - 960 MHz	1710 - 1990 MHz
Relación de onda estacionaria (VSWR)	≤ 2:1	≤ 1,5:1
Coeficiente de reflexión (Tx)	≈ 10 dB	≈ 14 dB
Ganancia de la antena	0 dB	
Impedancia	50 ohmios	
Potencia máx.	10 W	
Cable de antena	Cable HF RG 174 (fijo) con conector macho SMA	
Longitud del cable	1,2 m	
Grado de protección	IP64	
Rango de temperatura admisible	De -40 °C a +75 °C	
Inflamabilidad	UL 94 V2	
Material externo	ABS Polylac PA-765, gris claro (RAL 7035)	
Dimensiones (an x la x al) en mm	70,5 x 146,5 x 20,5	
Peso	130 g	

A.16.3 CM 1243-2 AS-i Master

A.16.3.1 Datos técnicos del maestro AS-i CM 1243-2

Tabla A-248 Datos técnicos del maestro AS-i CM 12432-2

Datos técnicos	
Referencia	3RK7243-2AA30-0XB0
Versión de firmware	V1.0
Fecha	01.12.2011
Interfaces	
Consumo máximo de corriente Del bus de fondo del S7-1200 Del cable AS-i	 Máx. 250 mA, tensión de alimentación del bus de comunicación S7-1200 5 V DC Máx. 100 mA
Capacidad máxima de carga de corriente entre los terminales ASI+/ASI-	8 A
Asignación de los pines	Véase el apartado Conexiones eléctricas del maestro AS-i (Página 1417)
Sección del conductor	0,2 mm² (AWG 24) ... 3,3 mm² (AWG 12)
Par de apriete del conector ASI	0,56 Nm
Condiciones ambientales admisibles	
Temperatura ambiente Durante el almacenamiento Durante el transporte Durante la fase operativa con instalación vertical (perfil DIN simétrico horizontal) Durante la fase operativa con instalación horizontal (perfil DIN simétrico vertical)	 De -40 °C a 70 °C De -40 °C a 70 °C De 0 °C a 55 °C De 0 °C a 45 °C
Humedad relativa a 25 °C durante la fase operativa, sin condensación, máxima	95 %
Grado de protección	IP20
Fuente de alimentación, consumo de corriente, pérdidas	
Tipo de fuente de alimentación	DC
Consumo de corriente (típico) Del bus de fondo del S7-1200	 200 mA
Disipación total (típica): • Del bus de fondo del S7-1200 • Del cable AS-i	 1 W 2,4 W
Dimensiones y peso	

Datos técnicos	
Ancho	30 mm
Altura	100 mm
Profundidad	75 mm
Peso	
Peso neto	122 g
Peso con embalaje	159 g

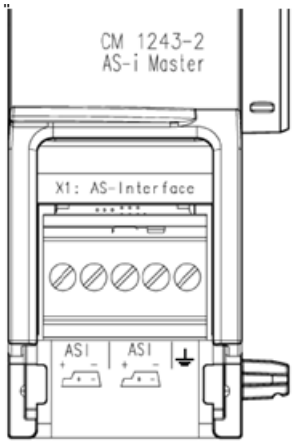
A.16.3.2 Conexiones eléctricas del maestro AS-i

Fuente de alimentación del maestro AS-i CM 1243-2

El maestro AS-i CM 1243-2 se alimenta a través del bus de comunicaciones del S7-1200. Esto significa que es posible enviar un aviso de diagnóstico al S7-1200 tras un fallo de la alimentación AS-i. La conexión del bus de comunicación se encuentra en el lado derecho del maestro AS-i CM 1243-2.

Terminales de la AS-Interface

El terminal desenchufable para conectar el cable AS-i se encuentra bajo la tapa inferior en el frente del maestro AS-i CM 1243-2.



En caso de utilizar el cable AS-i, la polaridad correcta se reconoce por el símbolo



Encontrará información acerca de cómo eliminar y volver a instalar el bloque de terminales en el capítulo Instalación (Página 56).

Nota

Capacidad máxima de carga de corriente de los contactos de terminales

La capacidad máxima de carga de corriente de los contactos de conexión es de 8 A. Si se excede este valor en el cable AS-i, el maestro AS-i CM 1243-2 no debe "enlazarse" al cable AS-i sino que debe conectarse mediante un cable de derivación (solo un par de conexión asignado al maestro AS-i CM 1243-2).
Asegúrese también de que los cables usados son adecuados para temperaturas de funcionamiento de mín. 75 °C si la corriente se conduce mediante el maestro AS-i y se dan corrientes superiores a 4 amperios.
Encontrará más información sobre la conexión de cables AS-i en el apartado "Montaje, conexión y puesta en marcha de los módulos" del manual de producto "Maestro AS-i CM 1243-2 y módulo de desacoplamiento de datos AS-i DCM 1271 para SIMATIC S7-1200".

Asignaciones de terminales

Etiqueta	Significado
ASI+	Conexión AS-i – polaridad positiva
ASI–	Conexión AS-i – polaridad negativa
	Tierra funcional

A.16.4 RS232, RS422 y RS485

A.16.4.1 Datos técnicos de CB 1241 RS485

Nota

Para utilizar esta CB el firmware de la CPU debe tener la versión 2.0 o superior.

Tabla A-249 Especificaciones generales

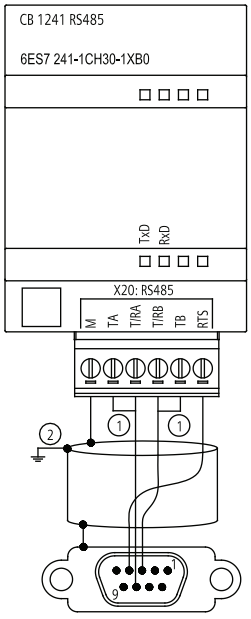
Datos técnicos	CB 1241 RS485
Referencia	6ES7241-1CH30-1XB0
Dimensiones A x A x P (mm)	38 x 62 x 21
Peso	40 gramos

Tabla A-250 Transmisor y receptor

Datos técnicos	**CB 1241 RS485**
Tipo	RS485 (semidúplex de 2 hilos)
Rango de tensión en modo común	-7 V a +12 V, 1 segundo, 3 VRMS continuo
Tensión de salida diferencial del transmisor	2 V mín. a R_L = 100 Ω 1,5 V mín. a R_L = 54 Ω
Terminación y polarización	10 K a +5 V en B, pin 3 RS485 10K a GND en A, pin 4 RS485
Terminación opcional	Conexión del pin TB al pin T/RB, la impedancia de terminación efectiva es de 127 Ω, se conecta al pin 3 RS485 Conexión del pin TA al pin T/RA, la impedancia de terminación efectiva es de 127 Ω, se conecta al pin 4 RS485
Impedancia de entrada del receptor	5,4K Ω mín. incluyendo terminación
Umbral/sensibilidad del receptor	+/- 0,2 V mín., 60 mV de histéresis típica
Aislamiento Señal RS485 a conexión a masa Señal RS485 a lógica de la CPU	707 V DC (ensayo de tipo)
Longitud de cable, apantallado	1000 m máx.
Velocidad de transferencia	300 baudios, 600 baudios, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (valor predeterminado), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Paridad	Sin paridad (valor predeterminado), par, impar, marca (bit de paridad siempre puesto a 1), espacio (bit de paridad siempre puesto a 0)
Número de bits de parada	1 (valor predeterminado), 2
Control de flujo	No soportado
Tiempo de espera	0 a 65535 ms

Tabla A-251 Alimentación eléctrica

Datos técnicos	**CB 1241 RS485**
Pérdida de potencia (disipación)	1,5 W
Consumo de corriente (bus SM), máx.	50 mA
Consumo de corriente (24 V DC) máx.	80 mA

CB 1241 RS485 (6ES7241-1CH30-1XB0)	
	
① Conecte "TA" y TB" tal y como se indica para finalizar la red. (En la red RS485 solo deben finalizarse los dispositivos terminadores.)	
② Utilice un cable de par trenzado apantallado y conéctelo a tierra.	

Solo se terminan los dos extremos de la red RS485. Los dispositivos que están entre los dos dispositivos terminadores no se terminan ni polarizan. Consulte el artículo "Polarizar y terminar un conector de red RS485" (Página 938)

Tabla A-252 Asignación de pines de conectores para CB 1241 RS485 (6ES7241-1CH30-1XB0)

Pin	Conector de 9 pines	X20
1	RS485/GND de lógica	--
2	RS485/No utilizado	--
3	RS485/TxD+	4 - T/RB
4	RS485/RTS	6 - RTS
5	RS485/GND de lógica	--
6	RS485 / alimentación 5 V	--
7	RS485/No utilizado	--
8	RS485/TxD-	3 - T/RA
9	RS485/No utilizado	--
Carcasa		1 - M

A.16.4.2 Datos técnicos del CM 1241 RS232

Tabla A-253 Especificaciones generales

Datos técnicos	**CM 1241 RS232**
Referencia	6ES7241-1AH32-0XB0
Dimensiones (mm)	30 x 100 x 75
Peso	150 gramos

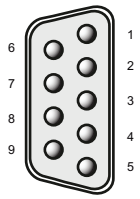
Tabla A-254 Transmisor y receptor

Datos técnicos	**CM 1241 RS232**
Tipo	RS232 (dúplex completo)
Tensión de salida del transmisor	+/- 5 V mín. a R_L = 3K Ω
Tensión de salida del transmisor	+/- 15 V DC máx.
Impedancia de entrada del receptor	3 K Ω mín.
Umbral/sensibilidad del receptor	0,8 V mín. bajo, 2,4 máx. alto histéresis típica: 0,5 V
Tensión de entrada del receptor	+/- 30 V DC máx.
Aislamiento Señal RS 232 a conexión a masa Señal RS 232 a lógica de la CPU	707 V DC (ensayo de tipo)
Longitud de cable, apantallado	10 m máx.
Velocidad de transferencia	300 baudios, 600 baudios, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (valor predeterminado), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Paridad	Sin paridad (valor predeterminado), par, impar, marca (bit de paridad siempre puesto a 1), espacio (bit de paridad siempre puesto a 0)
Número de bits de parada	1 (valor predeterminado), 2
Control de flujo	Hardware, software
Tiempo de espera	0 a 65535 ms

Tabla A-255 Alimentación eléctrica

Datos técnicos	**CM 1241 RS232**
Pérdida de potencia (disipación)	1 W
A partir de +5 V DC	200 mA

Tabla A-256 Conector RS232 (macho)

Pin	Descripción	Conector (macho)	Pin	Descripción
1 DCD	Detección de portadora de datos: Entrada		6 DSR	Equipo de datos listo: Entrada
2 RxD	Datos recibidos de DCE: Entrada		7 RTS	Petición de transmitir Salida
3 TxD	Datos transmitidos a DCE: Salida		8 CTS	Listo para transmitir: Entrada
4 DTR	Terminal de datos disponible: Salida		9 RI	Indicación de timbre (no utilizado)
5 GND	Masa lógica		SHELL	Conexión a masa

A.16.4.3 Datos técnicos del CM 1241 RS422/485

Datos técnicos del CM 1241 RS422/485

Tabla A-257 Especificaciones generales

Datos técnicos	CM 1241 RS422/485
Referencia	6ES7241-1CH32-0XB0
Dimensiones A x A x P (mm)	30 x 100 x 75
Peso	155 gramos

Tabla A-258 Transmisor y receptor

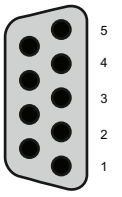
Datos técnicos	CM 1241 RS422/485
Tipo	RS422 o RS485, conector hembra sub-D de 9 polos
Rango de tensión en modo común	-7 V a +12 V, 1 segundo, 3 VRMS continuo
Tensión de salida diferencial del transmisor	2 V mín. a R_L = 100 Ω 1,5 V mín. a R_L = 54 Ω
Terminación y polarización	10K Ω a +5 V en B, pin PROFIBUS 3 10K Ω a GND en A, pin PROFIBUS 8 Opciones de polarización interna disponibles o bien sin polarización interna. En todos los casos se requiere terminación externa; consulte Polarizar y terminar un conector de red RS485 (Página 938) y Configurar RS422 y RS485 en el Manual de sistema del controlador programable S7-1200. (Página 989)
Impedancia de entrada del receptor	5,4K Ω mín. incluyendo terminación
Umbral/sensibilidad del receptor	+/- 0,2 V mín., 60 mV de histéresis típica
Aislamiento Señal RS485 a conexión a masa Señal RS485 a lógica de la CPU	707 V DC (ensayo de tipo)
Longitud de cable, apantallado	máx. 1000 m (en función de la velocidad de transferencia)
Velocidad de transferencia	300 baudios, 600 baudios, 1,2 kbits, 2,4 kbits, 4,8 kbits, 9,6 kbits (valor predeterminado), 19,2 kbits, 38,4 kbits, 57,6 kbits, 76,8 kbits, 115,2 kbits
Paridad	Sin paridad (valor predeterminado), par, impar, marca (bit de paridad siempre puesto a 1), espacio (bit de paridad siempre puesto a 0)

Datos técnicos	CM 1241 RS422/485
Número de bits de parada	1 (valor predeterminado), 2
Control de flujo	Se soporta XON/XOFF para el modo RS422
Tiempo de espera	0 a 65535 ms

Tabla A-259 Alimentación eléctrica

Datos técnicos	CM 1241 RS422/485
Pérdida de potencia (disipación)	1,1 W
A partir de +5 V DC	220 mA

Tabla A-260 Conector RS485 o RS422 (hembra)

Pin	Descripción	Conector (hembra)	Pin	Descripción
1	Masa lógica o de comunicación		6 PWR	+5 V con resistor en serie de 100 ohmios: Salida
2 TxD+ [1]	Conectada para RS422 Sin uso para RS485: salida		7	Sin conexión
3 TxD+ [2]	Señal B (RxD/TxD+): entrada/salida		8 TXD- [2]	Señal A (RxD/TxD-): Entrada/salida
4 RTS [3]	Petición de transmitir (nivel TTL) salida		9 TXD- [1]	Conectada para RS422 Sin uso para RS485: salida
5 GND	Masa lógica o de comunicación		SHELL	Conexión a masa

[1] Los pines 2 (TxD+) y 9 (TxD-) transmiten las señales para RS422.

[2] Los pines 3 (RxD/Tx+) y 8 (RxD/TxD-) transmiten y reciben las señales para RS485. Para RS422, el pin 3 es RxD+ y el pin 8 es RxD-.

[3] RTS es una señal de nivel TTL y se puede emplear para controlar otro dispositivo semidúplex basado en esta señal. Se activa al transmitir datos y se desactiva el resto del tiempo.

A.17 TeleService (TS Adapter y TS Adapter modular)

Los manuales siguientes contienen los datos técnicos del TS Adapter IE Basic y el TS Adapter modular:

- Herramientas de ingeniería para software industrial
  TS Adapter modular
- Herramientas de ingeniería para software industrial
  TS Adapter IE Basic

Para obtener más información sobre este producto y consultar la documentación del producto, visite la página web del catálogo de producto del TS Adapter (https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=TS%20Adapter%20IE%20basic&tab=).

A.18 SIMATIC Memory Cards

Capacidad	Referencia
32 GB	6ES7954-8LT02-0AA0
2 GB	6ES7954-8LP01-0AA0
256 MB	6ES7954-8LL02-0AA0
24 MB	6ES7954-8LF02-0AA0
12 MB	6ES7954-8LE02-0AA0
4 MB	6ES7954-8LC02-0AA0

A.19 Simuladores de entradas

Tabla A-261 Especificaciones generales

Datos técnicos	Simulador de 8 entradas	Simulador de 14 entradas	Simulador de la CPU 1217C
Referencia	6ES7274-1XF30-0XA0	6ES7274-1XH30-0XA0	6ES7274-1XK30-0XA0
Dimensiones A x A x P (mm)	43 x 35 x 23	67 x 35 x 23	93 x 40 x 23
Peso	20 gramos	30 gramos	43 gramos
Entradas	8	14	14
Uso con CPU	CPU 1211C, CPU 1212C	CPU 1214C, CPU 1215C	CPU 1217C

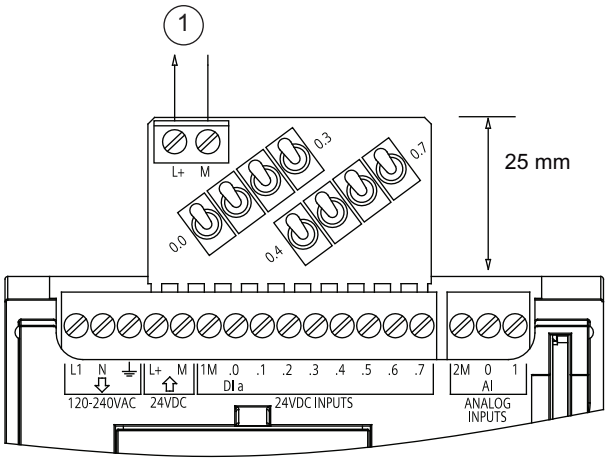
⚠ ADVERTENCIA

Uso seguro de los simuladores de entradas

Estos simuladores de entradas no están aprobados para ser utilizados en ubicaciones peligrosas ("hazardous locations") Class I DIV 2 o Class I Zone 2. Los interruptores pueden producir chispas o explotar si se utilizan en ubicaciones Class I DIV 2 o Class I Zone 2. El uso no autorizado puede ocasionar la muerte o lesiones graves al personal o daños al equipo.

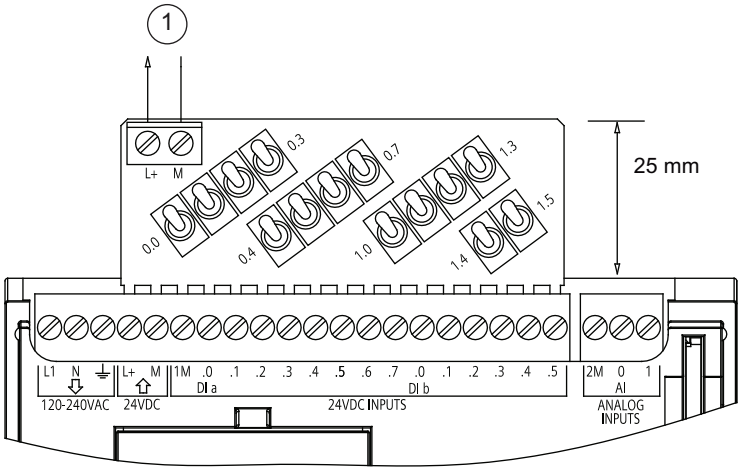
Use estos simuladores de entradas solo en emplazamientos no peligrosos. No los utilice en emplazamientos peligrosos ("hazardous locations") conforme a la clase I, categoría 2 o según la clase I, sección 2.

Simulador de 8 posiciones (6ES7274-1XF30-0XA0)



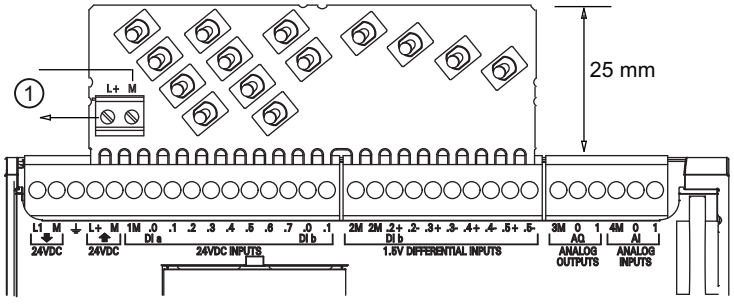
① Salida de alimentación de sensores 24 V DC

Simulador de 14 posiciones (6ES7274-1XF30-0XA0)



① Salida de alimentación de sensores 24 V DC

Simulador de CPU 1217C (6ES7274-1XK30-0XA0)



① Salida de alimentación de sensores 24 V DC

A.20 Módulo del potenciómetro de S7-1200

El módulo del potenciómetro de S7-1200 es un accesorio de la CPU S7-1200. Cada potenciómetro crea una tensión de salida proporcional a la posición del potenciómetro para accionar cada una de las dos entradas analógicas de la CPU de 0 V DC a 10 V DC. Para instalar el potenciómetro:

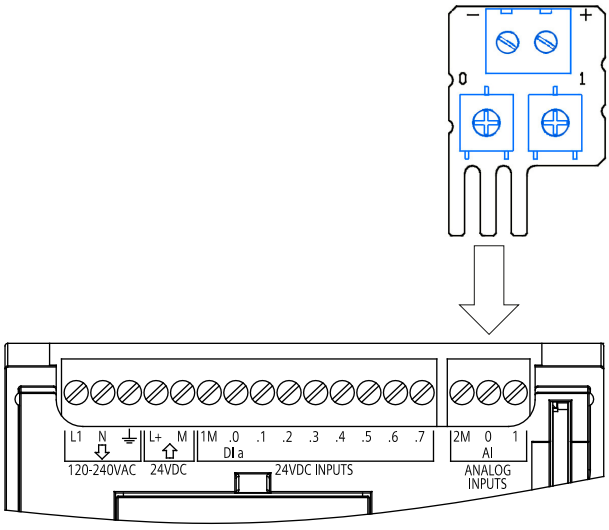
1. Inserte los 'dedos' de la placa de circuito en cualquier bloque de terminales de entrada analógica de la CPU S7-1200 y conecte una fuente de alimentación DC externa al conector de 2 posiciones en el módulo del potenciómetro.
2. Utilice un destornillador pequeño para ajustar los potenciómetros. Gire el potenciómetro hacia la derecha para incrementar la salida de tensión, o bien hacia la izquierda para reducirla.

Nota

Siga las directrices sobre descarga electrostática (ESD) cuando maneje el módulo del potenciómetro de S7-1200.

Datos técnicos	Módulo del potenciómetro de S7-1200
Referencia	6ES7274-1XA30-0XA0
Uso con CPU	Todas las CPU S7-1200
Número de potenciómetros	2
Dimensiones A x A x P (mm)	20 x 33 x 14
Peso	26 gramos
Entrada de tensión suministrada por el usuario en el conector de 2 posiciones[1] (Clase 2, potencia limitada o alimentación de sensores del PLC)	De 16,4 V DC a 28,8 V DC
Longitud de cable (metros)/tipo	<30 m, par trenzado apantallado
Consumo de corriente de entrada	10 mA máx.
Salida de tensión del potenciómetro para entradas analógicas de la CPU S7-1200[1]	De 0 V DC a 10,5 V DC mín.
Aislamiento	Sin aislamiento
Rango de temperatura ambiente	De -20 °C a 60 °C

[1] La estabilidad de la tensión de salida del módulo del potenciómetro depende de la calidad de la entrada de tensión suministrada por el usuario en el conector de 2 posiciones: considérela una tensión de entrada analógica.

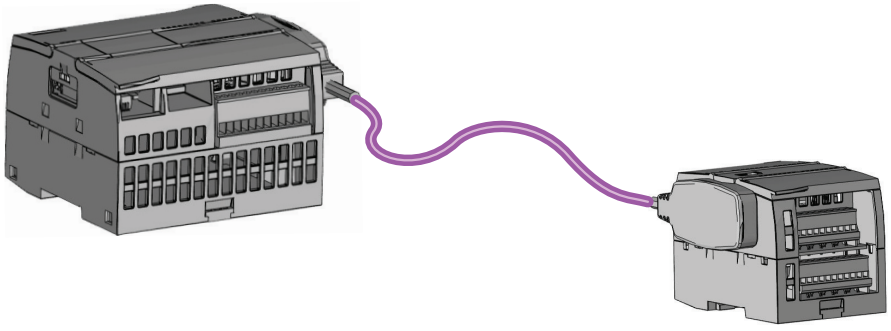


A.21 Cable para módulos de ampliación

Tabla A-262 Cables de ampliación

Datos técnicos	
Referencia	6ES7290-6AA30-0XA0
Longitud del cable	2 m
Peso	200 g

Consulte el apartado de montaje (Página 57) para más información acerca del montaje y desmontaje del cable de ampliación del S7-1200.



A.22 Productos adicionales

A.22.1 Módulo de alimentación PM 1207

El PM 1207 es un módulo de alimentación para el SIMATIC S7-1200. Ofrece las funciones siguientes:

- Entrada 120/230 V AC, salida 24 V DC/2,5A

Para obtener más información sobre este producto y consultar la documentación del producto, visite la página web del catálogo de productos del PM 1207 (https://mall.industry.siemens.com/mall/en/de/Catalog/Product/6EP1332-1SH71).

A.22.2 Módulo de conmutación compacto CSM 1277

El CSM1277 es un Compact switch module Industrial Ethernet. Se puede emplear para multiplicar la interfaz Ethernet del S7-1200 y permitir así una comunicación simultánea con paneles de operador, programadoras u otros controladores. Ofrece las funciones siguientes:

- 4 sockets RJ45 para la conexión a Industrial Ethernet
- 3 bujías polares en la placa de bornes para la conexión de la alimentación externa de 24 V DC encima
- LEDs para mostrar el estado y el diagnóstico de puertos Industrial Ethernet
- Referencia 6GK7277-1AA00-0AA0

Para obtener más información sobre este producto y consultar la documentación del producto, visite la página web del catálogo de productos del CSM 1277 (https://eb.automation.siemens.com/mall/en/de/Catalog/Search?searchTerm=csm%201277&tab=).

A.22.3 Módulo CM CANopen

El módulo CM CANopen es un módulo que se enchufa entre el PLC SIMATIC S7-1200 y cualquier dispositivo que esté ejecutando CANopen. El CM CANopen puede configurarse como maestro o como esclavo. Hay dos CM CANopen modules: el módulo CANopen (referencia 021620-B) y el módulo CANopen (Ruggedized) (referencia 021730-B).

El módulo CANopen ofrece las funciones siguientes:

- posibilidad de conectar 3 módulos por CPU
- conexión de hasta 16 nodos esclavos CANopen
- 256 bytes de entrada y 256 bytes de salida por módulo
- 3 LED que dan información de diagnóstico sobre el módulo, la red y el estado E/S
- almacenamiento de la configuración de red CANopen en el PLC
- capacidad de integración en el catálogo de hardware del paquete de configuración del TIA Portal

- configuración de CANopen mediante CANopen Configuration Studio (incluido) u otra herramienta de configuración externa para CANopen
- cumplimiento de los perfiles de comunicación de CANopen CiA 301 rev. 4.2 y CiA 302 rev. 4.1
- soporte de CAN 2.0A transparente para una gestión de protocolos personalizada
- bloques de función preelaborados disponibles para cualquier programación de PLC en el TIA Portal
- Los módulos CM CANopen incluyen: DSUB con terminales de tornillo para subred, CD con CM CANopen Configuration y cable de configuración USB

Para obtener más información sobre este producto y consultar la documentación correspondiente, visite la página web del catálogo de productos CM CANopen.

A.22.4 Módulo de comunicación RF120C

El RF10C permite conectar directa y fácilmente los sistemas de lectura de códigos y RFID de Siemens a un S7-1200. El lector se conecta al RF120C por medio de una conexión punto a punto. A la izquierda de la CPU es posible conectar un máximo de tres módulos de comunicación a un S7-1200. El módulo de comunicación RF120C se configura a través del TIA Portal. La referencia del módulo de comunicación RF120C es 6GT2002-0LA00.

Para obtener más información sobre este producto y consultar la documentación correspondiente, visite la página web del catálogo de productos del RF120C.

A.22.5 Módulo SM 1238 Energy Meter

El SM 1238 Energy Meter 480 V AC está diseñado para un despliegue a nivel de máquina en un sistema S7-1200. Registra más de 200 mediciones eléctricas y valores energéticos diferentes. Permite gestionar con mayor transparencia las necesidades energéticas de los componentes individuales, desde una planta de producción hasta el nivel de máquina. Empleando los valores medidos proporcionados por el módulo SM 1238 Energy Meter es posible determinar el consumo de energía y la demanda de potencia.

Para obtener más información sobre este producto y consultar la documentación y las especificaciones del producto, visite la página web del catálogo de productos del módulo SM 1238 Energy Meter (https://support.industry.siemens.com/cs/ww/en/view/109483435).

A.22.6 Sistemas de pesaje electrónico SIWAREX

SIWAREX WP231, WP241 y WP251

Los sistemas de pesaje electrónico SIWAREX WP231, WP241 y WP251 pueden utilizarse en el S7-1200. Estos módulos utilizan todas las características de un sistema de automatización moderno, como la comunicación integrada, el funcionamiento y la vigilancia, el sistema de diagnóstico y las herramientas de configuración del TIA Portal.

- SIWAREX WP231 (https://support.industry.siemens.com/cs/ww/es/view/90229056), electrónica de pesaje para calibrador (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 MV/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20 MA)
- SIWAREX WP241 (https://support.industry.siemens.com/cs/ww/es/view/90229063), electrónica de pesaje para cinta (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 M/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20 MA)
- SIWAREX WP251, electrónica de pesaje para procesos por lotes y de llenado (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 MV/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20 MA),

Consulte también

SIWAREX WP251 (https://support.industry.siemens.com/cs/ww/es/view/109481751/en)

Calcular la corriente necesaria

B

La CPU incorpora una fuente de alimentación interna capaz de abastecer la CPU, los módulos de ampliación y otros consumidores de 24 V DC.

Hay cuatro tipos de módulos de ampliación, a saber:

- Los módulos de señales (SM) se montan a la derecha de la CPU. Toda CPU permite conectar un número máximo posible de módulos de señales, sin considerar la corriente disponible.
  - La CPU 1214C, la CPU 1215C y la CPU 1217C permiten conectar 8 módulos de señales.
  - La CPU 1212C permite conectar 2 módulos de señales.
  - La CPU 1211C no permite conectar módulos de señales.
- Los módulos de comunicación (CM) se montan a la izquierda de la CPU. Se permiten como máximo 3 módulos de comunicación para cualquier CPU, sin considerar la corriente disponible.
- Las Signal Boards (SB), placas de comunicación (CB) y Battery Boards (BB) se montan en el lado superior de la CPU. Se permite como máximo 1 Signal Board, Communication Board o Battery Board para cualquier CPU.

Utilice la información siguiente como guía para determinar cuánta energía (o corriente) puede suministrar la CPU a la configuración.

Toda CPU suministra alimentación tanto de 5 V DC como de 24 V DC:

- La CPU suministra 5 V DC a los módulos de ampliación cuando son conectados. Si el consumo de 5 V DC de los módulos de ampliación excede la corriente que ofrece la CPU, habrá que desconectar tantos módulos de ampliación como sea necesario para no excederla.
- Toda CPU incorpora una fuente de alimentación de sensores de 24 V DC que puede suministrar 24 V DC a las entradas locales, o bien a las bobinas de relé de los módulos de ampliación. Si el consumo de 24 V DC excede la corriente disponible de la CPU, es posible agregar una fuente de alimentación externa de 24 V DC para suministrar 24 V DC a los módulos de ampliación. La alimentación de 24 V DC debe conectarse manualmente a las entradas o bobinas de relé.

⚠ ADVERTENCIA

Si se conecta una fuente de alimentación externa de 24 V DC en paralelo con la alimentación DC de sensores, podría surgir un conflicto entre ambas fuentes, ya que cada una intenta establecer su propio nivel de tensión de salida.

Este conflicto puede reducir la vida útil u ocasionar la avería inmediata de una o ambas fuentes de alimentación y, en consecuencia, el funcionamiento imprevisible del sistema PLC. El funcionamiento imprevisible puede producir la muerte, lesiones corporales graves y/o daños materiales.

La fuente de alimentación DC de sensores de la CPU y cualquier fuente de alimentación externa deben alimentar diferentes puntos. Se permite una conexión común de los cables neutros.

Algunos puertos de entrada de alimentación de 24 V DC del sistema PLC están interconectados y tienen un circuito lógico común que conecta varios bornes M. La fuente de alimentación de 24 V DC de la CPU, la entrada de alimentación de las bobinas de relé de un SM, o bien una fuente de alimentación analógica sin aislamiento galvánico son ejemplos de circuitos interconectados si no tienen aislamiento galvánico según las hojas de datos técnicos. Todos los bornes M sin aislamiento galvánico deben conectarse al mismo potencial de referencia externo.

⚠ ADVERTENCIA

Si los bornes M sin aislamiento galvánico se conectan a diferentes potenciales de referencia, circularán corrientes indeseadas que podrían averiar o causar reacciones inesperadas en el PLC y los equipos conectados.

Las averías o reacciones inesperadas podrían producir la muerte, lesiones corporales graves y/o daños materiales.

Asegúrese que todos los bornes M sin aislamiento galvánico de un sistema PLC están conectados al mismo potencial de referencia.

Para más información sobre la corriente disponible de las distintas CPUs y el consumo de corriente de los módulos de señales, consulte los datos técnicos (Página 1245).

Nota

Si se excede la corriente disponible de la CPU, es posible que no se pueda conectar el número máximo de módulos soportado.

Ejemplo de corriente disponible

El ejemplo siguiente muestra el cálculo del consumo de corriente de una configuración que incluye una CPU 1214C AC/DC/relé, una SB 1223 de 2 entradas de 24 V DC / 2 salidas de 24 V DC, un CM 1241, tres SM 1223 de 8 entradas DC y 8 salidas de relé, así como un SM 1221 de 8 entradas DC. Este ejemplo incluye 48 entradas y 36 salidas en total.

Nota

La CPU ya ha asignado la corriente necesaria para accionar las bobinas de relé internas. Por tanto, no es necesario incluir el consumo de corriente de las bobinas de relé internas en el cálculo.

La CPU de este ejemplo suministra suficiente corriente de 5 V DC a los SM, pero la alimentación de sensores no suministra suficiente corriente de 24 V DC a todas las entradas y bobinas de relé de ampliación. Las E/S requieren 456 mA, pero la CPU solo puede suministrar 400 mA. Para esta configuración se necesita una fuente de alimentación adicional de 56 mA como mínimo a 24 V DC para operar todas las entradas y salidas de 24 V DC.

Tabla B-1 Ejemplo de corriente disponible

Corriente disponible de la CPU	**5 V DC**	**24 V DC**
CPU 1214C AC/DC/relé	1600 mA	400 mA
Menos		

Corriente disponible de la CPU	**5 V DC**	**24 V DC**
Requisitos del sistema	**5 V DC**	**24 V DC**
CPU 1214C, 14 entradas	-	14 * 4 mA = 56 mA
1 SB 1223 2 entradas de 24 V DC / 2 salidas de 24 V DC	50 mA	2 * 4 mA = 8 mA
1 CM 1241 RS422/485, 5 V de corriente	220 mA	
3 SM 1223, 5 V de corriente	3 * 145 mA = 435 mA	-
1 SM 1221, 5 V de corriente	1 * 105 mA = 105 mA	-
3 SM 1223, 8 entradas c/u	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, 8 salidas de relé c/u	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, 8 entradas c/u	-	8 * 4 mA = 32 mA
Consumo total	810 mA	456 mA
Igual a		
Balance de corriente	**5 V DC**	**24 V DC**
Balance total de corriente	790 mA	(56 mA)

Forma de calcular la corriente necesaria

Utilice la tabla siguiente para determinar cuánta energía (o corriente) puede suministrar la CPU S7-1200 a la configuración en cuestión. En los datos técnicos (Página 1245) encontrará información sobre la corriente disponible de la CPU y el consumo de los módulos de señales.

Tabla B-2 Cálculo de la corriente necesaria

Corriente disponible de la CPU	**5 V DC**	**24 V DC**
Menos		
Requisitos del sistema	**5 V DC**	**24 V DC**
Consumo total		
Igual a		
Balance de corriente	**5 V DC**	**24 V DC**
Balance total de corriente		

Información de pedido

C

C.1 Módulos CPU

Tabla C-1 CPUs S7-1200

Modelos de CPU		Referencia
CPU 1211C	CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
	CPU 1211C AC/DC/relé	6ES7211-1BE40-0XB0
	CPU 1211C DC/DC/relé	6ES7211-1HE40-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
	CPU 1212C AC/DC/relé	6ES7212-1BE40-0XB0
	CPU 1212C DC/DC/relé	6ES7212-1HE40-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
	CPU 1214C AC/DC/relé	6ES7214-1BG40-0XB0
	CPU 1214C DC/DC/relé	6ES7214-1HG40-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
	CPU 1215C AC/DC/relé	6ES7215-1BG40-0XB0
	CPU 1215C DC/DC/relé	6ES7215-1HG40-0XB0
CPU 1217C	CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0

C.2 Módulos de señales (SM), Signal Boards (SB) y Battery Boards (BB)

Tabla C-2 Módulos de señales (SM)

Módulos de señales		Referencia
Entrada digital	SM 1221 8 entradas de 24 V DC (NPN/PNP)	6ES7221-1BF32-0XB0
	SM 1221 16 entradas de 24 V DC (NPN/PNP)	6ES7221-1BH32-0XB0
Salida digital	SM 1222 8 salidas de 24 V DC (Source)	6ES7222-1BF32-0XB0
	SM 1222 16 salidas de 24 V DC (Source)	6ES7222-1BH32-0XB0
	SM 1222 16 salidas de 24 V DC (Sinking)	6ES7222-1BH32-1XB0
	SM 1222 8 x Relay Output	6ES7222-1HF32-0XB0
	SM 1222 8 x Relay Output (Changeover)	6ES7222-1XF32-0XB0
	SM 1222 16 x Relay Output	6ES7222-1HH32-0XB0

Módulos de señales		**Referencia**
Entrada/salida digital	SM 1223 8 entradas de 24 V DC (NPN/PNP) / 8 salidas de 24 V DC (PNP)	6ES7223-1BH32-0XB0
	SM 1223 16 entradas de 24 V DC (NPN/PNP) / 16 salidas de 24 V DC (PNP)	6ES7223-1BL32-0XB0
	SM 1223 16 entradas de 24 V DC / 16 salidas de 24 V DC (Sinking)	6ES7223-1BL32-1XB0
	SM 1223 8 entradas de 24 V DC (Sink/Source) / 8 salidas de relé	6ES7223-1PH32-0XB0
	SM 1223 16 entradas de 24 V DC (Sink/Source) / 16 salidas de relé	6ES7223-1PL32-0XB0
	SM 1223 8 entradas de 120/230 V AC (Sink/Source) / 8 salidas de relé	6ES7223-1QH32-0XB0
Entrada analógica	SM 1231 4 entradas analógicas	6ES7231-4HD32-0XB0
	SM 1231 8 entradas analógicas	6ES7231-4HF32-0XB0
	SM 1231 4 x entradas analógicas x 16 bits (high feature)	6ES7231-5ND32-0XB0
	SM 1238 Energy Meter 480 V AC	6ES7238-5XA32-0XB0
Salida analógica	SM 1232 2 salidas analógicas	6ES7232-4HB32-0XB0
	SM 1232 4 salidas analógicas	6ES7232-4HD32-0XB0
Entrada/salida analógica	SM 1234 4 entradas analógicas / 2 salidas analógicas	6ES7234-4HE32-0XB0
RTD y termopar	SM 1231 TC 4 x 16 bits	6ES7231-5QD32-0XB0
	SM 1231 TC 8 x 16 bit	6ES7231-5QF32-0XB0
	SM 1231 RTD 4 x 16 bit	6ES7231-5PD32-0XB0
	SM 1231 RTD 8 x 16 bits	6ES7231-5PF32-0XB0
Módulos tecnológicos	SM 1278 4 maestros IO-Link	6ES7278-4BD32-0XB0
	SIWAREX WP231, electrónica de pesaje para calibrador (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 MV/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20 MA)	7MH4960-2AA01
	SIWAREX WP241, electrónica de pesaje para cinta (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 M/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20 MA)	7MH4960-4AA01
	SIWAREX WP251, electrónica de pesaje para procesos por lotes y de llenado (1 canal) para células de carga de galga extensométrica / puentes integrales (1-4 MV/V) para SIMATIC S7-1200, interfaz RS485 y Ethernet, E/S integradas: 4 DI / 4 DO, 1 AO (0/4...20MA),	7MH4960-6AA01

Tabla C-3 Signal Boards (SB) y Battery Boards (BB)

Signal Boards y Battery Boards		**Referencia**
Entrada digital	SB 1221 200 kHz 4 entradas de 24 V DC (PNP)	6ES7221-3BD30-0XB0
	SB 1221 200 kHz 4 entradas de 5 V DC (PNP)	6ES7221-3AD30-0XB0
Salida digital	SB 1222 200 kHz 4 salidas de 24 V DC (NPN/PNP)	6ES7222-1BD30-0XB0
	SB 1222 200 kHz 4 salidas de 5 V DC (NPN/PNP)	6ES7222-1AD30-0XB0

Signal Boards y Battery Boards		Referencia
Entrada/salida digital	SB 1223 2 entradas de 24 V DC (NPN) / 2 salidas de 24 V DC (PNP)	6ES7223-0BD30-0XB0
	SB 1223 200 kHz 2 entradas de 24 V DC (NPN) / 2 salidas de 24 V DC (NPN/PNP)	6ES7223-3BD30-0XB0
	SB 1223 200 kHz 2 entradas de 5 V DC (NPN) / 2 salidas de 5 V DC (NPN/PNP)	6ES7223-3AD30-0XB0
Analógico	SB 1232 de 1 salida analógica	6ES7232-4HA30-0XB0
	SB 1231 de 1 entrada analógica	6ES7231-4HA30-0XB0
	SB 1231 de termopar de 1 entrada analógica	6ES7231-5QA30-0XB0
	SB 1231 de 1 entrada analógica RTD	6ES7231-5PA30-0XB0
Batería	BB 1297 Battery Board (tipo de batería CR1025 no incluida)	6ES7297-0AX30-0XA0

C.3 Comunicación

Tabla C-4 Módulo de comunicación (CM)

Módulo de comunicación (CM)			Referencia
RS232, RS422 y RS485	CM 1241 RS232	RS232	6ES7241-1AH32-0XB0
	CM 1241 RS422/485	RS422/485	6ES7241-1CH32-0XB0
PROFIBUS	CM 1243-5	Maestro PROFIBUS	6GK7243-5DX30-0XE0
	CM 1242-5	Esclavo PROFIBUS	6GK7242-5DX30-0XE0
AS-i Master	CM 1243-2	AS-i Master	3RK7243-2AA30-0XB0

Tabla C-5 Communication Board (CB)

Communication Board (CB)			Referencia
RS485	CB 1241 RS485	RS485	6ES7241-1CH30-1XB0

Tabla C-6 Procesador de comunicaciones (CP)

CP	Interfaz	Referencia
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	Interfaz IE	6GK7243-1BX30-0XE0
CP 1243-8 IRC	Interfaz IE y serie	6GK7243-8RX30-0XE0

Tabla C-7 TeleService

TS Adapter	Referencia
Módulo TS RS232	6ES7972-0MS00-0XA0
Módulo TS Módem	6ES7972-0MM00-0XA0

Tabla C-8 Accesorios

Accesorio			Referencia
Antena	ANT794-4MR	Antena GSM/GPRS	6NH9860-1AA00
	ANT794-3M	Antena plana	6NH9870-1AA00

Tabla C-9 Conectores

Tipo de conector		Referencia
RS485	Salida del cable de 35 grados, conexión de terminal roscado	6ES7972-0BA42-0XA0
	Salida del cable de 35 grados, conexión FastConnect	6ES7972-0BA60-0XA0

C.4 CPUs y módulos de señales de seguridad positiva

Tabla C-10 CPUs de seguridad positiva

Modelos de CPU de seguridad positiva		Referencia
CPU 1212FC	CPU 1212FC DC/DC/DC	6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/relé	6ES7212-1HF40-0XB0
CPU 1214FC	CPU 1214FC DC/DC/DC	6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/relé	6ES7214-1HF40-0XB0
CPU 1215FC	CPU 1215FC DC/DC/DC	6ES7215-1AF40-0XB0
	CPU 1215FC DC/DC/relé	6ES7215-1HF40-0XB0

Tabla C-11 Módulos de señales de seguridad positiva

Módulos de señales de seguridad funcional		Referencia
Entrada digital	SM 1226 F-DI 16 x 24 V DC	6ES7226-6BA32-0XB0
Salida digital	SM 1226 F-DQ 4 x 24 V DC	6ES7226-6DA32-0XB0
	SM 1226 F-DQ 2 x Relay	6ES7226-6RA32-0XB0

C.5 Otros módulos

Tabla C-12 Productos adicionales

Producto		Referencia
Alimentación eléctrica	Fuente de alimentación PM 1207	6EP1332-1SH71
Switch Ethernet	Switch Ethernet CSM 1277 - 4 puertos	6GK7277-1AA10-0AA0
CM CANopen	CANopen para SIMATIC S7-1200	021620-B
	CANopen (Ruggedized) para SIMATIC S7-1200	021730-B
RF120C	Módulo de comunicación RF120C	6GT2002-0LA00

C.6 Memory Cards

Tabla C-13 Memory Cards

SIMATIC Memory Cards	Referencia
SIMATIC MC 32 GB	6ES7954-8LT03-0AA0
SIMATIC MC 2 GB	6ES7954-8LP02-0AA0
SIMATIC MC 256 MB	6ES7954-8LL03-0AA0
SIMATIC MC 24 MB	6ES7954-8LF03-0AA0
SIMATIC MC 12 MB	6ES7954-8LE03-0AA0
SIMATIC MC 4 MB	6ES7954-8LC03-0AA0

C.7 Dispositivos HMI Basic

Tabla C-14 Dispositivos HMI

Paneles HMI Basic	Referencia
KTP400 Basic (Mono, 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 Repuestos y hardware adicional

Tabla C-15 Cables de ampliación, simuladores y escuadras finales

Producto		Referencia
Cable para módulos de ampliación	Cable de ampliación E/S, 2 m	6ES7290-6AA30-0XA0
Simulador E/S	Simulador (1211C/1212C - 8 posiciones)	6ES7274-1XF30-0XA0
	Simulador (1214C/1215C - 14 posiciones)	6ES7274-1XH30-0XA0
	Simulador, CPU 1217C	6ES7274-1XK30-0XA0
Módulo del potenciómetro	Módulo del potenciómetro de S7-1200	6ES7274-1XA30-0XA0
Descarga de tensión Ethernet	Descarga de tensión RJ45 para puerto individual, 10/100 Mbits/s	6ES7290-3AA30-0XA0
	Descarga de tensión RJ45 para dos puertos, 10/100 Mbits/s	6ES7290-3AB30-0XA0

Producto		Referencia
Kit de tapas de repuesto	CPU 1211C/1212C	6ES7291-1AA30-0XA0
	CPU 1214C	6ES7291-1AB30-0XA0
	CPU 1215C	6ES7291-1AC30-0XA0
	CPU 1217C	6ES7291-1AD30-0XA0
	Módulo de señales, 45 mm	6ES7291-1BA30-0XA0
	Módulo de señales, 70 mm	6ES7291-1BB30-0XA0
	Módulo de comunicación (para usar con módulos 6ES72xx-xxx32-0XB0 y 6ES72xx-xxx30-0XB0)	6ES7291-1CC30-0XA0
Escuadra final	Escuadra final de termoplástico, 10 MM	8WA1808
	Escuadra final de acero, 10,3 MM	8WA1805

Sustitución del conector del bloque de bornes

Es importante utilizar el bloque de bornes correcto para cada módulo. Consulte las tablas inferiores y las especificaciones del módulo para determinar la sustitución adecuada del bloque de bornes.

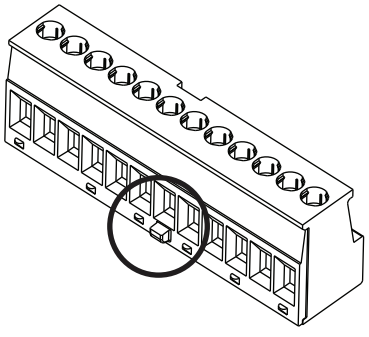
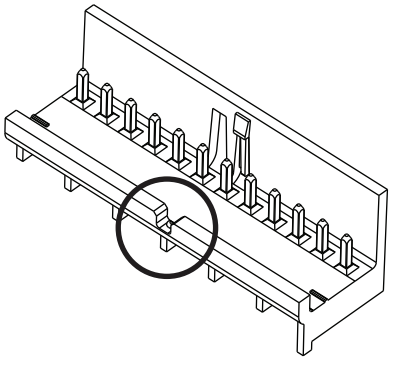
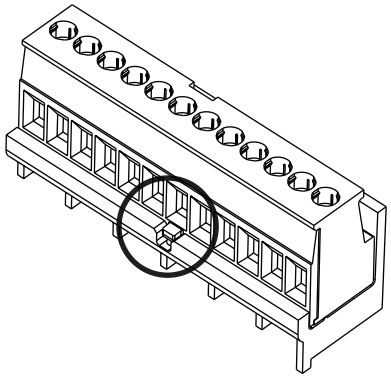
Nota

Bloques de bornes desenchufables codificados

Es imprescindible cablear siempre correctamente los PLC para garantizar la seguridad y el funcionamiento adecuado.

A la hora de sustituir el bloque de bornes de la CPU o el SM, es importante utilizar el bloque de bornes correcto y la fuente de cableado adecuada para el módulo.

El diseño mecánico impide que se enchufe un bloque de bornes de alta tensión en un módulo de baja tensión, o que se enchufe un bloque de bornes de tensión especial en un módulo de tensión normal. Algunos bloques de bornes tienen la codificación orientada a la izquierda, otros a la derecha y otros en el centro.

		
Bloque de bornes desenchufable (se muestra un ejemplo con codificación)	Codificación en el dispositivo	La codificación en el dispositivo solo encaja en el bloque de bornes desenchufable adecuado

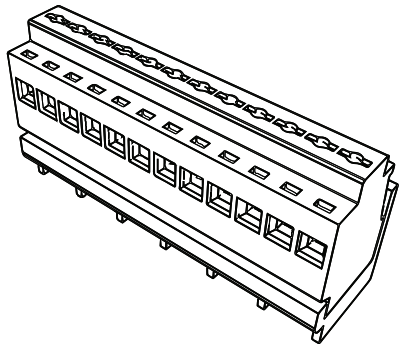
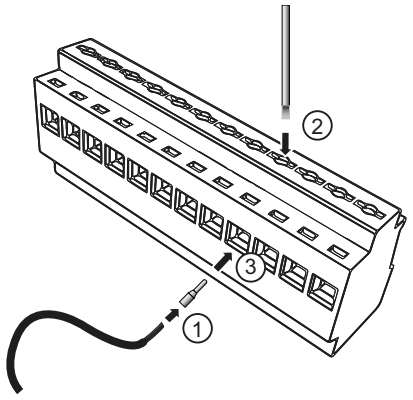
	
Bloque de bornes de inserción directa (push in)	Proceda del siguiente modo: 1. Inserte el hilo y el conector engastado. 2. Presione la membrana utilizando un destornillador pequeño para abrir la ranura. 3. Inserte el conector engastado en la ranura abierta del bloque de bornes.

Tabla C-16 Kits de repuesto de bloques de bornes para CPU S7-1200 V4.0 y superiores

Si tiene una CPU S7-1200 V4.0 o superior (referencia)	**Use este kit de repuesto de bloques de bornes (4/pq)**		
	Referencia del bloque de bornes (de tornillo)	**Referencia del bloque de bornes de inserción directa (push in) equivalente**	**Descripción del bloque de bornes**
CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8 pines, estañados
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14 pines, estañados
CPU 1211C DC/DC/relé (6ES7211-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8 pines, estañados, codificado
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14 pines, estañados
CPU 1211C AC/DC/Relay (6ES7211-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8 pines, estañados, codificado
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14 pines, estañados, codificado
CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8 pines, estañados
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14 pines, estañados
CPU 1212C DC/DC/Relay (6ES7212-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8 pines, estañados, codificado
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14 pines, estañados

Si tiene una CPU S7-1200 V4.0 o superior (referencia)	Use este kit de repuesto de bloques de bornes (4/pq)		
	Referencia del bloque de bornes (de tornillo)	Referencia del bloque de bornes de inserción directa (push in) equivalente	Descripción del bloque de bornes
CPU 1212C AC/DC/Relay (6ES7212-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8 pines, estañados, codificado
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14 pines, estañados, codificado
CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12 pines, estañados
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1214C DC/DC/Relay (6ES7214-1HG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12 pines, estañados, codificado
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1214C AC/DC/Relay (6ES7214-1BG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12 pines, estañados, codificado
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20 pines, estañados, codificado
CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12 pines, estañados
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1215C DC/DC/Relay (6ES7215-1HG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12 pines, estañados, codificado
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1215C AC/DC/Relay (6ES7215-1BG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12 pines, estañados, codificado
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20 pines, estañados, codificado
CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AK30-0XA0	6ES7292-2AK30-0XA0	10 pines, estañados
	6ES7292-1AR30-0XA0	6ES7292-2AR30-0XA0	16 pines, estañados
	6ES7292-1AT30-0XA0	6ES7292-2AT30-0XA0	18 pines, estañados

Tabla C-17 Kits de repuesto de bloques de bornes para SM S7-1200 V3.2 y superiores

Si tiene un SM S7-1200 V3.2 o superior (referencia)	**Use este kit de repuesto de bloques de bornes (4/pq)**		
	Referencia del bloque de bornes (de tornillo)	**Referencia del bloque de bornes de inserción directa (push in) equivalente**	**Descripción del bloque de bornes**
SM 1221 DI 8 x DC (6ES7221-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1222 DQ 8 x DC (6ES7222-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1222 DQ 8 x Relay (6ES7222-1HF32-0XB0)	6ES7292-1AG40-0XA1	6ES7292-2AG40-0XA1	7 pines, estañados, codificado izquierda
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) para entrada de tensión (arriba)	6ES7292-1AG40-0XA2	6ES7292-2AG40-0XA2	7 pines, estañado, codificado centro
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) para entrada de intensidad (abajo)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañado
SM 1231 AI 4 x 13 bits (6ES7231-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1232 AQ 2 x 14 bits (6ES7232-4HB32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1231 AI 4 x TC (6ES7231-5QD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1231 AI 4 x 16 bits (6ES7231-5ND32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1221 DI 16 x DC (6ES7221-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1222 DQ 16 x DC (6ES7222-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1222 DQ 16 x Relay (6ES7222-1HH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7 pines, estañados, codificado derecha
SM 1223 DI 8 x DC/DQ 8 x DC (6ES7223-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1223 8 x DC/8 x Relay (6ES7223-1PH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7 pines, estañados, codificado derecha
SM 1223 8 x AC/8 x Relay (6ES7223-1QH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7 pines, estañados, codificado derecha
SM 1234 AI 4 / AQ 2 (6ES7234-4HE32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1231 AI 8 x 13 bits (6ES7231-4HF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1232 AQ 4 x 14 bits (6ES7232-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados

Si tiene un SM S7-1200 V3.2 o superior (referencia)	Use este kit de repuesto de bloques de bornes (4/pq)		
	Referencia del bloque de bornes (de tornillo)	Referencia del bloque de bornes de inserción directa (push in) equivalente	Descripción del bloque de bornes
SM 1231 AI 4 x RTD (6ES7231-5PD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1231 AI 8 x TC (6ES7231-5QF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7 pines, dorados
SM 1278 IO LINK (6ES7278-4BD32 0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7 pines, estañados
SM 1222 DQ 8 x Relay (Changeover) (6ES7222-1XF32-0XB0)	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11 pines, estañados, codificado
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11 pines, estañados
SM 1223 DI 16 x DC/DQ 16 x Relay (6ES7223-1PL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11 pines, estañados
	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11 pines, estañados, codificado
SM 1231 AI 8 x RTD (6ES7231-5PF32-0XB0)	6ES7292-1BL30-0XA0	6ES7292-2BL30-0XA0	11 pines, dorados

Tabla C-18 Kits de repuesto de bloques de bornes para SB, CB y BB del S7-1200

Si tiene SB, CB o BB del S7-1200 (referencia)	Use este kit de repuesto de bloques de bornes (4/pq)	
	Referencia del bloque de bornes (de tornillo)	Descripción del bloque de bornes
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)	6ES7292-1BF30-0XA0	6 pines
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)		
SB 1221 DI 4 x 24 V DC (6ES7221-3BD30-0XB0)		
SB 1222 DQ 4 x 5 V DC (6ES7222-1AD30-0XB0)		
SB 1222 DQ 4 x 24 V DC (6ES7222-1BD30-0XB0)		
SB 1223 DI 2x24 V DC/DQ 2x24 V DC (6ES7223-0BD30-0XB0)		
SB 1223 DI 2x5 V DC / DQ 2x5 V DC (6ES7223-3AD30-0XB0)		
SB 1223 DI 2x24 V DC / DQ 2x24 V DC (6ES7223-3BD30-0XB0)		
SB 1231 AI 1 x 12 bits (6ES7231-4HA30-0XB0)		
SB 1231 AI 1 x RTD (6ES7231-5PA30-0XB0)		
SB 1231 AI 1 x TC (6ES7231-5QA30-0XB0)		
SB 1232 AQ 1x12 bits (6ES7232-4HA30-0XB0)		
CB 1231 RS485 (6ES7241-1CH30-1XB0)		
BB 1297 Battery (6ES7297-0AX30-0XA0)		

Tabla C-19 Kit de repuesto de bloques de bornes para CPU de seguridad

Si tiene una CPU de seguridad positiva (referencia)	**Use este kit de repuesto de bloques de bornes (4/pq)**		
	Referencia del bloque de bornes	**Referencia equivalente del bloque de bornes de inserción directa (push in)**	**Descripción del bloque de bornes**
CPU 1212FC DC/DC/DC (6ES7212-1AF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	12 pines, estañados
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	20 pines, estañados
CPU 1212FC DC/DC/Relay (6ES7212-1HF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	3 pines, dorados
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	12 pines, estañados, codificado
CPU 1214FC DC/DC/DC (6ES7214-1AF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12 pines, estañados
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1214FC DC/DC/relé (6ES7214-1HF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12 pines, estañados, codificado
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1215FC DC/DC/DC (6ES7215-1AF40 0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12 pines, estañados
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados
CPU 1215FC DC/DC/relé (6ES7215-1HF40 0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6 pines, dorados
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	2 pines, estañados, codificado
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20 pines, estañados

Tabla C-20 Kit de repuesto de bloques de bornes para módulos de señales de seguridad

Si tiene un módulo de señales de seguridad positiva (referencia)	**Utilice este kit de repuesto de bloques de bornes (4/pq)**		
	Referencia del bloque de bornes	**Referencia equivalente del bloque de bornes de inserción directa (push in)**	**Descripción del bloque de bornes**
SM 1226 F-DI (6ES7226-6BA32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11 pines, estañados
SM 1226 F-DQ (6ES7226-6DA32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11 pines, estañados
SM 1226 relé F (6ES7226-6RA32-0XB0)	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11 pines, estañados, codificado

C.9 Software de programación

Tabla C-21 Software de programación

Software SIMATIC		Referencia
Software de programación	STEP 7 Basic V17	6ES7822-0AA06-0YA5
	STEP 7 Professional V17	6ES7822-1AA06-0YA5
Software de visualización	WinCC Basic V17	6AV2100-0AA06-0AA5
	WinCC Comfort V17	6AV2101-0AA06-0AA5
	WinCC Advanced V17	6AV2102-0AA06-0AA5
	WinCC Professional 512 PowerTags V17	6AV2103-0DA06-0AA5
	WinCC Professional 4096 PowerTags V17	6AV2103-0HA06-0AA5
	WinCC Professional máx. PowerTags V17	6AV2103-0XA06-0AA5

C.10 Licencias OPC UA

Tabla C-22 Licencias OPC UA para S7-1200

Licencias OPC UA		Referencia
Licencia SIMATIC	SIMATIC OPC UA S7-1200 Basic DVD	6ES7823-0BA00-2BA0
	SIMATIC OPC UA S7-1200 Basic DL	6ES7823-0BE00-2BA0

Cambio de dispositivo y compatibilidad de los repuestos

D

D.1 Sustituir una CPU con protección de datos de configuración confidenciales del PLC

La asignación de contraseñas para proteger los datos de configuración confidenciales del PLC también tiene un impacto en recambio de piezas.

Reglas para el recambio de piezas

Observe las siguientes reglas para el recambio de piezas:

Configuración de la CPU de repuesto en TIA Portal

- Si es posible, utilice como CPU de repuesto una CPU que no tenga una configuración de proyecto o una contraseña configurada para la protección de los datos de configuración confidenciales del PLC.
  Ventaja: Es posible cargar el proyecto en la CPU de repuesto sin preparativos adicionales.
- Si la CPU de repuesto ya se ha configurado, debe restablecer la configuración de fábrica de la CPU (Página 1206) y seleccionar estas opciones:
  - Borrar contraseña para la protección de datos de configuración confidenciales del PLC
  - Formatear Memory Card si la CPU tiene una SIMATIC Memory Card

Sustituir una CPU con datos de configuración en una SIMATIC Memory Card

- Si en el proyecto **no** se ha asignado ninguna contraseña a una CPU para proteger los datos de configuración confidenciales del PLC, es posible insertar la SIMATIC Memory Card de la CPU que se ha de sustituir en una CPU por estrenar sin preparativos adicionales.
- Si la CPU de repuesto ya se ha configurado con una contraseña, primero debe restablecer la configuración de fábrica de esa CPU (Página 1206) con la opción "Borrar contraseña para la protección de datos de configuración confidenciales del PLC".
- Si se ha asignado la misma contraseña para la protección de datos de configuración confidenciales del PLC a un grupo de CPU, también es posible asignar la contraseña de grupo a la CPU de repuesto desde la configuración de dispositivos (Página 161) en TIA Portal. En ese caso, por ejemplo, enchufe una SIMATIC Memory Card con el proyecto actual en la CPU y póngala en funcionamiento sin gestión adicional de la contraseña.
- Si asigna contraseñas distintas a cada CPU en el proyecto, establezca una conexión online (Página 1201) y luego establezca la contraseña para la protección de datos confidenciales de la CPU de repuesto con las herramientas Online y diagnóstico.Seleccione "Establecer contraseña de protección de datos de configuración del PLC" (Página 1205) desde las funciones online.

D.2 Sustitución de una CPU V3.0 por una CPU V4.x

Para actualizar una CPU V3.0 a una CPU V4.x es preciso sustituir el hardware de la CPU. No es posible actualizar una CPU V3.0 a una CPU V4.x actualizando el firmware.

Seguidamente, en el proyecto de STEP 7 es posible sustituir la CPU V3.0 por una CPU V4.x (Página 152) y usar el proyecto de STEP 7 que se ha diseñado para la CPU V3.0.

Al reemplazar una CPU V3.0 por una CPU V4.x también es posible comprobar si hay actualizaciones de firmware (Página 132) para los módulos de comunicación y señales conectados y aplicarlas.

Nota

No es posible sustituir dispositivos de la versión V4.x por dispositivos de la versión V3.0 en STEP 7

Es posible sustituir una CPU V3.0 por una CPU V4.x, pero no sustituir una CPU V4.x por una CPU V3.0 una vez que se ha descargado la configuración. Si desea ver o utilizar de cualquier otra forma el proyecto existente de STEP 7 V3.0, haga una copia de seguridad del mismo antes de cambiar el dispositivo.

Si todavía no ha descargado la configuración del dispositivo reemplazado puede deshacerla. Sin embargo, una vez descargada no se podrá deshacer el cambio de V3.0 a V4.x.

Deben tenerse en cuenta algunos cambios en la configuración y el funcionamiento entre las dos versiones de CPU:

Actualizar proyectos de STEP 7

No es posible actualizar proyectos de STEP 7 V11 o V12 directamente a STEP 7 V15. Primero hay que actualizar dichos proyectos a STEP 7 V13 SP1 o STEP 7 V13 SP2. A continuación, utilice ese proyecto como base para actualizar a STEP 7 V15.

⚠ ADVERTENCIA

Riesgos al copiar y pegar la lógica del programa de versiones anteriores de STEP 7

El hecho de copiar la lógica del programa de una versión anterior de STEP 7, como STEP 7 V12, a STEP 7 V15 puede causar un comportamiento impredecible en la ejecución del programa o errores de compilación. Las diferentes versiones de STEP 7 implementan los elementos del programa de forma distinta. El compilador no siempre detecta las diferencias si los cambios se realizan mediante copiar y pegar desde una versión anterior a STEP 7 V15. La ejecución de una lógica del programa impredecible puede causar la muerte o lesiones graves si no se corrige el programa.

Si emplea una lógica de programa de una versión de STEP 7 anterior a STEP 7 V15, actualice siempre todo el proyecto a STEP 7 V15. Después podrá copiar, cortar y pegar la lógica del programa según sea necesario. En STEP 7 V15 es posible abrir un proyecto de STEP 7 V 13 SP1 o posterior. Seguidamente, STEP 7 ejecuta las conversiones de compatibilidad necesarias y actualiza el programa de forma correcta. Estas actualizaciones y correcciones son necesarias para una ejecución y compilación debidas del programa. Si el proyecto es anterior a STEP 7 V13 SP1, deberá actualizarse el proyecto a STEP 7 V15.

Bloques de organización

Es posible configurar que la ejecución de los OB sea con o sin interrupciones (Página 87). Para proyectos de CPU V3.0, STEP 7 determina por defecto que todos los OB sean sin interrupciones.

STEP 7 pone todas las prioridades de los OB (Página 87) a los valores que tenían en el proyecto de STEP 7 de la CPU V3.0.

Es posible cambiar la configuración de las interrupciones o la prioridad más adelante.

La información de arranque del OB de alarma de diagnóstico (Página 80) se refiere al submódulo como un todo si no hay eventos de diagnóstico pendientes.

Protección por contraseña de la CPU

STEP 7 ajusta el nivel de protección por contraseña (Página 163) para la CPU V4.x al nivel equivalente al de la CPU V3.0 y asigna la contraseña de V3.0 a la contraseña "Acceso completo (sin protección)" de la CPU V4.x:

Nivel de protección V3.0	Nivel de acceso V4.x
Sin protección	Acceso completo (sin protección)
Protección contra escritura	Acceso de lectura
Protección contra escritura/lectura	Acceso HMI

Tenga en cuenta que el nivel de acceso V4.x "Sin acceso (protección completa)" no existía en la versión V3.0.

Servidor web

Si usa páginas web definidas por el usuario en su proyecto V3.0, guárdelas en la carpeta de instalación del proyecto, dentro de la subcarpeta "UserFiles\Webserver" antes de actualizar el proyecto. Si guarda las páginas definidas por el usuario en este emplazamiento, cuando guarde el proyecto de STEP 7 también se guardarán las páginas web definidas por el usuario.

Si sustituye una CPU V3.0 por una CPU V4.x, la configuración de proyecto del servidor web (Página 846) para la activación del servidor web y la configuración HTTPS serán las mismas que en V3.0. A continuación puede configurar usuarios, privilegios, contraseñas (Página 848) e idiomas (Página 846) de la forma necesaria para utilizar el servidor web. Si no configura usuarios con privilegios adicionales estará limitado a lo que se ve en las páginas web estándar (Página 857). La CPU S7-1200 V4.x no soporta el usuario y la contraseña "admin" preconfigurados en versiones anteriores.

La página DataLogs del servidor web de S7-1200 V3.0 ofrecía una operación de "Download and Clear". La página Navegador de archivos (Página 889) del servidor web V4.x, desde la que se accede a los registros de datos o DataLogs, ya no soporta esta función. En su lugar, el servidor web ofrece la posibilidad de descargar, cambiar el nombre y borrar archivos de registros de datos.

Incompatibilidad de la tarjeta de transferencia

No se puede utilizar una tarjeta de transferencia (Página 120) V3.0 para transferir un programa V3.0 a una CPU V4.x. Hay que abrir el proyecto V3.0 en STEP 7, sustituir el dispositivo por una CPU V4.x (Página 152) y cargar el proyecto de STEP 7 en la CPU V4.x. Después de haber migrado el proyecto a V4.x, se puede realizar una tarjeta de transferencia V4.x para posteriores transferencias del programa.

Comunicación GET/PUT

En su ajuste predeterminado, las CPU S7-1200 V3.0 tenían activada la comunicación GET/PUT. Al sustituir una CPU V3.0 por una CPU V4.x (Página 152) aparece un mensaje en la sección de información de compatibilidad indicando que GET/PUT está activado.

Soporte de Motion control

Las CPU S7-1200 V4.x no soportan las librerías de Motion Control de V1.0 y V2.0. Si se cambia un dispositivo para un proyecto de STEP 7 con librerías de Motion Control de V1.0 o V2.0, durante la compilación se reemplazarán las instrucciones de la librería de Motion Control de V1.0 o V2.0 por instrucciones de Motion Control compatibles de la versión V3.0.

Si se sustituye una CPU V3.0 por una CPU V4.x para un proyecto de STEP 7 que contenga dos versiones diferentes de instrucciones de Motion Control (V3.0 y V5.0), durante la compilación se reemplazarán las instrucciones de Motion Control compatibles con V5.0.

Al sustituir una CPU V3.0 por una CPU V4.x, la versión del objeto tecnológico (TO) de Motion Control no cambia automáticamente de la versión V3.0 a la V5.0. Si desea actualizar a versiones más actuales, vaya al árbol de instrucciones y seleccione la versión necesaria de S7-1200 Motion Control para el proyecto tal y como se muestra en la tabla siguiente:

Versión de CPU	Versiones de Motion Control permitidas
V4.3 (Motion Control V5.0)	V6.0, V5.0, V4.0 o V3.0
V4.2.x (Motion Control V5.0)	V6.0, V5.0, V4.0 o V3.0
V4.1 (Motion Control V5.0)	V5.0, V4.0 o V3.0
V4.0 (Motion Control V4.0)	V4.0 o V3.0
V3.0 (Motion Control V3.0)	V3.0

La estructura TO de las versiones de Motion Control V3.0 y V5.0 difiere. Todos los bloques asociados también cambian. Las interfaces de bloques, las tablas de observación y los traces se actualizan a la nueva estructura de Motion Control V5.0. Encontrará las diferencias entre los parámetros de Motion Control de la CPU V3.0 y la CPU V4.x para ejes en las dos tablas siguientes:

CPU V3.0 (Motion Control V3.0)	CPU V4.x (Motion Control 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

CPU V3.0 (Motion Control V3.0)	CPU V4.x (Motion Control V5.0)
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	Posición
MotionStatus.Velocity	Velocidad
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

El único parámetro de la tabla de comandos cuyo nombre cambia es la matriz con los comandos:

V3.0	V4.x
Config.Command[]	Command[]

Nota: La matriz "Command[]" es un UDT del tipo "TO_CmdTab_Config_Command" en V3.0 y "TO_Struct_Command" en V4.x.

Cambios en las instrucciones

Las instrucciones siguientes presentan cambios en los parámetros o en su comportamiento:

- RDREC y WRREC (Página 370)
- CONV (Página 284)

Comunicación con HMI Panels

Si tenía uno o varios HMI Panels (Página 32) conectados a la CPU S7-1200 V3.0, la comunicación con la CPU S7-1200 V4.x dependerá del tipo de comunicación que utilice y de la versión de firmware del panel HMI. Vuelva a compilar el proyecto y descárguelo en la CPU y el HMI y/o actualice el firmware HMI.

Requisitos para volver a compilar bloques de programa

Después de sustituir una CPU V3.0 por una CPU V4.x es necesario volver a compilar todos los bloques de programa antes de poder cargarlos en la CPU V4.x. Además, si alguno de los bloques tiene protección de know-how (Página 167) o protección de copia ligada al número de serie del PLC (Página 168), deberá eliminar la protección antes de compilar y cargar los bloques. (Sin embargo, no es necesario desactivar la protección de copia ligada a una Memory Card.) Una vez que la compilación ha finalizado correctamente, puede volver a configurar la protección de know-how y/o la protección de copia vinculada al número de serie del PLC. Tenga en cuenta que si el proyecto incluye bloques con protección de know-how suministrados por un OEM (Original Equipment Manufacturer) deberá ponerse en contacto con el OEM para que le suministre versiones V4.x de dichos bloques.

Por norma general, Siemens recomienda volver a compilar la configuración hardware y el software en STEP 7 y cargarlos en todos los dispositivos del proyecto después de cambiar un dispositivo. Corrija todos los errores que se encuentren al compilar el proyecto y vuelva a compilarlo hasta que no haya errores. Seguidamente, puede descargar el proyecto a la CPU V4.x.

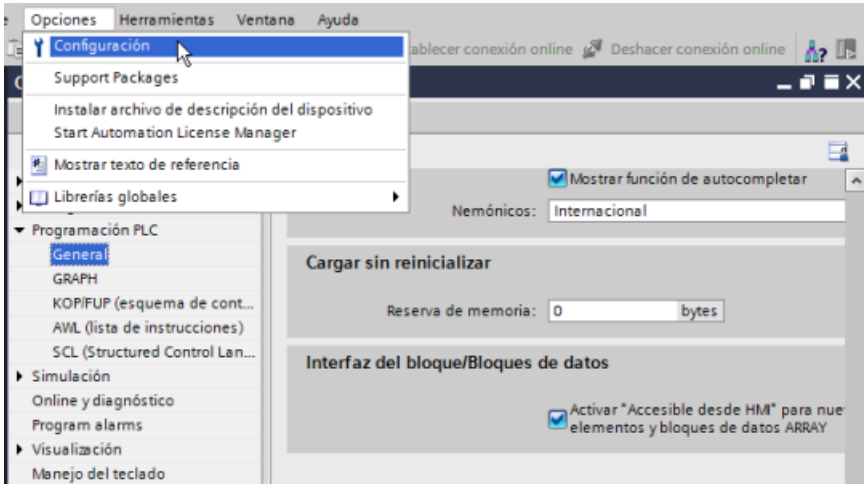
Los proyectos S7-1200 V3.0 pueden no ser aptos para CPU S7-1200 V4.x

En S7-1200 V4.0 y superior se ha agregado un área reservada de 100 bytes para cada DB para permitir la descarga sin reinicializar.

Se puede eliminar el área de reserva de 100 bytes de los DB antes de intentar descargar un proyecto V3.0 en una CPU V4.x.

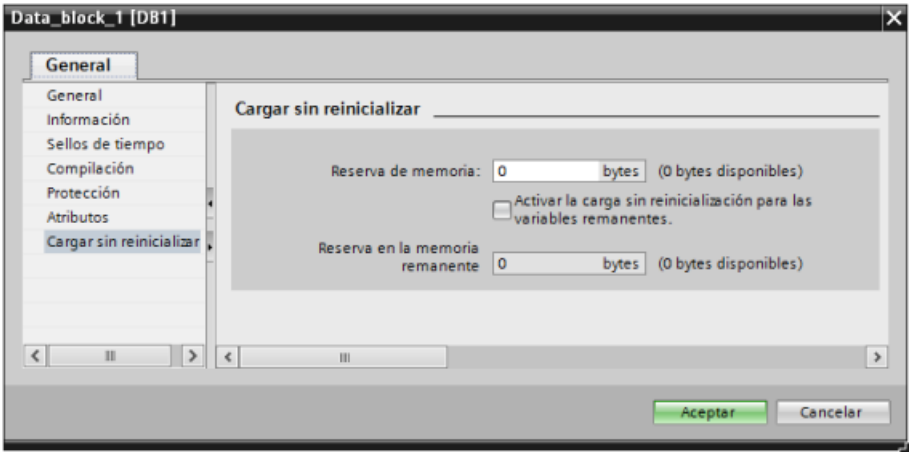
Para eliminar el área de reserva de 100 bytes siga los pasos siguientes antes de cambiar el dispositivo:

1. Desde el menú principal del TIA Portal, elija el comando Opciones > Configuración.
2. Desde el árbol de navegación, abra Programación PLC > Nodo general.
3. En el área "Cargar sin reinicializar", ajuste la reserva de memoria a 0 bytes.



Si ya ha sustituido el dispositivo, tiene que eliminar la reserva de 100 bytes individualmente de cada bloque:

1. Desde el árbol del proyecto, haga clic con el botón derecho del ratón en un bloque de datos de la carpeta Bloques de programa y elija Propiedades en el menú contextual.
2. En el cuadro de diálogo de las propiedades del bloque de datos, elija el nodo "Cargar sin reinicializar".
3. Ajuste la reserva de memoria a 0 bytes.
4. Repita el procedimiento para cada bloque de datos del proyecto.



D.3 Kits de repuesto de bloques de terminales para S7-1200 V3.0 e inferiores

Tabla D-1 Kits de repuesto de bloques de terminales para CPU S7-1200 V3.0 e inferiores

Si tiene una CPU S7-1200 V3.0 o inferior (referencia)	**Utilice este kit de repuesto de bloques de terminales (4/pq)**	
	Referencia del bloque de terminales	**Descripción del bloque de terminales**
CPU 1211C DC/DC/DC (6ES7211-1AE31-0XB0)	6ES7292-1BC30-0XA0	3 pines, dorados
CPU 1211C AC/DC/relé (6ES7211-1BE31-0XB0)	6ES7292-1AH30-0XA0	8 pines, dorados
CPU 1211C DC/DC/relé (6ES7211-1HE31-0XB0)	6ES7292-1AP30-0XA0	14 pines, estañados
CPU 1212C DC/DC/DC (6ES7212-1AE31-0XB0)		
CPU 1212C AC/DC/relé (6ES7212-1BE31-0XB0)		
CPU 1212C DC/DC/relé (6ES7212-1HE31-0XB0)		
CPU 1214C DC/DC/DC (6ES7214-1AG31-0XB0)	6ES7292-1BC30-0XA0	3 pines, dorados
CPU 1214C AC/DC/relé (6ES7214-1BG31-0XB0)	6ES7292-1AM30-0XA0	12 pines, estañados
CPU 1214C DC/DC/relé (6ES7214-1HG31-0XB0)	6ES7292-1AV30-0XA0	20 pines, estañados
CPU 1215C DC/DC/DC (6ES7215-1AG31-0XB0)	6ES7292-1BF30-0XB0	6 pines, dorados
CPU 1215C AC/DC/relé (6ES7215-1BG31-0XB0)	6ES7292-1AM30-0XA0	12 pines, estañados
CPU 1215C DC/DC/relé (6ES7215-1HG31-0XB0)	6ES7292-1AV30-0XA0	20 pines, estañados

Tabla D-2 Kits de repuesto de bloques de terminales para SM S7-1200 V3.0 e inferiores

Si tiene un SM S7-1200 V3.0 o inferior (referencia)	**Utilice este kit de repuesto de bloques de terminales (4/pq)**	
	Referencia del bloque de terminales	**Descripción del bloque de terminales**
SM 1221 DI 8 x DC (6ES7221-1BF30-0XB0)	6ES7292-1AG30-0XA0	7 pines, estañados
SM 1222 DQ 8 x DC (6ES7222-1BF30-0XB0)		
SM 1222 DQ 8 x relé (6ES7222-1HF30-0XB0)		
SM 1231 AI 4 x 13 bits (6ES7231-4HD30-0XB0)	6ES7292-1BG30-0XA0	7 pines, dorados
SM 1232 AQ 2 x 14 bits (6ES7232-4HB30-0XB0)		
SM 1231 AI 4 x TC (6ES7231-5QD30-0XB0)		
SM 1231 AI 4 x 16 bits (6ES7231-5ND30-0XB0)		
SM 1221 DI 16 x DC (6ES7221-1BH30-0XB0)	6ES7292-1AG30-0XA0	7 pines, estañados
SM 1222 DQ 16 x DC (6ES7222-1BH30-0XB0)		
SM 1222 DQ 16 x relé (6ES7222-1HH30-0XB0)		
SM 1223 DI 8 x DC/DQ 8x DC (6ES7223-1BH30-0XB0)		
SM 1223 8 x DC/8 x relé (6ES7223-1PH30-0XB0)		
SM 1223 8 x AC/8 x relé (6ES7223-1QH30-0XB0)		
SM 1234 AI 4 / AQ 2 (6ES7234-4HE30-0XB0)	6ES7292-1BG30-0XA0	7 pines, dorados
SM 1231 AI 8 x 13 bits (6ES7231-4HF30-0XB0)		
SM 1232 AQ 4 x 14 bits (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 relé (inversor) (6ES7222-1XF30-0XB0)	6ES7292-1AL30-0XA0	11 pines, estañados
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL30-0XB0)		
SM 1223 16 x DC/16 x relé (6ES7223-1PL30-0XB0)		
SM 1231 AI 8 x RTD (6ES7231-5PF30-0XB0)	6ES7292-1BL30-0XA0	11 pines, dorados

Índice alfabético

A

ABS (calcular valor absoluto), 240
AC
  Directrices de aislamiento, 60
  Directrices de cableado, 59, 61
  Puesta a tierra, 61
Acceder
  páginas web definidas por el usuario, 911
Accionamiento MicroMaster, conexión, 1010
Accionamientos, configurar accionamientos MM4, 1012
ACOS (calcular arcocoseno), 242
ACT_TINT (activar alarma horaria), 409
Actualización de firmware
  Con una SIMATIC Memory Card, 132
  Desde el servidor web, 870
  Desde STEP 7, 1204
Actualización del firmware
  Con una SIMATIC Memory Card, 132
  Desde el servidor web, 870
  Desde STEP 7, 1204
Actualizar páginas web definidas por el usuario, 892
Actualizar una CPU V3.0 a V4.x, 1448
Acumulador de tiempo (TONR), 217
ADD (sumar), 237
Agregar dispositivo
  CPU, 139
  CPU sin especificar, 140
  Detectar hardware existente, 140
Ajuste de válvulas PID, 581
Alarmas
  ATTACH (asignar OB a evento de alarma), 400
  CAN_DINT (anular alarma de retardo), 410
  DETACH (deshacer asignación entre OB y evento de alarma), 400
  Latencia de alarmas, 87
  QRY_DINT (consultar estado de una alarma de retardo), 410
  SRT_DINT (iniciar alarma de retardo), 410
  Vista general, 75
Alarmas de retardo, 410
Alias en páginas web definidas por el usuario, 900
Ambiental
  condiciones de manejo, 1250
  Condiciones de transporte y almacenamiento, 1250
Ampliar las prestaciones del S7-1200, 31
Analizador lógico, 1234
AND (operación lógica Y), 317
Archivo GSD, 745
Áreas de memoria
  Acceso inmediato, 100
  Direccionamiento de valores booleanos o de bit, 100
  Imagen de proceso, 100
Áreas de memoria, visualización con servidor web, 864
Arquitectura de sondeo, 985
Arquitectura de sondeo del esclavo, 985
Arquitectura de sondeo del maestro, 985
Arranque en caliente, 71
Arranque tras POWER ON, 71
  Procesamiento del arranque, 73
AS-i
  Agregar esclavo AS-i, 786
  Agregar módulo maestro AS-i CM1243-2, 785
  Asignación de sistema, 789
  Asignación del sistema de direcciones de esclavos, 789
  Conexión de red, 786
  Configuración de esclavos con STEP 7, 791
  Configuración de esclavos sin STEP 7, 789
  Dirección, 787
  Instrucciones E/S descentralizadas, 369
  Maestro AS-i CM 1243-2, 784
  RDREC (leer registro), 370
  Transferir valores analógicos, 791
  Transferir valores digitales, 791
  WRREC (escribir registro), 370
Asignación a la memoria local en la estructura de llamada, 103
Asignar tipos de enumeración, páginas web definidas por el usuario, 901
ASIN (calcular arcoseno), 242
Asistente de seguridad, 160
ATH (convertir cadena de caracteres ASCII en número hexadecimal), 346
ATTACH (asignar OB a evento de alarma), 400
ATTR_DB (leer atributos de un bloque de datos), 521
Australia y Nueva Zelanda - Homologación del marcado RCM, 1248
Autonegotiation, 606
AWP_Enum_Def, 901
AWP_Import_Fragment, 904
AWP_In_Variable, 895, 899
AWP_Out_Variable, 897

AWP_Start_Fragment, 903

B

Battery Board (BB)
BB 1297, 1407
Insertar la batería, 1408
BB 1297, 1407
Bits de parada, 944
Bits de toma de impulsos, configuración de entradas digitales, 156
Bloque de administración de datos (DHB), 182
Bloque de datos
Acceso estándar, 183
Acceso optimizado, 183
Bloque de datos de instancia, 99
Bloque de datos global, 99, 182
Bloques de organización (OB), 178
Borrar con DELETE_DB, 523
CONF_DATA, 716
Crear con CREATE_DB, 514
Estructura, 67
FB único con varios DBs de instancia, 182
Importar fragmentos a páginas web definidas por el usuario, 904
Leer atributos con ATTR_DB, 521
Obtener y restablecer valores, 1217
READ_DBL (leer de un bloque de datos de la memoria de carga), 518
Sincronización de los valores iniciales online y offline, 202
Vista general, 67, 182
WRIT_DBL (escribir en un bloque de datos de la memoria de carga), 518
Bloque de datos de instancia, 99
Bloque de datos global, 99, 182
Bloque de función (FB)
Bloque de datos de instancia, 181
FB único con varios DBs de instancia, 182
Llamar bloques lógicos en el programa de usuario, 177
Números válidos de FB, 67
Parámetros de salida, 181
Programas lineales y estructurados, 176
Protección de know-how, 167
Valor inicial, 181
Vista general, 67, 181
Bloque de organización
Alarma cíclica, 77
Asignación a la memoria temporal, 103
Clases de prioridad, 75
Configurar el funcionamiento, 180
Crear, 179
De ciclo, varios, 179
Función, 75
Llamada, 75
Llamar bloques lógicos en el programa de usuario, 177
Procesamiento del arranque, 73
Procesar, 178
Programación lineal y estructurada, 176
Protección de know-how, 167
Vista general, 67
Bloque de organización (OB)
Leer la información de arranque con RD_SINFO, 417
Bloque de transferencia (bloque T), 730
Bloque lógico
Alarmas, 30, 1258, 1270, 1282, 1295, 1309
Bloque de datos (DB) de instancia, 181
Bloques de organización (OB), 30, 178, 1258, 1270, 1282, 1295, 1309
Contadores (requisitos de memoria y cantidad), 30, 1258, 1270, 1282, 1295, 1310
DB (bloque de datos), 67, 182
Enlazar a una CPU, Memory Card o contraseña, 168
FB (bloque de función), 67, 181
FC (función), 67, 180
Llamadas de bloque, 67
Llamar bloques lógicos en el programa de usuario, 177
Número de bloques lógicos, 29, 1257, 1270, 1282, 1294, 1309
Número de OB, 30, 1258, 1270, 1282, 1295, 1309
Números válidos de FC, FB y DB, 67
Observar, 29, 1257, 1270, 1282, 1294, 1309
Profundidad de anidamiento, 29, 1257, 1270, 1282, 1294, 1309
Programas lineales y estructurados, 176
Protección anticopia, 168
Protección de know-how, 167
Tamaño del programa de usuario, 29, 1257, 1270, 1282, 1294, 1309
Temporizadores (requisitos de memoria y cantidad), 30, 1258, 1270, 1282, 1295, 1309
Tipos de bloques lógicos, 67
Valor inicial de un FB, 181
Bloques
Alarmas, 30, 87, 1258, 1270, 1282, 1295, 1309
Bloque de datos (DB), 67
Bloque de datos (DB) de instancia, 181
Bloque de función (FB), 67, 181

Bloques de organización (OB), 30, 67, 75, 87, 1258, 1270, 1282, 1295, 1309
Cargar, 199
comprobación de coherencia, 207
Contadores (requisitos de memoria y cantidad), 30, 1258, 1270, 1282, 1295, 1310
Copiar bloques desde una CPU online, 204
DB de instancia única o multiinstancia, 181
Eventos, 87
Función (FC), 67, 180
Llamadas de bloque, 67
Llamar un FB o una FC con SCL, 191
Número de bloques lógicos, 29, 1257, 1270, 1282, 1294, 1309
Número de OB, 30, 87, 1258, 1270, 1282, 1295, 1309
Números válidos de FC, FB y DB, 67
OB de arranque, 87
Observar, 29, 1257, 1270, 1282, 1294, 1309
Profundidad de anidamiento, 29, 67, 1257, 1270, 1282, 1294, 1309
Programas lineales y estructurados, 176
Protección por contraseña, 167
Tamaño del programa de usuario, 29, 67, 1257, 1270, 1282, 1294, 1309
Temporizadores (requisitos de memoria y cantidad), 30, 1258, 1270, 1282, 1295, 1309
Tipos de, 67
Tipos de bloques lógicos, 67
Valor inicial de un FB, 181
Bloques de datos estándar, 183
Bloques de datos optimizados, 183
Bobina normalmente abierta/cerrada, 211
Bobinas, 209
Byte de marcas de ciclo, 96
Byte de marcas de sistema, 96

C

Cable
Ampliación, 1427
Comunicación de red, 938
Cable de ampliación, 1427
Desmontaje, 57
Instalación, 57
CALCULATE (calcular), 236
Calendario, 325
Cambiar dispositivo, 152
Cambiar idiomas, páginas web definidas por el usuario, 924
Cambio de dispositivo
CPU V3.0 por una CPU V4.x, 1448
Procedimiento, 152
CAN_DINT (anular alarma de retardo), 410
CAN_TINT (anular alarma horaria), 408
Captura de impulsos, 156
Capturar impulsos, 170
Carácter de fin del mensaje, 953
Carácter de inicio del mensaje, 949
Caracteres especiales
Páginas web definidas por el usuario, 905
Cargar
Actualización de firmware, 132
Certificado de seguridad Siemens a PC, 863
Copiar bloques desde una CPU online, 204
DBs de páginas web definidas por el usuario, 910
Programa de usuario, 199, 204
Proyecto, 199
Visualizar las direcciones MAC e IP, 608
Cargar en estado operativo RUN
Cargar bloques seleccionados, 1227
Cargar sin reinicialización, 1229
Configuración de reserva de memoria global, 1231
Error en la descarga, 1232
Errores de compilación, 1228
Factores, 1232
Iniciar desde STEP 7, 1226
Interfaz de bloque extendida, 1229
Requisitos previos, 1225
Reserva de memoria y reserva de memoria remanente, 1229
Restricciones, 1231
Vista general, 1224
Cargas de lámpara, 62
Cargas inductivas, 63
Carpetas, idiomas para páginas web definidas por el usuario, 924
Categoría de sobretensión, 1251
CB 1241
Terminación y polarización, 939
CB 1241 RS485, 1419
CEIL (redondear un número en coma flotante al siguiente entero superior), 289
Certificado de seguridad Siemens, páginas web, 863
Char (tipo de datos Character), 112
Chars_TO_Strg (convertir Array of CHAR en cadena de caracteres), 344
Ciclo
Operación de forzado permanente, 1222
Vista general, 91

Circuitos de protección contra sobretensión para cargas inductivas, 63
Circuitos supresores para cargas inductivas, 63
Clase de protección, 1251
Clases de error PtP, 960, 1112
Cola de espera, 87
Comandos AWP, 893
  Combinar definiciones, 905
  definir un tipo de enumeración, 901
  Escribir variables, 895
  escribir variables especiales, 899
  Generar fragmentos, 903
  importar fragmentos, 904
  Leer variables especiales, 897
  referenciar un tipo de enumeración, 901
  utilizar un alias, 900
Communication Board (CB)
  Agregar módulos, 142
  CB 1241 RS485, 1419
  Configuración de dispositivos, 137
  Configuración de parámetros, 169
  Desmontaje, 52
  Indicadores LED, 937, 1197
  Instalación, 52
  Programación, 984
  RS485, 937
  Tabla de comparación, 31
  Vista general, 31
Compact switch module CSM 1277, 1428
Compact switch module, CSM 1277, 1428
Comparar valores, 231
Comparar y sincronizar CPU online/offline, 1213
Compatibilidad, 40
Compatibilidad electromagnética, 1249
Compatibilidad electromagnética (CEM), 1249
Compensación de unión fría, termopar, 1357, 1400
Comprobación de coherencia, 207
Comprobar el programa, 205
Comprobar la conexión, 678
Comunicación
  Activa/pasiva, 597, 618, 801
  Arquitectura de sondeo, 985
  Carga de comunicación, 92
  Conexión de hardware, 725
  Conexión de red, 596
  Configuración, 597, 618, 801
  Control de flujo, 945
  Dirección AS-i, 787
  Dirección IP, 603
  Dirección MAC, 603
  Dirección PROFIBUS, 782
  ID de conexión, 616
  Número de conexiones (PROFINET/PROFIBUS), 592
  Parámetros de transmisión y recepción, 946
  Pérdida, extracción o inserción de módulos, 82
  PROFINET y PROFIBUS, 585
  Propiedad de sincronización horaria (PROFINET), 611
  Protocolos, 614
  Red, 724
  TCON_Param, 618
  Tiempo de ciclo, 92
Comunicación activa/pasiva
  Configurar los interlocutores, 597, 801
  ID de conexión, 616
  Parámetros, 618
Comunicación de red, 724
  Cable de polarización y terminación, 938
Comunicación pasiva/activa
  Configurar los interlocutores, 597, 801
  ID de conexión, 616
  Parámetros, 618
Comunicación PtP, 940
  Configuración del programa de ejemplo, 987
  configurar parámetros, 946
  Configurar puertos, 943
  emulador de terminal para el programa de ejemplo, 993
  programa de ejemplo, 986
  programa de ejemplo, en proceso, 994
  programa de ejemplo, programación de STEP 7, 992
  Programación, 984
Comunicación PtP, 3964(R)
  Configurar la prioridad y los parámetros de protocolo, 957
  Configurar puertos, 956
Comunicación punto a punto, 940
Comunicación S7
  Configurar la conexión, 598
Comunicación segura, 589
Comunicación serie, 940
Comunicación TCP/IP, 613
Comunicación Teleservice
  TM_MAIL (transmitir e-mail), 1189
CONCAT (agrupar cadenas de caracteres), 350
Condiciones de fin, 951
Condiciones de inicio, 948
Conector de bloque de terminales, 56
Conector de bus, 32
Conector del bloque de bornes, 1440
Conector RS485
  Terminación y polarización, 938

Conector, instalación y desmontaje, 56
Conexión activa/pasiva, 596
Conexión de red
  Conexión de dispositivos, 596
  múltiples CPU, 733
  Varias CPUs, 728, 729, 781, 786
Conexión inalámbrica al servidor web, 852
Conexión local/interlocutor, 596
Conexiones
  Conexión S7, 799
  Configuración, 618
  ID de conexión, 616
  Interlocutores, 597, 801
  Número de conexiones (PROFINET/ PROFIBUS), 592
  Protocolos Ethernet, 799
  Servidor web, 933
  Tipos de comunicación, 585
  Tipos, conexiones multinodo, 799
Conexiones HTTP, servidor web, 933
Conexiones multinodo
  Protocolos Ethernet, 799
  Tipos de conexión, 799
Conexiones servidor web máximas, 933
Configuración
  Agregar módulos, 142
  AS-i, 787
  Carga de comunicación, 92
  Cargar, 199
  Comunicación entre PLCs, 728
  Conexión de red, 596
  Determinar, 140
  Dirección IP, 603
  Dirección MAC, 603
  Dirección PROFIBUS, 782
  HSC (contador rápido), 555
  Interfaces de comunicación, 943
  Módulos, 169
  Parámetros de arranque, 122
  Parámetros de la CPU, 153
  PROFIBUS, 782
  Propiedad de sincronización horaria (PROFINET), 611
  Puerto AS-i, 786
  Puerto Ethernet, 603
  Puerto PROFINET, 603
  Puertos, 943
  Recepción de mensajes, 947
  RS422, estados operativos, 989
  RS485, estados operativos, 991
  Tiempo de ciclo, 91
Configuración de CPU
  Canales de impulsos, 473
  Comunicación con la HMI, 727
  Parámetros operativos, 153
  Propiedades de módulo, 169
  Varias CPU, 728
  Vigilancia del tiempo de ciclo, 91
Configuración de dispositivos, 137, 725
  Agregar dispositivo, 139
  Agregar módulos, 142
  AS-i, 787
  Cambiar un tipo de dispositivo, 152
  Cargar, 199
  Conexión de red, 596
  Configurar la CPU, 153
  Configurar los módulos, 169
  Determinar, 140
  PROFIBUS, 782
  Propiedad de sincronización horaria (PROFINET), 611
  Puerto AS-i, 787
  Puerto Ethernet, 603
  Puerto PROFINET, 603
Configuración de la recepción de mensajes
  Configuración de dispositivos PtP, 947
  Programa de ejemplo PtP, 988
Configuración de la transferencia de mensajes
  Configuración de dispositivos PtP, 946
  Programa de ejemplo PtP, 987
Configuración de los parámetros de recepción, 732
Configuración de mensajes
  Instrucciones, 984
  Recibir, 947
  Transmitir, 946
Configuración de parámetros
  LENGTH y BUFFER para SEND_P2P, 975
  Recibir, 732
  Transmitir, 731
Configuración de parámetros de transmisión, 597, 731, 801
Configuración de puerto, 3964(R), 956
Configuración de usuario, servidor web, 848
Configuración del envío de mensajes, 946
Configuración del mensaje transmitido, 946
Configuración del puerto, 943
  Errores, 963, 1114
  Instrucciones, 984
  Programa de ejemplo PtP, 987
Configuración futura (control de configuración), 142
Configuración hardware, 137
  Agregar dispositivo, 139
  Agregar módulos, 142

AS-i, 787
Cargar, 199
Conexión de red, 596
Configurar la CPU, 153
Configurar los módulos, 169
Determinar, 140
PROFIBUS, 782
Puerto AS-i, 787
Puerto Ethernet, 603
Puerto PROFINET, 603
Configuración, 3964(R)
Interfaces de comunicación, 956
Prioridad y parámetros de protocolo, 957
Puertos, 956
Configuración, páginas web definidas por el usuario
Configuración de idiomas múltiples, 927
Configuración de STEP 7, 907
Consumo de corriente, 44, 1431
Contacto normalmente abierto/cerrado, 209
Contactos, 209
Contactos de conexión
Capacidad máxima de carga de corriente, 1418
Contador de horas de funcionamiento (RTM), 332
Contador rápido, 540, 566
Configuración, 555
Fase de operación, 557
Modos de contaje, 556
No se puede forzar permanentemente, 1223
Contadores
Cantidad, 30, 1258, 1270, 1282, 1295, 1310
Configuración del HSC, 555
CTD (decrementar contador), 226
CTRL_HSC (controlar contadores rápidos), 566
CTRL_HSC_EXT (controlar contadores rápidos (avanzado)), 540
CTU (incrementar contador), 226
CTUD (incrementar y decrementar contador), 226
Operación (contadores estándar), 227
Tamaño, 30, 1258, 1270, 1282, 1295, 1310
CONTINUE, SCL, 315
Contraseña olvidada, 135
Control de configuración (configuración futura), 142
Ejemplo, 149
Juego de datos de control, 146
Control de flujo, 944
Configuración, 944
Gestionar, 945
Control de flujo por hardware, 945
Control de flujo por software, 946
Control del programa (SCL), 309
CASE, 311
CONTINUE, 315
EXIT, 315
FOR, 312
GO TO, 316
IF-THEN, 310
REPEAT, 314
RETURN, 316
WHILE, 313
Control manual de DB de fragmentos, 927
CONV (convertir valor), 284
Convenciones para comillas, servidor web, 905
Conversión (instrucciones SCL), 285
Cookie, siemens_automation_language, 924
Copiar bloques desde una CPU online, 204
Copiar, cortar y pegar en STEP 7, 40
Correo electrónico, enviar con TMAIL_C, 682
Corriente disponible, 44
Ejemplo, 1432
forma de cálculo, 1433
Vista general, 1431
COS (calcular coseno), 242
CountOfElements (Consultar número de elementos de ARRAY), 263
CPU
Agregar dispositivo, 139
Agregar módulos, 142
AS-i, 786
Asignar una dirección IP a una CPU online, 602
Cable de ampliación, 57
Carga de comunicación, 92
Cargar, 199
Cargar en dispositivo, 608
Cargas de lámparas, 62
Cargas inductivas, 63
Communication Boards (CB), 31
comparar y sincronizar bloques, 1213
Compatibilidad de versiones, 40
Comportamiento en caso de sobrecarga, 88
Comunicación, 594
Conector de bloque de terminales, 56
Conexión de red, 596
Configuración de dispositivos, 137
Configuración del HSC, 555
Configuración del tiempo de ciclo, 92
Consumo de corriente, 1431
Contraseña olvidada, 135
Copiar bloques desde una CPU online, 204
Corriente disponible, 44
CPU 1211C AC/DC/Relay, 1255
CPU 1211C DC/DC/DC, 1255
CPU 1211C DC/DC/relé, 1255
CPU 1212C AC/DC/relé, 1268
CPU 1212C DC/DC/DC, 1268

CPU 1212C DC/DC/relé, 1268
CPU 1214C AC/DC/relé, 1280
CPU 1214C DC/DC/DC, 1280
CPU 1214C DC/DC/relé, 1280
CPU 1215C AC/DC/relé, 1292
CPU 1215C DC/DC/DC, 1292
CPU 1215C DC/DC/relé, 1292
CPU 1217C DC/DC/DC, 1307
CPU sin especificar, 140
Desbloquear las salidas en modo STOP, 1221
Dirección AS-i, 787
Dirección IP, 603
Dirección MAC, 603, 608
Dirección PROFIBUS, 782
Directrices de aislamiento, 60
Directrices de cableado, 59, 61
Ejecución del programa, 67
Establecer conexión online, 1201
Estados operativos, 71
Estados operativos RUN/STOP, 1210
Forzado permanente, 1222
Forzar permanentemente, 1222
Indicadores LED, 1197
Instalación, 48, 49
niveles de seguridad, 163
Número de conexiones de comunicación, 592
Obtener y restablecer valores del DB, 1217
Online, 1204
Panel de control (CPU online), 1210
Parámetros de arranque, 122
Procesamiento del arranque, 73
Procesar los OB, 178
PROFINET IO, 733
Propiedad de sincronización horaria, 611
Protección de acceso con contraseñas, 163
Protección de know-how, 167
Puerto AS-i, 786
Puerto Ethernet, 603
Puerto PROFINET, 603
Puesta a tierra, 61
Realizar una copia de seguridad, 1240
Recuperación de una contraseña perdida, 135
Restablecer los ajustes de fábrica, 1206
Restaurar una copia de seguridad, 1243
RTM (contador de horas de funcionamiento), 332
Salidas de impulsos, 471
Signal Boards (SB), 31
Tabla comparativa, 28
Tabla de observación, 1218
Tarjeta de transferencia vacía, 135
Tiempos de respuesta a un escalón, 1263, 1276, 1288, 1300, 1317
Tipos de comunicación, 585
Vigilancia online, 1215
Vista general, 27
Visualizar las direcciones MAC e IP, 608
Zona de disipación, 44, 47
CPU sin especificar, 140
Creación de páginas web definidas por el usuario, 892
Crear DB de páginas web definidas por el usuario, 908
Crear una conexión de red
Entre PLCs, 596
CREATE_DB (crear bloque de datos), 514
CTD (decrementar contador), 226
CTRL_HSC (controlar contadores rápidos), 566
CTRL_HSC_EXT (controlar contadores rápidos (avanzado)), 540
CTS (control de flujo por hardware, PtP), 945
CTU (incrementar contador), 226
CTUD (incrementar y decrementar contador), 226
cuadro & (operación lógica Y FUP), 210
cuadro /= (asignación negativa FUP), 211
cuadro = (asignación FUP), 211
cuadro >=1 (operación lógica O FUP), 210
Cuadro N= y bobina N (activar operando con flanco de señal descendente), 215
Cuadro P= y bobina P (activar operando con flanco de señal ascendente), 215
cuadro x (operación lógica O-exclusiva FUP), 210
Customer support, 3

D

D_ACT_DP, 380
DataLogClear, 499
DataLogDelete, 501
Date
DTL (tipo de datos Data and Time Long), 111
Tipo de datos Date, 110
Datos técnicos, 1245
BB 1297, 1407
CB 1241 RS485, 1419
CM 1241 RS232, 1421
CM 1241 RS422/485, 1422
Compatibilidad electromagnética (CEM), 1249
Condiciones ambientales, 1250
CPU 1211C AC/DC/Relay, 1255
CPU 1211C DC/DC/DC, 1255
CPU 1211C DC/DC/relé, 1255
CPU 1212C AC/DC/relé, 1268
CPU 1212C DC/DC/DC, 1268
CPU 1212C DC/DC/relé, 1268

CPU 1214C AC/DC/relé, 1280
CPU 1214C DC/DC/DC, 1280
CPU 1214C DC/DC/relé, 1280
CPU 1215C AC/DC/relé, 1292
CPU 1215C DC/DC/DC, 1292
CPU 1215C DC/DC/relé, 1292
CPU 1217C DC/DC/DC, 1307
datos técnicos generales, 1245
entornos industriales, 1249
Homologaciones, 1245
Memory Cards, 1424
Módulo del potenciómetro, 1426
Representación de entradas analógicas (intensidad), 1352, 1396
Representación de entradas analógicas (tensión), 1352, 1395
Representación de salidas analógicas (intensidad), 1353, 1397
Representación de salidas analógicas (tensión), 1353, 1396
SB 1221 DI 4 x 24 V DC, 200 kHz, 1380
SB 1221 DI 4 x 5 V DC, 200 kHz, 1380
SB 1222 DQ 4 x 24 V DC, 200 kHz, 1382
SB 1222 DQ 4 x 5 V DC, 200 kHz, 1382
SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1388
SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1385
SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1385
SB 1231 AI 1 x 12 bits, 1391
SB 1231 AI 1 x 16 bits de termopar, 1398
SB 1231 AI 1 x 16 bits RTD, 1403
SB 1232 AQ 1 x 12 bits, 1393
Simuladores de entradas, 1424
SM 1221 DI 16 x 24 V DC, 1323
SM 1221 DI 8 x 24 V DC, 1323
SM 1222 DQ 16 x 24 V DC, 1326
SM 1222 DQ 16 x Relay, 1326
SM 1222 DQ 8 Relay Changeover, 1325
SM 1222 DQ 8 x 24 V DC, 1325
SM 1222 DQ 8 x Relay, 1325
SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1332
SM 1223 DI 16 x 24 V DC, DQ 16 x Relay, 1332
SM 1223 DI 8 x 120/230 V AC/DQ 8 x relé, 1339
SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1332
SM 1223 DI 8 x 24 V DC, DQ 8 x Relay, 1332
SM 1231 AI 4 x 13 bits, 1341
SM 1231 AI 4 x 16 bits, 1341
SM 1231 AI 4 x 16 bits TC, 1354
SM 1231 AI 4 x RTD x 16 bits módulo de señales, 1360
SM 1231 AI 8 x 13 bits, 1341
SM 1231 AI 8 x 16 bits TC, 1354
SM 1231 AI 8 x RTD x 16 bits módulo de señales, 1360
SM 1232 AQ 2 x 14 bit, 1346
SM 1232 AQ 4 x 14 bit, 1346
SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 1348
SM 1278 4xIO-Link Master, 1366
Tensiones nominales, 1251
Tiempos de respuesta a un escalón (CPU), 1263, 1276, 1288, 1300, 1317
Tiempos de respuesta a un escalón (SB), 1395
Tiempos de respuesta a un escalón (SM), 1351
DB (bloque de datos), (Bloque de datos)
DB de control para páginas web definidas por el usuario
Comandos globales, 927
Comandos y estados de órdenes, 927
Parámetro para instrucción WWW, 909
DB_ANY_TO_VARIANT (convertir DB_ANY en VARIANT), 294
DBs de fragmentos (páginas web definidas por el usuario)
Crear a partir de comando AWP, 903
generar, 908
importar con comando AWP, 904
DC
Cargas inductivas, 63
Directrices de aislamiento, 60
Directrices de cableado, 59, 61
Puesta a tierra, 61
Salidas, 1252
DEC (Decrementar), 239
DECO (descodificar), 318
Definiciones múltiples de variables AWP, 905
Definir tipos de enumeración, páginas web definidas por el usuario, 901
DELETE (borrar caracteres de una cadena), 352
DELETE_DB (borrar bloque de datos), 523
DEMUX (desmultiplexar), 321
Depuración en modo RUN, 1224
Depurar en estado operativo RUN, 1232
Deserializar, 248
Desfase, OBs de alarma cíclica, 77
DETACH (deshacer asignación entre OB y evento de alarma), 400
Determinar para cargar una CPU online, 140
DeviceStates (leer la información de estado del módulo de un sistema E/S), 445
DeviceStates, ejemplo, 446
Diagnóstico
Búfer, 97
búfer de diagnóstico, 1211
carga de la memoria, 1211

DeviceStates (leer la información de estado del módulo de un sistema E/S), 445
GET_DIAG (leer información de diagnóstico), 455
Get_IM_Data (leer datos de identificación y mantenimiento), 428
Indicador de estado, 96
Indicadores LED, 1197
LED (leer estado del LED), 427
ModuleStates (leer información de estado del submódulo de un módulo), 450
Tabla de observación, 1218
Tiempo de ciclo, 1211
Diagnóstico, reducir los eventos de seguridad, 97
Diagramas de cableado
CB 1241 RS 485, 1420
CPU 1211C, 1265
CPU 1212C, 1277
CPU 1214C, 1289
CPU 1215C, 1303
CPU 1217C, 1320
SB 1221, 1381
SB 1222, 1384
SB 1223, 1387, 1390
SB 1231, 1393
SB 1231 de termopar, 1402
SB 1231 RTD, 1404
SB 1232, 1394
SM 1221, 1324
SM 1222, 1328
SM 1223, 1335, 1341
SM 1231, 1344
SM 1231 de termopar, 1355
SM 1231 RTD, 1361
SM 1232, 1347
SM 1234, 1350
SM 1278 maestro IO-Link, 1368
Diferencias
En instrucciones Modbus RTU, 1084
En instrucciones Modbus TCP, 1016
En instrucciones TCON, TDISCON, TSEND y TRCV, 651
En instrucciones TSEND_C y TRCV_C, 630
En instrucciones USS, 995
en las instrucciones de comunicación punto a punto, 940
Dirección IP, 604
Asignar, 600, 607
Asignar online, 602
Configuración de dispositivos, 153
Configurar, 603
configurar CPU online, 1204
Dirección MAC, 603
Dirección IP del router, 604
Dirección IP, emergencia (temporalmente), 808
Dirección MAC, 603, 608
Direccionamiento
Áreas de memoria, 100
Entradas (I) o salidas (Q) individuales, 100
Imagen de proceso, 100
Valores booleanos o de bit, 100
Direcciones
Leer dirección de estación con GetStationInfo, 435
Leer la dirección de MAC con GetStationInfo, 435
Directorios, idiomas para páginas web definidas por el usuario, 924
Directrices
Aislamiento, 60
Cargas de lámparas, 62
Cargas inductivas, 63
Directrices de cableado, 59, 61
Instalación, 43
Instalación de la CPU, 49
Procedimientos de instalación, 48
Puesta a tierra, 61
Directrices de aislamiento, 60
Directrices de cableado, 61
Espacio libre para ventilación y refrigeración, 44
Puesta a tierra, 61
Requisitos, 59
DIS_AIRT (retardar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad), 413
Diseñar un sistema PLC, 175, 176
Disparar
Valores en la tabla de observación, 1220
Dispositivo
Compartido, 748
Nombres de dispositivo PROFINET IO, 734
PROFINET IO, 733
Dispositivo móvil, acceso al servidor web, 852
Dispositivos accesibles
Formatear una SIMATIC Memory Card, 1209
Dispositivos accesibles, actualización del firmware, 1205
Dispositivos HMI
Conexión de red, 596
Configurar la comunicación PROFINET, 727
Vista general, 32
Dispositivos móviles
Estructura de página web, 858
Dispositivos PROFINET IO
Escribir todas las salidas con SETIO, 374

Escribir una parte de las salidas con SETIO_PART, 376
Leer una parte de las entradas con GETIO_PART, 375
DIV (dividir), 237
Documentación, 4
DPNRM_DG, 396
DPRD_DAT (leer datos coherentes de un esclavo DP normalizado), 389
DPWR_DAT (escribir datos coherentes de un esclavo DP normalizado), 389

E

E/S
Cargas inductivas, 63
Direccionamiento, 104
Indicadores de estado analógicos, 1199, 1201
Indicadores de estado digitales, 1198
observar estado en KOP, 1216
Operación de forzado permanente, 1222
Representación de entradas analógicas (intensidad), 1352, 1396
Representación de entradas analógicas (tensión), 1352, 1395
Representación de salidas analógicas (intensidad), 1353, 1397
Representación de salidas analógicas (tensión), 1353, 1396
Tiempos de respuesta a un escalón (CPU), 1263, 1276, 1288, 1300, 1317
Tiempos de respuesta a un escalón (SB), 1395
Tiempos de respuesta a un escalón (SM), 1351
Vigilar con una tabla de observación, 1218
E/S analógicas
Configuración, 170
Conversión a unidades de ingeniería, 105, 291
Indicadores de estado, 1199, 1201
Representación de entradas (intensidad), 1352, 1396
Representación de entradas (tensión), 1352, 1395
Representación de salidas (intensidad), 1353, 1397
Representación de salidas (tensión), 1353, 1396
Tiempos de respuesta a un escalón (CPU), 1263, 1276, 1288, 1300, 1317
Tiempos de respuesta a un escalón (SB), 1395
Tiempos de respuesta a un escalón (SM), 1351
E/S digitales
Capturar impulsos, 170
Configuración, 170
Indicadores de estado, 1198
Editar en estado operativo RUN, (Cargar en estado operativo RUN)
Editor de programas
estado, 1216
observar, 1216
Ejecución de eventos y cola de espera, 87
Ejecución del programa, 67
Ejecutar comandos de eje como secuencia de movimientos (MC_CommandTable), 576
Ejemplo de ModuleStates, 452
Ejemplo de programa de receta, 487
Ejemplos varios
Acceder a elementos Array, 265
Cálculo de la corriente disponible, 1432
Conexión de maestro S7-1200 IO-Link, 1372
Control de configuración (configuración futura), 149
CPU 1217C Entrada diferencial y aplicación, 1321
CPU 1217C Salida diferencial y aplicación, 1322
Descargar bloques seleccionados en estado operativo RUN, 1227
Evaluación de ENO en SCL, 197
Instrucciones CASE anidadas, SCL, 312
Procesamiento de valores analógicos, 291
Programa de registro, 509
Receta, 487
Ejemplos, comunicación
Comunicación de CPU con conexiones de transmisión y recepción separadas, 616
Comunicación de CPU con una conexión de transmisión y recepción común, 617
Comunicación de CPU mediante conexiones TSEND_C o TRCV_C, 618
Configurar una conexión S7 PROFIBUS, 806
Configurar una conexión S7 PROFINET, 804
Direccionamiento de esclavo AS-i, 787
I-device como dispositivo IO y controlador IO, 740
Protocolos de comunicación PROFINET, 613
Shared Device, 748
Shared I-device, 753
T_CONFIG, cambiar parámetros IP, 719
T_CONFIG, cambiar parámetros IP de los servidores NTP, 722
T_CONFIG, cambiar parámetros IP y nombres de dispositivo PROFINET IO, 721
Telecontrol, 1184
Ejemplos, comunicación PtP
Comunicación Legacy PtP, RCV_CFG, 1118
Condición de fin de mensaje, 954
Condición de inicio de mensaje, 949

Configuración, 987
Ejecutar el ejemplo de emulador de terminal, 994
Emulador de terminal, 986, 993
Longitud del mensaje dentro del mensaje, 954
Programación de STEP 7, 992
Receive_Config, 968
Ejemplos, comunicación USS
Notificación de error de comunicación USS, 1009
Notificación de errores de comunicación Legacy USS, 1140
Ejemplos, CP Modbus antiguo
Parámetro MB_HOLD_REG, 1150
Ejemplos, instrucciones
ATH (ASCII a hexadecimal), 347
Bobinas de temporizador, 219
CONTINUE, SCL, 315
CTRL_PWM, 475
DECO (Descodificar), 319
Deserializar, 249
DeviceStates, PROFIBUS y PROFINET, 446
Ejecución de alarma cíclica y parámetro de tiempo con SET_CINT, 404
EXIT, SCL, 316
GET_DIAG y modos, 461
GOTO (SCL), 316
HTA (hexadecimal a ASCII), 348
LIMIT (ajustar valor límite), 242
ModuleStates, PROFIBUS y PROFINET, 452
RETURN, SCL, 316
ROR (Rotar a la derecha), SCL, 324
RUNTIME (Medir tiempo de ejecución), 308
S_CONV (convertir cadena de caracteres), 342
Serializar, 251
SHL (Desplazar a la izquierda), SCL, 323
STRG_VAL (convertir cadena en valor numérico), 343
SWAP (cambiar disposición), 254
TM_MAIL, 1194
VAL_STRG (convertir valor numérico en cadena de caracteres), 344
Variantes de PEEK y POKE, 195, 258
Ejemplos, instrucciones de cadenas en tiempo de ejecución
GetBlockName, 368
GetInstanceName, 364
GetInstancePath, 365
GetSymbolName, 358
GetSymbolPath, 361
Ejemplos, Legacy Modbus
Legacy Modbus RTU, direccionamiento de registro de retención, 1171
Legacy Modbus RTU, ejemplos de parámetros MB_HOLD_REG, 1169
Ejemplos, Modbus
Coordinación de varias peticiones Modbus TCP MB_CLIENT, 1083
MB_CLIENT Peticiones múltiples con conexión Modbus TCP común, 1080
MB_SERVER conexiones Modbus TCP múltiples, 1079
Modbus TCP, direccionamiento del registro de retención, 1034
Modbus TCP, ejemplos de parámetros, MB_HOLD_REG, 1031
Modbus TCP, parámetros de conexión MB_CLIENT, 1023
Modbus TCP, parámetros de conexión MB_SERVER, 1029
Petición de escritura de memoria imagen de salida Modbus TCP MB_CLIENT, 1082
Peticiones múltiples MB_CLIENT con varias conexiones Modbus TCP, 1081
Programa de esclavo Modbus RTU, 1111
Programa de maestro Modbus RTU, 1108
Ejemplos, Modbus RTU antiguo
Programa esclavo, 1175
Programa maestro, 1174
Ejemplos, Modbus TCP antiguo
Conexiones Modbus TCP múltiples con MB_SERVER, 1154
Coordinación de varias peticiones Modbus TCP MB_CLIENT, 1157
Direccionamiento del registro de retención, 1152
MB_CLIENT: peticiones múltiples con conexión Modbus TCP común, 1154
Petición de escritura de memoria imagen de salida MB_CLIENT, 1156
Peticiones múltiples MB_CLIENT con varias conexiones Modbus TCP, 1155
Ejemplos, servidor web
Acceso desde dispositivo móvil, 853
Alias, 895, 900
Caracteres especiales en comandos AWP, 906
Combinar declaraciones AWP, 905
DBs de fragmentos, 904
Escribir variables, 896, 917
Escribir variables especiales, 899, 918
Leer variables, 895, 915
Leer variables especiales, 898
Página web definida por el usuario, 913, 918

Página web definida por el usuario para cambiar idiomas, 925
Programa STEP 7 para comprobar fragmentos, 930
Tipos de enumeración, 901, 902, 916
Ejemplos, varios
Función de traza y analizador lógico, 1235
Procesamiento de valores analógicos, 105
Receta, 479
Segmento de un tipo de datos marcado, 117
Superposición de variable AT, 118
Emulador de terminal para el programa de ejemplo PtP, 993
EN y ENO (flujo de corriente), 197
EN_AIRT (habilitar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad), 413
ENCO (codificar), 318
ENDIS_PW (limitar y habilitar legitimación de la contraseña), 300
Enlace dinámico, 168
Enlazar a una CPU, Memory Card o contraseña, 168
Entornos industriales
Homologaciones, 1249
Entrada principal, 840
Entradas
Bits de captura de impulsos, 156
Entradas y salidas
observar, 1215
EQ_ElemType (Comparar el tipo de datos de un elemento ARRAY para UNEQUAL con el tipo de datos de una variable), 234
EQ_Type (Comparar el tipo de datos para EQUAL con el tipo de datos de una variable), 234
Error de versión de CPU desconocida, 1199
Error de versión de CPU incompatible, 1199
Errores
Errores comunes de las instrucciones avanzadas, 531
Errores de diagnóstico, 80
Errores de tiempo, 79
Errores de configuración de la recepción, 970, 1119
Errores de configuración de la transmisión, 965, 1115
Errores de procesamiento de señales, 980, 981, 1126, 1127
Errores diversos de parámetros PtP, 960
Errores en tiempo de ejecución de la transmisión, 974, 1122
Escalar valores analógicos, 291
Esclavo PN
Activar y desactivar con D_ACT_DP, 380
Esclavos DP normalizados
Escribir todas las salidas con SETIO, 374
Escribir una parte de las salidas con SETIO_PART, 376
Leer todas las entradas con GETIO, 373
Leer una parte de las entradas con GETIO_PART, 375
Escritura en DB, E/S o memoria, 195, 258
Espacio libre, ventilación y refrigeración, 44
Especificaciones
SM 1222 DQ 16 x 24 V DC sumidero, 1326
Establecer conexión online, 1201
Estación
Leer información con GetStationInfo, 435
Estado
Indicadores LED, 1197
Indicadores LED (interfaz de comunicación), 937
Estado operativo
cambiar STOP/RUN, 1210
Estados operativos de la CPU, 71
Estado operativo RUN, 71, 74, 1210
Estado operativo STOP, 71, 1210
Desbloquear las salidas en modo STOP, 1221
Estructura de llamadas, 207
Estructura del programa, 178
Ethernet
Compact switch module CSM 1277, 1428
Conexión de red, 596
Dirección IP, 603
Dirección MAC, 603
DPNRM_DG (leer datos de diagnóstico de un esclavo DP), 396
DPRD_DAT (leer datos coherentes de un esclavo DP normalizado), 389
DPWR_DAT (escribir datos coherentes de un esclavo DP normalizado), 389
GET (leer datos de una CPU remota), 795
ID de conexión, 616
Instrucciones Legacy TCON, TDISCON, TSEND y TRCV, 668
Modo ad hoc, 615
Número de conexiones de comunicación, 592
PRVREC (proveer registro), 394
PUT (escribir datos en una CPU remota), 795
RALRM (recibir alarma), 377
RCVREC (recibir registro), 392
RDREC (leer registro), 370
T_CONFIG (configurar interfaz), 712
TCON, 652
TDISCON, 652
Tipos de comunicación, 585
TRCV, 652

TRCV_C, 631
TRCV_C Legacy (recibir datos vía Ethernet (TCP)), 644
TSEND, 652
TSEND_C, 631
TSEND_C Legacy (enviar datos vía Ethernet (TCP)), 644
TURCV (recibir datos vía Ethernet (UDP)), 707
TUSEND (enviar datos vía Ethernet (UDP)), 707
Vista general, 613
WRREC (escribir registro), 370
Etiquetas de bloques remanentes
Cargar en estado operativo RUN, 1230
Eventos de seguridad en el búfer de diagnóstico, 97
EXIT, SCL, 315
EXP (calcular valor exponencial), 242
EXPT (elevar a potencia), 242

F

F_TRIG (activar variable con flanco de señal descendente), 216
FAQs, 4
Fase de operación
HSC (contador rápido), 557
FB (bloque de función)
Vista general, 67
FC (función), 67, 180
Fecha
SET_TIMEZONE (ajustar zona horaria), 331
T_ADD (sumar tiempos), 326
T_COMBINE (combinar tiempos), 327
T_CONV (convertir y extraer tiempos), 325
T_DIFF (diferencia de tiempo), 327
T_SUB (restar tiempos), 326
FieldRead (leer campo), 264
FieldWrite (escribir campo), 264
FileDelete, 537
FileWriteC, 534
FILL_BLK (rellenar área), 253
FIND (buscar caracteres en una cadena), 355
FLOOR (redondear un número en coma flotante al siguiente entero inferior), 289
FOR, SCL, 312
Formatear una SIMATIC Memory Card, 1209
Forzado permanente
Ciclo, 1222
Entradas de periferia, 1222
Entradas y salidas, 1222
Memoria I, 1222
Tabla de observación, 1218
Forzar permanentemente, 1222
FRAC (determinar decimales), 242
Frecuencia, bits de reloj, 96
Función (FC)
Llamar bloques lógicos en el programa de usuario, 177
Números válidos de FC, 67
Programas lineales y estructurados, 176
Protección de know-how, 167
Vista general, 67, 180
Función de trazas, 1234
Funcionalidad, I-device, 736
Funciones nuevas, 35
FUP (Diagrama de funciones), 189

G

Gen_UsrMsg (crear avisos de diagnóstico de usuario), 414
Generar DB de páginas web definidas por el usuario, 908
GEO2LOG (Determinar la ID de hardware a partir del slot), 524
GEOADDR, 530
GET (leer datos de una CPU remota), 795
Configurar la conexión, 598
GET_DIAG (leer información de diagnóstico), 455
GET_ERROR (consultar error localmente), 304
GET_ERROR_ID (consultar ID de error localmente), 306
Get_Features (consultar funciones avanzadas), 982
Get_IM_Data (leer datos de identificación y mantenimiento), 428
GetBlockName (leer el nombre del bloque), 367
GetInstanceName (leer el nombre de la instancia de bloque), 362
GetInstancePath (consultar el nombre global compuesto de la instancia de bloque), 365
GETIO, 373
GETIO_PART, 375
GETSMCInfo, 462
GetSMCInfo (leer información sobre la SIMATIC Memory Card), 462
GetStationInfo, 435
GetSymbolName (leer nombre de una variable en el parámetro de entrada), 356
GetSymbolPath (consultar el nombre global compuesto de la alimentación del parámetro de entrada), 359
GOTO, SCL, 316
Grado de contaminación, 1251
Grado de contaminación/categoría de sobretensión, 1251

Grado de protección, 1251
Guardar archivos de copia de seguridad, 1241

H

Herramientas online y diagnóstico
  Cargar en estado operativo RUN, 1224
Homologación ATEX, 1247
Homologación CE, 1245
Homologación coreana, 1248
Homologación cULus, 1246
Homologación FM, 1246
Homologación naval, 1248
Homologaciones
  ATEX, 1247
  Australia y Nueva Zelanda - Marcado RCM, 1248
  CE, 1245
  cULus, 1246
  FM, 1246
  Homologación CCCEx, 1247
  Homologación coreana, 1248
  Marítima, 1248
Hora
  configurar CPU online, 1204
  RD_LOC_T (leer hora local), 328
  RD_SYS_T (leer la hora), 328
  SET_TIMEZONE (ajustar zona horaria), 331
  T_COMBINE (combinar tiempos), 327
  T_CONV (convertir y extraer tiempos), 325
  T_DIFF (diferencia de tiempo), 327
  WR_LOC_T (escribir hora local), 328
  WR_SYS_T (ajustar la hora), 328
Hora local
  RD_LOC_T (leer hora local), 328
  WR_LOC_T (escribir hora local), 328
Hotline, 3
HSC (contador rápido)
  Configuración, 555
  Fase de operación, 557
  Modos de contaje, 556
HTA (convertir número hexadecimal en cadena de caracteres ASCII), 346

I

Identificación de la CPU, visualización con servidor web, 864
I-device (dispositivo IO inteligente)
  Compartido, 753
  Configurar, 743
  Configurar con un archivo GSD, 745
  Funcionalidad, 736
  Propiedades, 737
  Sistema PN IO subordinado, 737
Idiomas, páginas web definidas por el usuario, 924
Idle Line, 947, 948
IF-THEN, SCL, 310
Imagen de proceso
  estado, 1216, 1222
  Forzar permanentemente, 1222
  observar, 1216
  Operación de forzado permanente, 1222
  Vigilancia de estado o valor, 1215
Impulso (TP), 217
IN_Range (valor dentro del rango), 232
INC (Incrementar), 239
Indexación de matrices con variables, 265
Indicador de primer ciclo, 96
Indicadores LED
  Estado de la CPU, 1197
  interfaz de comunicación, 1197
  Interfaz de comunicación, 937
Índice de variable para una matriz, 265
Información de contacto, 3, 152
Información del programa
  En la estructura de llamadas, 207
Inicializar temporizador (RT), 217
Iniciar
  Trazas, 1234
Inicio/cierre de sesión, páginas web estándar, 860
Inmunidad a sobretensiones, 1250
INSERT (insertar caracteres en una cadena), 353
Insertar la Memory Card en la CPU, 120
Insertar un dispositivo
  CPU sin especificar, 140
Instalación
  Cable de ampliación, 57
  Cargas de lámparas, 62
  Cargas inductivas, 63
  Conector de bloque de terminales, 56
  Corriente disponible, 44
  CPU, 49
  Dimensiones de montaje, 47
  Directrices, 43
  Directrices de aislamiento, 60
  Directrices de cableado, 59, 61
  Espacio, 44
  Módulo de comunicación (CM), 55
  Módulo de señales (SM), 54
  Módulos de señales (SM), 32
  Placa de comunicación (CB), 52
  Puesta a tierra, 61
  Refrigeración, 44

Requisitos, 37
Signal Board (SB), 52
Ventilación, 44
Vista general, 43, 48
Zona de disipación, 44, 47
Instantánea de valores del DB, 1217
Instrucción Modbus_Comm_Load (Configurar SIPLUS I/O o un puerto en el módulo PtP para Modbus RTU), 1087
Instrucciones
-(/)- (bobina normalmente cerrada), 211
-(·)- (bobina normalmente abierta), 211
-(N)- (activar operando con flanco de señal descendente), 215
-(P)- (activar operando con flanco de señal ascendente), 215
-(RESET_BF) (desactivar mapa de bits), 213
-(SET_BF) (activar mapa de bits), 213
-|/|- (contacto normalmente cerrado), 209, 266, 270, 275, 279
-||- (contacto normalmente abierto), 209
-|N|- (consultar flanco de señal descendente de un operando), 214
-|P|- (consultar flanco de señal ascendente de un operando), 214
ABS (calcular valor absoluto), 240
ACOS (calcular arcocoseno), 242
ACT_TINT (activar alarma horaria), 409
Activar salida, 212
ADD (sumar), 237
AND (operación lógica Y), 317
ASIN (calcular arcoseno), 242
ATAN (calcular arcotangente), 242
ATH (convertir cadena de caracteres ASCII en número hexadecimal), 346
ATTACH (asignar OB a evento de alarma), 400
ATTR_DB (leer atributos de un bloque de datos), 521
CALCULATE (calcular), 236
Calendario, 325
CAN_DINT (anular alarma de retardo), 410
CAN_TINT (anular alarma horaria), 408
CASE (SCL), 311
CEIL (redondear un número en coma flotante al siguiente entero superior), 289
Chars_TO_Strg (convertir Array of CHAR en cadena de caracteres), 344
Códigos de estado Legacy USS, 1139
Códigos de estado USS, 1008
Columnas y encabezados, 631, 644, 651, 667, 997, 1016, 1086, 1128, 1141, 1158
Comparar valores, 231
CONCAT (agrupar cadenas de caracteres), 350
CONTINUE (SCL), 315
Control del programa (SCL), 309
CONV (convertir valor), 284
COS (calcular coseno), 242
CountOfElements (Consultar número de elementos de ARRAY), 263
CREATE_DB (crear bloque de datos), 514
CTD (decrementar contador), 226
CTRL_HSC (controlar contadores rápidos), 566
CTRL_HSC_EXT (controlar contadores rápidos (avanzado)), 540
CTRL_PTO (tren de impulsos), 468
CTRL_PWM (modulación del ancho de impulso), 466
CTU (incrementar contador), 226
CTUD (incrementar y decrementar contador), 226
cuadro & (operación lógica Y FUP), 210
cuadro /= (asignación negativa FUP), 211
cuadro = (asignación FUP), 211
cuadro >=1 (operación lógica O FUP), 210
Cuadro N= y bobina N (activar operando con flanco de señal descendente), 215
Cuadro P= y bobina P (activar operando con flanco de señal ascendente), 215
cuadro x (operación lógica O-exclusiva FUP), 210
DataLogClose (cerrar Data Log), 500
DataLogCreate (crear registro), 491
DataLogNewFile (Data Log en archivo nuevo), 503
DataLogOpen (abrir Data Log), 495
DataLogWrite (escribir Data Log), 497
DB_ANY_TO_VARIANT (convertir DB_ANY en VARIANT), 294
DEC (Decrementar), 239
DECO (descodificar), 318
DELETE (borrar caracteres de una cadena), 352
DELETE_DB (borrar bloque de datos), 523
DEMUX (desmultiplexar), 321
Desactivar salida, 212
Deserializar, 248
DETACH (deshacer asignación entre OB y evento de alarma), 400
DeviceStates (leer la información de estado del módulo de un sistema E/S), 445
DIS_AIRT (retardar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad), 413
DIV (dividir), 237
DPNRM_DG (leer datos de diagnóstico de un esclavo DP), 396

DPRD_DAT (leer datos coherentes de un esclavo DP normalizado), 389
DPWR_DAT (escribir datos coherentes de un esclavo DP normalizado), 389
E/S descentralizadas AS-i, 369
E/S descentralizadas PROFIBUS, 369
E/S descentralizadas PROFINET, 369
EN_AIRT (habilitar tratamiento de eventos de alarma y errores asíncronos de mayor prioridad), 413
ENCO (codificar), 318
ENDIS_PW (limitar y habilitar legitimación de la contraseña), 300
EQ_ElemType (Comparar el tipo de datos de un elemento ARRAY para EQUAL con el tipo de datos de una variable), 234
EQ_Type (Comparar el tipo de datos para EQUAL con el tipo de datos de una variable), 234
estado, 1216
EXIT (SCL), 315
EXP (calcular valor exponencial), 242
EXPT (elevar a potencia), 242
F_TRIG (activar variable con flanco de señal descendente), 216
Fecha, 325
FieldRead (leer campo), 264
FieldWrite (escribir campo), 264
FileDelete, 537
FileReadC, 532
FileWriteC, 534
FILL_BLK (rellenar área), 253
FIND (buscar caracteres en una cadena), 355
FLOOR (redondear un número en coma flotante al siguiente entero inferior), 289
FOR (SCL), 312
FRAC (determinar decimales), 242
Gen_UsrMsg (crear avisos de diagnóstico de usuario), 414
GEO2LOG (Determinar la ID de hardware a partir del slot), 524
GET (leer datos de una CPU remota), 795
GET_DIAG (leer información de diagnóstico), 455
GET_ERROR (consultar error localmente), 304
GET_ERROR_ID (consultar ID de error localmente), 306
Get_Features (consultar funciones avanzadas), 982
Get_IM_Data (leer datos de identificación y mantenimiento), 428
GetBlockName (leer el nombre del bloque), 367
GetInstanceName (leer el nombre de la instancia de bloque), 362
GetInstancePath (consultar el nombre global compuesto de la instancia de bloque), 365
GETSMCInfo, 462
GetSMCInfo (leer información sobre la SIMATIC Memory Card), 462
GetSymbolName (leer nombre de una variable en el parámetro de entrada), 356
GetSymbolPath (consultar el nombre global compuesto de la alimentación del parámetro de entrada), 359
GOTO (SCL), 316
HTA (convertir número hexadecimal en cadena de caracteres ASCII), 346
IF-THEN (SCL), 310
IN_Range (valor dentro del rango), 232
INC (Incrementar), 239
INSERT (insertar caracteres en una cadena), 353
Instrucciones de conversión de SCL, 285
Instrucciones Legacy TCON, TDISCON, TSEND y TRCV, 668
INV (complemento a 1), 318
IO2MOD (determinar la ID de hardware a partir de una dirección ES), 527
IS_ARRAY (Consultar si es un ARRAY), 235
IS_NULL (Consultar si es un puntero IGUAL A CERO), 235
JMP (saltar si RLO = 1), 295
JMP_LIST (definir lista de saltos), 296
JMPN (saltar si RLO = 0), 295
LABEL (etiqueta), 295
LED (leer estado del LED), 427
LEFT (leer los caracteres izquierdos de una cadena), 351
LEN (determinar la longitud de una cadena de caracteres), 349
LIMIT (ajustar valor límite), 242
LN (calcular logaritmo natural), 242
LOG2GEO (determinar el slot a partir de la ID de hardware), 526
LOWER_BOUND (leer límite inferior del ARRAY), 255
MAX (determinar máximo), 241
MAX_LEN (longitud máxima de una cadena de caracteres), 349
MB_CLIENT, 1017
MB_RED_CLIENT, 1039
MB_RED_SERVER, 1057
MC_ChangeDynamic, 578
MC_CommandTable, 576
MC_Halt (detener eje), 574
MC_Home (referenciar eje), 573

MC_MoveAbsolute (posicionamiento absoluto del eje), 574
MC_MoveJog (desplazamiento del eje en modo Jog), 576
MC_MoveRelative (posicionamiento relativo del eje), 575
MC_MoveVelocity (mover el eje a la velocidad predefinida), 575
MC_Power (habilitar/bloquear eje), 572
MC_ReadParam (leer parámetros de un objeto tecnológico), 577
MC_Reset (confirmar error), 573
MC_WriteParam (escribir parámetros de un objeto tecnológico), 577
MID (leer los caracteres centrales de una cadena), 351
MIN (determinar mínimo), 241
MOD (obtener resto de división), 238
Modbus_Comm_Load (Configurar SIPLUS I/O o un puerto en el módulo PtP para Modbus RTU), 1087
Modbus_Master (Comunicar vía SIPLUS I/O o el puerto PtP como maestro Modbus RTU), 1091
Modbus_Slave (Comunicar vía SIPLUS I/O o el puerto PtP como esclavo Modbus RTU), 1098
ModuleStates (leer información de estado del submódulo de un módulo), 450
Motion Control, 571
MOVE (copiar valor), 244
MOVE_BLK (copiar área), 244
MUL (multiplicar), 237
MUX (multiplexar), 320
N (consultar flanco de señal descendente de un operando), 214
N_TRIG (consultar flanco de señal descendente del RLO), 215
NE_ElemType (Comparar el tipo de datos para UNEQUAL con el tipo de datos de una variable), 234
NE_Type (Comparar el tipo de datos para UNEQUAL con el tipo de datos de una variable), 234
NEG (generar complemento a dos), 239
NORM_X (normalizar), 290
NOT (invertir RLO), 211
NOT_NULL (Consultar si es un puntero DIFERENTE DE CERO), 235
NOT_OK (comprobar invalidez), 233
observar, 1216
OK (comprobar validez), 233
Operación de forzado permanente, 1222
OR (operación lógica O), 317
OUT_Range (valor fuera del rango), 232
P (consultar flanco de señal ascendente de un operando), 214
P_TRIG (consultar flanco de señal ascendente del RLO), 215
P3964_Config (configurar el protocolo 3964(R)), 971
Parámetros comunes, 723
PID_Compact (controlador PID universal con ajuste integrado), 580
PID_Temp (regulador PID universal que permite manejar el control de temperatura), 582
Port_Config (configuración de puerto), 961
PRVREC (proveer registro), 394
PUT (escribir datos en una CPU remota), 795
QRY_CINT (consultar parámetros de alarma cíclica), 405
QRY_DINT (consultar estado de una alarma de retardo), 410
QRY_TINT (consultar estado de alarma horaria), 409
R (desactivar salida), 212
R_TRIG (activar variable con flanco de señal ascendente), 216
RALRM (recibir alarma), 377
RCVREC (recibir registro), 392
RD_ADDR (determinar las direcciones ES a partir de la ID de hardware), 529
RD_LOC_T (leer hora local), 328
RD_SYS_T (leer la hora), 328
RDREC (leer registro), 370
RE_TRIGR, 91
RE_TRIGR (reiniciar tiempo de vigilancia del ciclo), 303
READ_BIG (Leer datos en formato Big-Endian), 259
READ_DBL (leer de un bloque de datos de la memoria de carga), 518
READ_LITTLE (Leer datos en formato Little-Endian), 259
Receive_Config (configuración de recepción), 966
Receive_P2P (recepción punto a punto), 976
Receive_Reset (inicializar receptor), 978
RecipeExport (exportar receta), 483
RecipeImport (importar receta), 485
Reloj, 328
REPEAT (SCL), 314
REPLACE (reemplazar caracteres de una cadena), 354
RESET_BF (desactivar mapa de bits), 213
RET (retroceder), 299
RETURN (SCL), 316

RIGHT (leer los caracteres derechos de una cadena), 351
ROL (rotar a la izquierda) y ROR (rotar a la derecha), 323
ROUND (redondear número), 288
RS (flipflop de desactivación/activación), 213
RT (inicializar temporizador), 217
RTM (contador de horas de funcionamiento), 332
RUNTIME (Medir tiempo de ejecución), 307
S (activar salida), 212
S_CONV (convertir cadena de caracteres), 335
S_MOVE (desplazar cadena de caracteres), 335
SCALE_X (escalar), 290
SEL (seleccionar), 319
Send_Config (configuración de transmisión), 964
Send_P2P (enviar datos punto a punto), 973
Serializar, 250
SET_BF (activar mapa de bits), 213
SET_CINT (ajustar parámetros de alarma cíclica), 403
Set_Features (activar funciones avanzadas), 983
SET_TIMEZONE (ajustar zona horaria), 331
SET_TINTL (ajustar alarma horaria), 407
SGN_GET (consultar señales RS232), 979
SHL (desplazar a la izquierda) y SHR (desplazar a la derecha), 322
Signal_Set (activar señales RS-232), 980
SIN (calcular seno), 242
SQR (calcular cuadrado), 242
SQRT (calcular raíz cuadrada), 242
SR (flipflop de activación/desactivación), 213
SRT_DINT (iniciar alarma de retardo), 410
STP (finalizar programa), 304
Strg_TO_Chars (convertir cadena de caracteres en Array of CHAR), 344
STRG_VAL (convertir cadena de caracteres en valor numérico), 335
SUB (restar), 237
SWAP (cambiar disposición), 254
SWITCH (distribuidor de saltos), 297
T_ADD (sumar tiempos), 326
T_COMBINE (combinar tiempos), 327
T_CONFIG (configurar interfaz), 712
T_CONV (convertir y extraer tiempos), 325
T_DIAG, 678
T_DIFF (diferencia de tiempo), 327
T_RESET, 676
T_SUB (restar tiempos), 326
TAN (calcular tangente), 242
TCON, 652
TDISCON, 652
Temporizador, 217
Time, 325
TM_MAIL (transmitir e-mail), 1189
TOF (retardo al desconectar), 217
TON (retardo al conectar), 217
TONR (acumulador de tiempo), 217
TP (impulso), 217
TRCV, 652
TRCV_C, 631, 731
TRCV_C Legacy (recibir datos vía Ethernet (TCP)), 644
TRUNC (truncar a entero), 288
TSEND, 652
TSEND_C, 631, 730
TSEND_C Legacy (enviar datos vía Ethernet (TCP)), 644
TURCV (recibir datos vía Ethernet (UDP)), 707
TUSEND (enviar datos vía Ethernet (UDP)), 707
UFILL_BLK (rellenar área sin interrupciones), 253
UMOVE_BLK (copiar área sin interrupciones), 244
UPPER_BOUND (leer límite superior de ARRAY), 256
USS_Drive_Control (intercambiar datos con el accionamiento), 1002
USS_Port_Scan (procesar comunicación vía red USS), 1000
USS_Read_Param (leer parámetros del accionamiento), 1005
USS_Write_Param (modificar parámetros en el accionamiento), 1006
VAL_STRG (convertir valor numérico en cadena de caracteres), 335
VARIANT_TO_DB_ANY (convertir VARIANT en DB_ANY), 293
Variantes de PEEK y POKE, 195, 258
VariantGet (Leer valor de una variable VARIANT), 261
VariantPut (Escribir valor en una variable VARIANT), 262
Versiones de instrucciones, 631, 644, 651, 667, 997, 1016, 1086, 1128, 1141, 1158
Vigilancia de estado o valor, 1215
WHILE (SCL), 313
WR_LOC_T (escribir hora local), 328
WR_SYS_T (ajustar la hora), 328
WRIT_DBL (escribir en un bloque de datos de la memoria de carga), 518
WRITE_BIG (Escribir datos en formato Big-Endian), 259
WRITE_LITTLE (Escribir datos en formato Little-Endian), 259
WRREC (escribir registro), 370

WWW (sincronización de páginas web definidas por el usuario), 909
XOR (operación lógica O-exclusiva), 317
Instrucciones antiguas
MB_CLIENT (comunicar como cliente TCP Modbus vía PROFINET), 1142
MB_COMM_LOAD (configurar puerto en módulo PtP para Modbus-RTU), 1159
MB_MASTER (comunicar como maestro Modbus vía puerto PtP), 1162
MB_SERVER (comunicar como servidor TCP Modbus vía PROFINET), 1149
MB_SLAVE (comunicar como esclavo Modbus vía puerto PtP), 1168
PORT_CFG (configurar parámetros de comunicación dinámicamente), 1112
RCV_CFG (configurar parámetros de recepción serie dinámicamente), 1115
RCV_PTP (habilitar recepción de mensajes), 1122
RCV_RST (borrar búfer de recepción), 1124
SEND_CFG (configurar parámetros de transmisión serie dinámicamente), 1114
SEND_PTP (transmitir datos del búfer de transmisión), 1120
SGN_GET (consultar señales RS232), 1125
SGN_SET (activar señales RS232), 1126
USS_DRV (intercambiar datos con el accionamiento), 1133
USS_PORT (procesar comunicación vía red USS), 1131
USS_RPM (leer parámetros del accionamiento), 1136
USS_WPM (modificar parámetros en el accionamiento), 1137
Instrucciones de flanco ascendente y descendente, 214
Instrucciones de Motion control, 571
Instrucciones Legacy TCON, TDISCON, TSEND y TRCV, 668
Instrucciones PROFINET
Instrucciones Legacy TCON, TDISCON, TSEND y TRCV, 668
T_CONFIG (configurar interfaz), 712
T_DIAG, 678
T_RESET, 676
TCON, 652
TDISCON, 652
TRCV, 652
TRCV_C, 631, 731
TRCV_C Legacy (recibir datos vía Ethernet (TCP)), 644
TSEND, 652
TSEND_C, 631
TSEND_C Legacy (enviar datos vía Ethernet (TCP)), 644
TURCV (recibir datos vía Ethernet (UDP)), 707
TUSEND (enviar datos vía Ethernet (UDP)), 707
Instrucciones tecnológicas, 540, 566
Instrucciones trigonométricas, 242
Intercambio de datos entre sistemas IO, 740
Interfaces de comunicación
Agregar módulos, 142
CB 1241 RS485, 1419
CM 1241 RS232, 1421
Configuración, 943
Configuración de dispositivos, 137
Indicadores LED, 1197
Programación, 984
RS232 y RS485, 937
Tabla de comparación de módulos, 31
Interfaces de comunicación, 3964(R), 956
Interfaz de bloque extendida
Cargar en estado operativo RUN, 1229
Interfaz de usuario
Vistas del proyecto y del portal de STEP 7, 39
INV (complemento a 1), 318
IO2MOD (determinar la ID de hardware a partir de una dirección ES), 527
IO-Link
Almacenamiento de dispositivos, 1371
Asignación de los pines, 1371
Cambiar parámetros en tiempo de ejecución, 1375
Configurar, 1374
Diagnóstico, 1378
Espacio de direcciones, 1374
Esquema, 1373
Funciones, 1370
Indicador LED, 1377
Mensajes de error, 1375, 1377, 1378
Parámetros, 1374
Perfil de dispositivo, 1369
Registro, 1375
Restablecer los ajustes de fábrica, 1370
Sustituir, 1370
IS_ARRAY (Consultar si es un ARRAY), 235
IS_NULL (Consultar si es un puntero IGUAL A CERO), 235
ISO on TCP
Configuración de la conexión, 597
ID de conexión, 616
Modo ad hoc, 615
Parámetros, 618

J

JavaScript, páginas web estándar, 934
JMP (saltar si RLO = 1), 295
JMP_LIST (definir lista de saltos), 296
JMPN (saltar si RLO = 0), 295

K

KOP (esquema de contactos)
editor de programas, 1216
estado, 1216
observar, 1216
KOP (Esquema de contactos)
estado, 1222
Vigilancia de estado o valor, 1215
Vista general, 188

L

LABEL (etiqueta), 295
Latencia, 87
Lectura de DBs, E/S o memoria, 195, 258
LED (leer estado del LED), 427
Leer variables HTTP, 897
LEFT (leer los caracteres izquierdos de una cadena), 351
LEN (determinar la longitud de una cadena de caracteres), 349
Librería del protocolo Legacy USS
Códigos de estado, 1139
Requisitos de uso, 1129
USS_DRV (intercambiar datos con el accionamiento), 1133
USS_PORT (procesar comunicación vía red USS), 1131
USS_RPM (leer parámetros del accionamiento), 1136
USS_WPM (modificar parámetros en el accionamiento), 1137
Vista general, 1127
Librería del protocolo USS
Códigos de estado, 1008
Requisitos de uso, 998
USS_Drive_Control (intercambiar datos con el accionamiento), 1002
USS_Port_Scan (procesar comunicación vía red USS), 1000
USS_Read_Param (leer parámetros del accionamiento), 1005
USS_Write_Param (modificar parámetros en el accionamiento), 1006
Vista general, 994
Librería global
Vista general del protocolo Legacy USS, 1127
Vista general del protocolo USS, 994
Licencias OPC UA, 1446
Licencias, OPC UA, 1446
LIMIT (ajustar valor límite), 242
Limitaciones
Páginas web definidas por el usuario, 911
Servidor web, 932
Llamada de métodos OPC UA, 837
Llamar bloques lógicos en el programa de usuario, 177
LN (calcular logaritmo natural), 242
LOG2GEO (determinar el slot a partir de la ID de hardware), 526
Lógica con bits
Bobinas normalmente abiertas y cerradas, 211
Contactos normalmente abiertos y cerrados, 209
Instrucción NOT (invertir RLO), 211
Instrucciones de activación y desactivación, 212
Instrucciones de flanco ascendente y descendente, 215
Instrucciones Y, O y O-exclusiva, 210
Longitud fija, 953
Longitud máxima del mensaje, 953
Longitud, mensaje PtP, 954
LOWER_BOUND (leer límite inferior del ARRAY), 255

M

Manuales, 4
Máquinas flexibles (control de configuración), 143
Máscara de subred, 604
Matemáticas, 236, 237
Matrices, acceso a los miembros, 265
MAX (determinar máximo), 241
MAX_LEN (longitud máxima de una cadena de caracteres), 349
MB_CLIENT, 1017
MB_CLIENT (comunicar como cliente TCP Modbus vía PROFINET), antigua, 1142
MB_COMM_LOAD (configurar puerto en módulo PtP para Modbus RTU), antigua, 1159
MB_MASTER (comunicar como maestro Modbus vía puerto PtP), antigua, 1162
MB_RED_CLIENT, 1039
MB_RED_SERVER, 1057
MB_SERVER, 1028

MB_SERVER (comunicar como servidor TCP Modbus vía PROFINET), antigua, 1149
MB_SLAVE (comunicar como esclavo Modbus vía puerto PtP), antigua, 1168
MC_ChangeDynamic (cambiar la configuración de la dinámica del eje), 578
MC_CommandTable, 576
MC_Halt (detener eje), 574
MC_Home (referenciar eje), 573
MC_MoveAbsolute (posicionamiento absoluto del eje), 574
MC_MoveJog (desplazamiento del eje en modo Jog), 576
MC_MoveRelative (posicionamiento relativo del eje), 575
MC_MoveVelocity (mover el eje a la velocidad predefinida), 575
MC_Power (habilitar/bloquear eje), 572
MC_ReadParam (leer parámetros de un objeto tecnológico), 577
MC_Reset (confirmar error), 573
MC_WriteParam (escribir parámetros de un objeto tecnológico), 577
Mediciones, órdenes de seguimiento (trace), 1235
Memoria
  Direcciones de entradas de periferia (tabla de forzado permanente), 1222
  I (memoria imagen de proceso de las entradas), 101
  L (memoria local), 99
  M (área de marcas), 103
  Marcas de ciclo, 95
  Marcas de sistema, 95
  Memoria de carga, 93
  Memoria de trabajo, 93
  Memoria remanente, 93
  Memoria temporal, 103
  Q (memoria imagen de proceso de las salidas), 102
  vigilar la carga de la memoria, 1211
Memoria de carga, 28
  CPU 1211C, 1255
  CPU 1212C, 1268
  CPU 1214C, 1280
  CPU 1215C, 1292
  CPU 1217C, 1307
  Páginas web definidas por el usuario, 911
Memoria de trabajo, 28
  CPU 1211C, 1255
  CPU 1212C, 1268
  CPU 1214C, 1280
  CPU 1215C, 1292
  CPU 1217C, 1307
Memoria I
  Direcciones de entradas de periferia (tabla de forzado permanente), 1222
  Forzar permanentemente, 1222
  observar, 1215
  observar KOP, 1216
  Operación de forzado permanente, 1222
  Tabla de forzado permanente, 1222
  tabla de observación, 1215
Memoria imagen de proceso
  Escribir salidas con SETIO, 374
  Leer el área de la memoria imagen de proceso con GETIO_PART, 375
  Leer entradas con GETIO, 373
  Transferir el área de la memoria imagen de proceso con SETIO_PART, 376
Memoria local
  Máxima por nivel de prioridad del OB, 103
  Uso por parte de bloques, 103
Memoria Q
  Configurar canales de impulsos, 473
  Salidas de impulsos, 471
Memoria remanente, 28, 93
  CPU 1211C, 1255
  CPU 1212C, 1268
  CPU 1214C, 1280
  CPU 1215C, 1292
  CPU 1217C, 1307
Memoria temporal
  Máxima por nivel de prioridad del OB, 103
  Uso por parte de bloques, 103
Memory Card, 1424
  Configurar los parámetros de arranque, 122
  Contraseña olvidada, 135
  Error de incompatibilidad, 1199
  Insertar en la CPU, 120
  Tarjeta de transferencia, 123
  Tarjeta de transferencia vacía para una contraseña olvidada, 135
Memory Card de CPU
  Insertar, 120
  Tarjeta de transferencia, 123
Mensaje
  Finalizar, 951
  Iniciar, 948
  Longitud, 953
MID (leer los caracteres centrales de una cadena), 351
MIN (determinar mínimo), 241
MOD (obtener resto de división), 238

Modbus
Códigos de función, 1013
Comunicación RTU, 1014
Direcciones de estación de red, 1014
Direcciones de memoria, 1014
MB_CLIENT (comunicar como cliente TCP Modbus vía PROFINET), antigua, 1142
MB_COMM_LOAD (configurar puerto en módulo PtP para Modbus RTU), antigua, 1159
MB_MASTER (comunicar como maestro Modbus vía puerto PtP), antigua, 1162
MB_SERVER (comunicar como servidor TCP Modbus vía PROFINET), antigua, 1149
MB_SLAVE (comunicar como esclavo Modbus vía puerto PtP), antigua, 1168
Modbus_Comm_Load (Configurar SIPLUS I/O o un puerto en el módulo PtP para Modbus RTU), 1087
Modbus_Master (Comunicar vía SIPLUS I/O o el puerto PtP como maestro Modbus RTU), 1091
Modbus_Slave (Comunicar vía SIPLUS I/O o el puerto PtP como esclavo Modbus RTU), 1098
Versiones, 997, 1086, 1128, 1158
MODBUS
MB_CLIENT, 1017
MB_RED_CLIENT, 1039
MB_RED_SERVER, 1057
MB_SERVER, 1028
Modbus RTU
Ejemplo de esclavo, 1111
Programa maestro, 1108
Modbus TCP
Versiones, 1016, 1141
Modbus_Master (Comunicar vía SIPLUS I/O o el puerto PtP como maestro Modbus RTU), 1091
Modbus_Slave (Comunicar vía SIPLUS I/O o el puerto PtP como esclavo Modbus RTU), 1098
Modificar
estado del editor de programas, 1216
Tabla de observación, 1218
Variables desde el servidor web, 875
Modo ad hoc, TCP e ISO on TCP, 615
Modo ARRANQUE
Operación de forzado permanente, 1222
Modo para comunicación PG/PC y HMI, 160
Modo RUN
Operación de forzado permanente, 1222
Modo STOP
Operación de forzado permanente, 1222
Modos de contaje
Contador rápido, 556
ModuleStates, 450
Módulo CP
Acceso al servidor web, 853
Página de inicio del servidor Web, 863
Módulo de alimentación
PM1207, 1428
Módulo de alimentación PM 1207, 1428
Módulo de comunicación (CM)
Agregar módulo CM 1243-5 (maestro DP), 781
Agregar módulo maestro AS-i CM1243-2, 785
Agregar módulos, 142
CM 1241 RS232, 1421
CM 1241 RS422/RS485, 1422
Configuración de dispositivos, 137
Configuración de parámetros, 169
Configuración del programa de ejemplo PtP, 987
Consumo de corriente, 1431
Desmontaje, 55
Indicadores LED, 937, 1197
Instalación, 55
Programación, 984
Recepción de datos, 976, 1122
RS232 y RS485, 937
Tabla de comparación, 31
Vista general, 32
Módulo de señales IO-Link Master, 1366
Módulo del potenciómetro
Datos técnicos, 1426
Módulo tecnológico, SM 1278 4xIO-Link Master, 1366
Módulos
Communication Boards (CB), 31
Configurar parámetros, 169
Módulo de comunicación (CM), 32
Módulos de señales (SM), 32
Procesador de comunicaciones (CP), 32
Signal Board (SB), 31
Tabla de comparación, 31
Zona de disipación, 44, 47
Módulos CANopen
021620-B, 021630-B, 1428
Módulos de comunicación RS232 y RS485, 937
Módulos de señales
SM 1222 DQ 16 x 24 V DC sumidero, 1326
Módulos de señales (SM)
Agregar módulos, 142
Cable de ampliación, 57
Configuración de parámetros, 169
Consumo de corriente, 1431
Desmontaje, 55
Instalación, 54
Representación de entradas analógicas (intensidad), 1352, 1396

Representación de entradas analógicas (tensión), 1352, 1395
Representación de salidas analógicas (intensidad), 1353, 1397
Representación de salidas analógicas (tensión), 1353, 1396
SM 1221 DI 16 x 24 V DC, 1323
SM 1221 DI 8 x 24 V DC, 1323
SM 1222 DQ 16 x 24 V DC, 1326
SM 1222 DQ 16 x Relay, 1326
SM 1222 DQ 8 Relay Changeover, 1325
SM 1222 DQ 8 x 24 V DC, 1325
SM 1222 DQ 8 x Relay, 1325
SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1332
SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC sinking, 1332
SM 1223 DI 16 x 24 V DC, DQ 16 x Relay, 1332
SM 1223 DI 8 x 120/230 V AC/DQ 8 x relé, 1339
SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1332
SM 1223 DI 8 x 24 V DC, DQ 8 x Relay, 1332
SM 1231 AI 4 x 13 bits, 1341
SM 1231 AI 4 x 16 bits, 1341
SM 1231 AI 4 x 16 bits TC, 1354
SM 1231 AI 4 x RTD x 16 bits, 1360
SM 1231 AI 8 x 13 bits, 1341
SM 1231 AI 8 x 16 bits TC, 1354
SM 1231 AI 8 x RTD x 16 bits, 1360
SM 1232 AQ 2 x 14 bit, 1346
SM 1232 AQ 4 x 14 bit, 1346
SM 1234 AI 4 x 13 bits / AQ 2 x 14 bits, 1348
SM 1278 4xIO-Link Master, 1366
Tiempos de respuesta a un escalón, 1351
Vista general, 32
Módulos de señales analógicos
SM 1231, 1341
SM 1231 de termopar, 1354
SM 1231 RTD, 1360
SM 1232, 1346
SM 1234, 1348
Módulos de señales digitales
SM 1221, 1323
SM 1222, 1325, 1326
SM 1223, 1332, 1339
Montaje
Aislamiento, 60
Cable de ampliación, 57
Cargas de lámparas, 62
Cargas inductivas, 63
Conector de bloque de terminales, 56
CPU, 49
Dimensiones, 47
Directrices, 43
Directrices de cableado, 59, 61
Espacio, 44
Módulo de comunicación (CM), 55
Módulo de señales (SM), 54
Placa de comunicación (CB), 52
Puesta a tierra, 61
Refrigeración, 44
Signal Board (SB), 52
Ventilación, 44
Vista general, 48
Zona de disipación, 44, 47
Motion Control
MC_CommandTable, 576
MC_Halt (detener eje), 574
MC_Home (referenciar eje), 573
MC_MoveAbsolute (posicionamiento absoluto del eje), 574
MC_MoveJog (desplazamiento del eje en modo Jog), 576
MC_MoveRelative (posicionamiento relativo del eje), 575
MC_MoveVelocity (mover el eje a la velocidad predefinida), 575
MC_Power (habilitar/bloquear eje), 572
MC_ReadParam (leer parámetros de un objeto tecnológico), 577
MC_Reset (confirmar error), 573
MC_WriteParam (escribir parámetros de un objeto tecnológico), 577
Vista general, 569
MOVE (copiar valor), 244
MOVE_BLK (copiar área), 244
MUL (multiplicar), 237
MUX (multiplexar), 320

N

N (consultar flanco de señal descendente de un operando), 214
N_TRIG (consultar flanco de señal descendente del RLO), 215
Navegadores soportados para el servidor web, 845
NE_ElemType (Comparar el tipo de datos para UNEQUAL con el tipo de datos de una variable), 234
NE_Type (Comparar el tipo de datos para UNEQUAL con el tipo de datos de una variable), 234
NEG (generar complemento a dos), 239
Network Time Protocol (NTP), 611
Nivel de protección
Bloque lógico, 167
Contraseña olvidada, 135

CPU, 163
Enlazar a una CPU, Memory Card o contraseña, 168
Nombres de dispositivo online
PROFINET IO, 1202
NORM_X (normalizar), 290
Normalizar valores analógicos, 291
NOT (invertir RLO), 211
NOT_NULL (Consultar si es un puntero DIFERENTE DE CERO), 235
NOT_OK (comprobar invalidez), 233
Nuevas funciones, 35
Números
Binario, 107
Entero, 108
Real, 109
Números de puerto
Asignación a interlocutores, 613
Restringidos, 724

O

OB, (Bloque de organización)
OB de actualización, 84
OB de alarma cíclica, 77
OB de alarma de error de diagnóstico, 80
OB de alarma de error de tiempo, 79
OB de alarma de proceso, 78
OB de arranque, 76
OB de ciclo, 75
OB de estado, 84
OB de fallo del rack o estación, 83
OB de hora, 84
OB de perfil, 85
OB de presencia de módulo, 82
OB de retardo, 76
OB MC-PostServo, 86
OB MC-PreServo, 85
Observar
Estado KOP, 1216
Estado KOP y uso de la tabla de observación, 1215
Obtener y restablecer valores del DB, 1217
Operación de forzado permanente, 1222
Tabla de forzado permanente, 1222
Tabla de observación, 1218
Observar el programa, 205
Observar variables desde el servidor web, 875
Obtener valores de un DB online, 1217
OK (comprobar validez), 233
Online
Asignar una dirección IP, 602
búfer de diagnóstico, 1211
carga de la memoria, 1211
comparar y sincronizar, 1213
Dirección IP, 1204
Establecer conexión online, 1201
estado, 1216
Forzar permanentemente, 1222
herramientas, 1215
hora, 1204
Obtener y restablecer valores del DB, 1217
Operación de forzado permanente, 1222
Panel de control, 1210
tabla de observación, 1215, 1216
Tabla de observación, 1218
Tiempo de ciclo, 1211
Vigilancia de estado o valor, 1215
OPC UA
Llamadas de métodos, 837
OPC, configuración, 1185
Open User Communication
Establecer una conexión y enviar datos con TSEND_C Legacy, 644
Establecer una conexión y leer datos con TRCV_C, 631
Establecer una conexión y leer datos con TRCV_C Legacy, 644
Establecer una conexión y transmitir datos con TSEND_C, 631
OR (operación lógica O), 317
OUT_Range (valor fuera del rango), 232

P

P (consultar flanco de señal ascendente de un operando), 214
P_TRIG (consultar flanco de señal ascendente del RLO), 215
P3964_Config (configurar el protocolo 3964(R)), 971
Errores, 972
Página web Estado de variables, 875
Página web estándar de arranque, 863
Página web estándar de comunicación, 872
Página web estándar de diagnóstico, 864
Página web estándar de información del módulo, 867
Página web estándar de introducción, 862
Página web estándar Diagnóstico, 866
Página web estándar Estado de variables, 875
Páginas HTML
Definidas por el usuario, 891
Listado, ejemplo de página web definida por el usuario, 918

Páginas HTML, definidas por el usuario
acceder a datos del S7-1200, 893
Actualizar, 892
Desarrollar, 892
ubicaciones de idiomas, 927
ubicaciones de páginas, 907
Páginas web
Servicio técnico, soporte y documentación de STEP 7, 4
Páginas web de STEP 7, 4
Páginas web definidas por el usuario, 844, 891
acceder desde un PC, 911
Activar con instrucción WWW, 909
Activar y desactivar desde el DB de control, 927
Actualizar, 892
borrar bloques de programa, 908
cargar DBs correspondientes, 910
Comandos AWP para acceder a datos del S7-1200, 893
configuración de idiomas múltiples, 927
Configurar, 907
Control manual de DB de fragmentos, 927
Crear con editor HTML, 892
Crear fragmentos, 903
Ejemplo, 913
Escribir variables, 895
escribir variables especiales, 899
generar bloques de programa, 908
Idiomas múltiples, 924
importar fragmentos, 904
Leer variables, 894
Leer variables especiales, 897
limitaciones de memoria de carga, 911
Listado HTML, 918
Procesamiento de caracteres especiales, 905
Programación en STEP 7, 909
Páginas web estándar, 843
Acceder desde un PC, 851
Acceso seguro, 852
Arranque, 863
cambiar el modo de operación, 863
Comunicación, 872
Diagnóstico, 864, 866
Estado de variables, 875
Información del módulo, 867
Inicio y cierre de sesión, 860
Introducción, 862
JavaScript, 934
Representación, 857
Restricciones de cookies, 934
Paneles (HMI), 32
Paneles Basic (HMI), 32
Paneles de operador, 32
Parametrización, 181
Parámetro BUFFER, SEND_P2P, 975
Parámetro LENGTH, SEND_P2P, 975
Parámetros de arranque, 122
Parámetros de salida, 181
Configurar canales de impulsos, 473
Salidas de impulsos, 471
Paridad, 944
Pausa, 947, 948
PEEK, PEEK_WORD, PEEK_BOOL, PEEK_DWORD, PEEK_BLK, 195, 258
Pérdida de la comunicación de la CPU con los módulos, 82
Perfil DIN, 48
PID
PID_3STEP (controlador PID con ajuste de válvulas), 581
PID_Compact (controlador PID universal con ajuste integrado), 580
PID_Temp (regulador PID universal que permite manejar el control de temperatura), 582
PLC
Agregar módulos, 142
Asignar una dirección IP a una CPU online, 602
Cable de ampliación, 57
Carga de comunicación, 92
Cargar, 199
comparar y sincronizar, 1213
Conector de bloque de terminales, 56
Configuración de dispositivos, 137
Configuración del HSC, 555
Copiar bloques desde una CPU online, 204
Corriente disponible, 44
Diseño del sistema, 175
Estados operativos, 71
Forzar permanentemente, 1222
Instalación, 48, 49
observar, 1215
Operación de forzado permanente, 1222
Procesamiento del arranque, 73
Propiedad de sincronización horaria, 611
Protección de know-how, 167
RTM (contador de horas de funcionamiento), 332
Tabla de observación, 1218
Tiempo de ciclo, 91, 92
Usar bloques, 176
Variables, 99
Vista general de la CPU, 27
POKE, POKE_BOOL, POKE_BLK, 195, 258
PORT_CFG (configurar los parámetros de comunicación dinámicamente), antigua, 1112

Port_Config (configuración de puerto), 961
Posición de los caracteres, longitud del mensaje, 954
Posiciones de memoria, 99, 101
Prioridad
  Clase de prioridad, 75
  Prioridad de procesamiento, 87
Procesador de comunicaciones (CP)
  Agregar módulos, 142
  Configuración de dispositivos, 137
  Configuración de parámetros, 169
  Tabla de comparación, 31
  Vista general, 32
PROFIBUS
  Agregar esclavo DP, 781
  Agregar módulo CM 1243-5 (maestro DP), 781
  Conexión de red, 596, 781
  Conexión S7, 799
  Dirección, 782
  Dirección, configurar, 782
  DPNRM_DG (leer datos de diagnóstico de un esclavo DP), 396
  DPRD_DAT (leer datos coherentes de un esclavo DP normalizado), 389
  DPWR_DAT (escribir datos coherentes de un esclavo DP normalizado), 389
  Esclavo, 777
  GET (leer datos de una CPU remota), 795
  Instrucciones E/S descentralizadas, 369
  Maestro, 777
  Módulo CM 1242-5 (esclavo DP), 778
  Módulo CM 1243-5 (maestro DP), 778
  Número de conexiones de comunicación, 592
  PUT (escribir datos en una CPU remota), 795
  RALRM (recibir alarma), 377
  RDREC (leer registro), 370
  WRREC (escribir registro), 370
PROFIBUS y PROFINET
  Ejemplo de DeviceStates, 446
  Ejemplo de ModuleStates, 452
PROFIenergy, 398
PROFINET
  Asignación de dirección IP, 612
  Asignación de nombre y dirección de dispositivo, 612
  Comprobar una red, 607
  Comunicación de CPU a CPU, 728
  Comunicación entre PLCs, 728
  Conexión de red, 596, 728, 729, 733
  Conexión S7, 799
  configurar la comunicación entre la CPU y el dispositivo HMI, 727
  Configurar la dirección IP, 153
  Dirección IP, 603
  Dirección MAC, 603
  DPRD_DAT (leer datos coherentes de un esclavo DP normalizado), 389
  DPWR_DAT (escribir datos coherentes de un esclavo DP normalizado), 389
  GET (leer datos de una CPU remota), 795
  ID de conexión, 616
  Instrucciones E/S descentralizadas, 369
  Modo ad hoc, 615
  Número de conexiones de comunicación, 592
  Propiedad de sincronización horaria, 611
  Propiedades de direcciones Ethernet, 604
  PRVREC (proveer registro), 394
  PUT (escribir datos en una CPU remota), 795
  RALRM (recibir alarma), 377
  RCVREC (recibir registro), 392
  RDREC (leer registro), 370
  Restablecer una conexión, 676
  Sincronización horaria, 153
  Tiempo de arranque del sistema, 612
  Tipos de comunicación, 585
  Vista general, 613
  WRREC (escribir registro), 370
PROFINET IO
  Agregar un dispositivo, 733
  Asignar nombres de dispositivo, 734
  Asignar nombres de dispositivo online, 1202
  Asignar una CPU, 734
  Dispositivos, 733
  Nombres de dispositivo, 734
  Nombres de dispositivo online, 1202
PROFINET RT, 613
Profundidad de anidamiento, 67
Programa
  Bloques de organización (OB), 178
  Cargar, 199
  Clase de prioridad, 75
  Copiar bloques desde una CPU online, 204
  Enlazar a una CPU, Memory Card o contraseña, 168
  Llamar bloques lógicos en el programa de usuario, 177
  Programas lineales y estructurados, 176
  Protección por contraseña, 167
  SIMATIC Memory Card, 119
Programa de usuario
  Bloques de organización (OB), 178
  Cargar, 199
  Copiar bloques desde una CPU online, 204
  Enlazar a una CPU, Memory Card o contraseña, 168

Llamar bloques lógicos en el programa de usuario, 177
Programas lineales y estructurados, 176
Protección por contraseña, 167
SIMATIC Memory Card, 119
Tarjeta de programa, 119
Tarjeta de transferencia, 119
Programación
Bloque de datos (DB), 67
Bloque de datos (DB) de instancia, 181
Bloque de función (FB), 67, 181
Clase de prioridad, 75
comparar y sincronizar bloques lógicos, 1213
CPU sin especificar, 140
Enlazar a una CPU, Memory Card o contraseña, 168
Estados operativos de la CPU, 71
Flujo de corriente (EN y ENO), 197
Función (FC), 180
FUP (Diagrama de funciones), 189
Hora del sistema, 328
Instrucciones PtP, 984
KOP (Esquema de contactos), 188
Llamadas de bloque, 67
Llamar bloques lógicos en el programa de usuario, 177
Números válidos de FC, FB y DB, 67
PID_3STEP (controlador PID con ajuste de válvulas), 581
PID_Compact (controlador PID universal con ajuste integrado), 580
PID_Temp (regulador PID universal que permite manejar el control de temperatura), 582
Programa estructurado, 176
Programa lineal, 176
RTM (contador de horas de funcionamiento), 332
SCL (Structured Control Language), 189, 190, 191
Tipos de bloques lógicos, 67
Valor inicial de un FB, 181
Programación de STEP 7
Páginas web definidas por el usuario, 909
programa de ejemplo PtP, 992
Programación estructurada, estructura de bloque, 176
Programación lineal, 176
Programación punto a punto, 984
Programar un cambio de idioma para la página web definida por el usuario, 924
Propiedad de sincronización horaria, 611
Propiedades de la CPU, para páginas web definidas por el usuario
Configuración de idiomas múltiples, 927
Configuración de STEP 7, 907
Protección anticopia
Enlazar a una CPU, Memory Card o contraseña, 168
Protección contra inversión de polaridad, 1252
Protección de acceso, CPU, 163
Protección de datos de configuración confidenciales del PLC, 160
Protección de know-how
Protección por contraseña, 167
Protección de know-how, visualización con servidor web, 864
Protección por contraseña
Acceso a la CPU, 163
Bloque lógico, 167
Contraseña olvidada, 135
ENDIS_PW (limitar y habilitar legitimación de la contraseña), 300
Enlazar a una CPU, Memory Card o contraseña, 168
Protección anticopia, 168
Tarjeta de transferencia vacía, 135
Proteger datos de configuración confidenciales del PLC, 130
Protocolo
Comunicación, 940
Freeport, 940
ISO on TCP, 613
Modbus, 940
PROFINET RT, 613
TCP, 613
UDP, 613
USS, 940
Protocolo Freeport, 940
Protocolo ISO on TCP, 613
Protocolo UDP, 613
Protocolos Ethernet, 613
Conexiones multinodo, 799
Protocolos, comunicación, 614
Proyecto
Cargar, 199
comparar y sincronizar, 1213
Contraseña olvidada, 135
Enlazar a una CPU, Memory Card o contraseña, 168
Protección de acceso con contraseñas, 163
Proteger un bloque lógico, 167
Tarjeta de programa, 127

Tarjeta de transferencia, 123
Tarjeta de transferencia vacía, 135
Proyectos de máquina estándar (control de configuración), 143
PRVREC (proveer registro), 394
PTO (tren de impulsos)
Configurar canales de impulsos, 473
CTRL_PTO (tren de impulsos), 468
CTRL_PWM (modulación del ancho de impulso), 466
Funcionamiento, 471
No se puede forzar permanentemente, 1223
Puerto en anillo, 772
Puerto PROFINET
Autonegotiation, 606
Punteros
Tipo de datos Variant, 116
PUT (escribir datos en una CPU remota), 795
Configurar la conexión, 598
PWM (modulación del ancho de impulso)
Configurar canales de impulsos, 473
Direcciones de periferia, 476
Duración de impulso, 474
Funcionamiento, 471
Modificar el tiempo de ciclo, 476
Modificar la duración de impulso, 476
Tiempo de ciclo, 474
PWM (Modulación del ancho de impulso)
CTRL_PTO (tren de impulsos), 468
CTRL_PWM (modulación del ancho de impulso), 466
No se puede forzar permanentemente, 1223

Q

QRY_CINT (consultar parámetros de alarma cíclica), 405
QRY_DINT (consultar estado de una alarma de retardo), 410
QRY_TINT (consultar estado de alarma horaria), 409

R

R (desactivar salida), 212
R_TRIG (activar variable con flanco de señal ascendente), 216
RALRM (recibir alarma), 377, 385
RCV_CFG (configurar parámetros de recepción serie dinámicamente), antigua, 1115
RCV_PTP (habilitar recepción de mensajes), antigua, 1122
RCV_RST (borrar búfer de recepción), antigua, 1124
RCVREC (recibir registro), 392
RD_ADDR (determinar las direcciones ES a partir de la ID de hardware), 529
RD_LOC_T (leer hora local), 328
RD_SINFO (leer información de arranque del OB actual), 417
RD_SYS_T (leer la hora), 328
RDREC (leer registro), 370, 385
RE_TRIGR (reiniciar tiempo de vigilancia del ciclo), 303
READ_BIG (Leer datos en formato Big-Endian), 259
READ_DBL (leer de un bloque de datos de la memoria de carga), 518
READ_LITTLE (Leer datos en formato Little-Endian), 259
Realizar una copia de seguridad de una CPU, 1240
Receive_Config (configuración de recepción), 966
Receive_P2P (recepción punto a punto), 976
Receive_Reset (inicializar receptor), 978
Receta
Estructura de DB, 479
Programa de ejemplo, 487
RecipeExport (exportar receta), 483
RecipeImport (importar receta), 485
Vista general, 478
Recursos de información, 4
Redundancia
Clientes de redundancia, 765
Dominios de redundancia, 767
Redundancia de medios
Configurar, 769
Funciones en la topología en anillo, 765
Referencia cruzada para mostrar la utilización, 206
Referenciar tipos de enumeración, páginas web definidas por el usuario, 901
Referencias
Bloques de conectores, 1439
Cables de ampliación, 1439
Conectores y conexiones en borne, 1438
CPU, 1435
CPU 1214FC, CPU 1215FC, 1438
Escuadra final, 1439
Fuente de alimentación PM 1207, 1438
Interfaces de comunicación (CM, CB y CP), 1437
Memory Cards, 1439
Módulos de señales, 1435
Módulos de señales FS, 1438
Paneles HMI Basic, 1439
Signal Boards, Battery Boards, 1436
Simuladores, 1439
Software de programación, 1446

Software de visualización, 1446
STEP 7, 1446
Switch Ethernet CSM 1277, 1438
WinCC, 1446
Refrigeración, 44
Registro
Borrar con DataLogDelete, 501
DataLogClose (cerrar Data Log), 500
DataLogNewFile (Data Log en archivo nuevo), 503
DataLogOpen (abrir Data Log), 495
DataLogWrite (escribir Data Log), 497
Estructura del registro, 490
Límite de tamaño y tamaño calculado, 506
Programa de ejemplo, 509
Sinopsis de los registros, 489
Vaciar con DataLogClear, 499
Ver registros de datos, 505
Registro de datos (Data log)
DataLogCreate (crear registro), 491
Reloj
RD_LOC_T (leer hora local), 328
RD_SYS_T (leer la hora), 328
Reloj en tiempo real, 98
WR_LOC_T (escribir hora local), 328
WR_SYS_T (ajustar la hora), 328
Reloj del sistema
RD_SYS_T (leer la hora), 328
WR_LOC_T (escribir hora local), 328
WR_SYS_T (ajustar la hora), 328
REPEAT, SCL, 314
REPLACE (reemplazar caracteres de una cadena), 354
Requisitos del ordenador, 37
Requisitos del sistema, 37
Requisitos, instalación, 37
RESET_BF (desactivar mapa de bits), 213
Restablecer los ajustes de fábrica, 1206
Restablecer los valores de un DB, 1217
Restablecimiento de los ajustes de fábrica, 1206
Restaurar una copia de seguridad, 1243
Restricciones de cookies en páginas web estándar, 934
RET (retroceder), 299
Retardo al conectar (TON), 217
Retardo al desconectar (TOF), 217
Retardo RTS ON, OFF, 947
RETURN, SCL, 316
RIGHT (leer los caracteres derechos de una cadena), 351
RileReadC, 532
ROL (rotar a la izquierda) y ROR (rotar a la derecha), 323
ROUND (redondear número), 288
Router IP, 603
Routing S7, 772
RS (flipflop de desactivación/activación), 213
RT (inicializar temporizador), 217
RTS (control de flujo por hardware, PtP), 945
RUNTIME (Medir tiempo de ejecución), 307

S

S (activar salida), 212
S_CONV (convertir cadena de caracteres), 335
S_MOVE (desplazar cadena de caracteres), 335
Salidas de impulsos, 471
SCALE_X (escalar), 290
SCL (Structured Control Language)
Comparar valores, 231
Condiciones, 191
Control del programa, 309
Direccionamiento, 191
Editor de programas, 190
EN y ENO (flujo de corriente), 197
Expresiones, 191
Instrucciones de control, 191, 309
Instrucciones de conversión, 285
Llamar bloques, 177
Llamar un FB o una FC, 191
Lógica con bits, 209
Operadores, 191
Prioridad de operadores, 191
Sección Var, 190
Temporizadores, 217
Vista general, 189
Secuencia de caracteres
Fin del mensaje, 953
Inicio del mensaje, 949
Secuencia de movimientos (MC_CommandTable), 576
Segmento (de un tipo de datos marcado), 116
Seguridad
Contraseña olvidada, 135
Enlazar a una CPU, Memory Card o contraseña, 168
Protección anticopia, 168
Protección de acceso de la CPU, 163
Protección de know-how de un bloque lógico, 167
SEL (seleccionar), 319
SEND_CFG (configurar los parámetros de transmisión en serie dinámicamente), antigua, 1114
Send_Config (configuración de transmisión), 964
Send_P2P (enviar datos punto a punto), 973

SEND_P2P (enviar datos punto a punto)
Parámetros LENGH y BUFFER, 975
SEND_PTP (transmitir datos del búfer de transmisión), antigua, 1120
Serializar, 250
Servicio y asistencia, 3
Servidor OPC UA
Activar, 810
Ajustes de configuración, 813
Autenticación de usuario, 818
Clientes de confianza, 818
Definir interfaz de servidor, 820
Directivas de seguridad soportadas, 816
Especificaciones Companion, 823
Interfaces, 823
Nodos del PLC, 821
Sinopsis de seguridad, 816
Tipos de datos soportados, 820
Tipos de nodos soportados, 821
Servidor web
Acceso con dispositivos móviles, 852
Acceso mediante el módulo CP, 853
Aspecto en dispositivo móvil, 858
Conexiones HTTP máximas, 933
Configuración de usuario, 848
Convenciones para comillas, 905
Frecuencia de actualización, 846
Habilitar, 846
Limitaciones, 932
Páginas web definidas por el usuario, 891
Páginas web estándar, 851
Soporte de navegadores, 845
SET_BF (activar mapa de bits), 213
SET_CINT (ajustar parámetros de alarma cíclica), 403
Set_Features (activar funciones avanzadas), 983
SET_TIMEZONE (ajustar zona horaria), 331
SET_TINTL (ajustar alarma horaria), 407
SETIO, 374
SETIO_PART, 376
SGN_GET (consultar señales RS232), 979
SGN_GET (consultar señales RS-232), antigua, 1125
SGN_SET (activar señales RS-232), antigua, 1126
Shared Device
Concepto, 745
Configuración, 748
Shared I-device, configuración, 753
SHL (desplazar a la izquierda) y SHR (desplazar a la derecha), 322
siemens_automation_language cookie, 924
Signal Boards (SB)
Agregar módulos, 142
Configuración de parámetros, 169
Consumo de corriente, 1431
Desmontaje, 52
Instalación, 52
Representación de entradas (intensidad), 1352, 1396
Representación de entradas (tensión), 1352, 1395
Representación de salidas analógicas (intensidad), 1353, 1397
Representación de salidas analógicas (tensión), 1353, 1396
SB 1221 DI 4 x 24 V DC, 200 kHz, 1380
SB 1221 DI 4 x 5 V DC, 200 kHz, 1380
SB 1222 DQ 4 x 24 V DC, 200 kHz, 1382
SB 1222 DQ 4 x 5 V DC, 200 kHz, 1382
SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1388
SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1385
SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1385
SB 1231 AI 1 x 12 bits, 1391
SB 1231 AI 1 x 16 bits de termopar, 1398
SB 1231 AI 1 x 16 bits RTD, 1403
SB 1232 AQ 1 x 12 bits, 1393
Vista general, 31
Signal Boards analógicas
SB 1231, 1391
SB 1231 de termopar, 1398
SB 1231 RTD, 1403
SB 1232, 1393
Signal Boards digitales
SB 1221, 1380
SB 1222, 1382
SB 1223, 1385, 1388
Signal_Set (activar señales RS-232), 980
SIMATIC Memory Card
Actualización de firmware, 132
Leer información, 462
Protección de datos de configuración confidenciales del PLC, 130
Tarjeta de programa, 127
Vista general, 119
SIMATIC Memory Card de CPU
Tarjeta de programa, 127
Simuladores, 1424
Simuladores de entradas, 1424
SIN (calcular seno), 242
Sin rearranque, 71
Sincronización
Propiedad de sincronización horaria (PROFINET), 611

Sincronización de los valores iniciales del bloque de datos, 202
Sincronización horaria, 173
Sistema IO, intercambio de datos, 740
SM 1231 RTD
tablas de selección, 1405
Tablas de selección, 1363
SM y SB
Configuración de dispositivos, 137
Tabla de comparación, 31
SMS, 1184
Solución de problemas
búfer de diagnóstico, 1211
Indicadores LED, 1197
Soporte técnico de Siemens, 3
SQR (calcular cuadrado), 242
SQRT (calcular raíz cuadrada), 242
SR (flipflop de activación/desactivación), 213
SRT_DINT (iniciar alarma de retardo), 410
STEP 7
Agregar dispositivo, 139
Agregar módulos, 142
Agregar un dispositivo PROFINET IO, 733
AS-i, 787
Asignar una dirección IP a una CPU online, 602
Bloque de datos (DB), 67
Bloque de datos (DB) de instancia, 181
Bloque de función (FB), 67, 181
Carga de comunicación, 92
Cargar, 199
Clase de prioridad (OB), 75
comparar y sincronizar, 1213
Compatibilidad de versiones, 40
Conexión de red, 596
Configuración de dispositivos, 137
Configuración del HSC, 555
Configurar la CPU, 153
Configurar los módulos, 169
Copiar bloques desde una CPU online, 204
Estados operativos, 71
Forzar permanentemente, 1222
Función (FC), 180
Funcionamiento, 1218
Llamadas de bloque, 67
Llamar bloques lógicos en el programa de usuario, 177
Números válidos de FC, FB y DB, 67
observar, 1215, 1216
Operación de forzado permanente, 1222
Procesamiento del arranque, 73
PROFIBUS, 782
Programas lineales y estructurados, 176
Propiedad de sincronización horaria (PROFINET), 611
Protección por contraseña, 167
Puerto AS-i, 786
Puerto Ethernet, 603
Puerto PROFINET, 603
RTM (contador de horas de funcionamiento), 332
Tarjeta de programa, 119
Tiempo de ciclo, 91, 92
Tipos de bloques lógicos, 67
Valor inicial de un FB, 181
Vista del portal y Vista del proyecto, 39
STP (finalizar programa), 304
Strg_TO_Chars (convertir cadena de caracteres en Array of CHAR), 344
STRG_VAL (convertir cadena de caracteres en valor numérico), 335
String
S_MOVE (desplazar cadena de caracteres), 335
Sinopsis del tipo de datos String, 334
Tipo de datos STRING, 112
Vista general de las operaciones con cadenas, 348
SUB (restar), 237
Superposición de variable AT, 117
Support, 3
Sustitución de una CPU V3.0 por una CPU V4.x, 1448
SWAP (cambiar disposición), 254
SWITCH (distribuidor de saltos), 297

T

T_ADD (sumar tiempos), 326
T_COMBINE (combinar tiempos), 327
T_CONFIG (configurar interfaz), 712
T_CONV (convertir y extraer tiempos), 325
T_DIAG, 678
T_DIFF (diferencia de tiempo), 327
T_RESET, 676
T_SUB (restar tiempos), 326
Tabla comparativa
Modelos de CPU, 28
Tabla de comparación
Dispositivos HMI, 32
Módulos, 31
Tabla de forzado permanente
direccionar entradas de periferia, 1222
Forzar permanentemente, 1222
Operación de forzado permanente, 1222
Tabla de observación
Desbloquear las salidas en modo STOP, 1221
Disparar valores, 1220

Forzado permanente, 205
Funcionamiento, 1218
observar, 1215
Tableta, acceso al servidor web, 852
TAN (calcular tangente), 242
Tarjeta de programa
Configurar los parámetros de arranque, 122
Crear, 127
Insertar en la CPU, 120
Vista general, 119
Tarjeta de transferencia, 123
Configurar los parámetros de arranque, 122
Contraseña olvidada, 135
Insertar en la CPU, 120
Tarjeta de transferencia vacía para una contraseña olvidada, 135
Vista general, 119
Tarjetas de transferencia (de programas), 1424
Task Cards
Columnas y encabezados, 631, 644, 651, 667, 997, 1016, 1086, 1128, 1141, 1158
TCON, 652
Configuración, 597
ID de conexión, 616
Parámetros de la conexión, 618
TCON, TDISCON, TSEND y TRCV
Versiones, 651, 667
TCON_Param, 618
TCP
Configuración de la conexión, 597
ID de conexión, 616
Modo ad hoc, 615
Parámetros, 618
Protocolo, 613
TDISCON, 652
Technical support, 3
Telecontrol, 1181
TeleControl
Procesadores de comunicaciones, 1177
Teléfono inteligente, acceso al servidor web, 852
TeleService vía GPRS, 1182
Temporizadores
Cantidad, 30, 1258, 1270, 1282, 1295, 1309
Funcionamiento, 220
RT (inicializar temporizador), 217
Tamaño, 30, 1258, 1270, 1282, 1295, 1309
TOF (retardo al desconectar), 217
TON (retardo al conectar), 217
TONR (acumulador de tiempo), 217
TP (impulso), 217
Tensiones nominales, 1251
Termopar
Compensación de unión fría, 1357, 1400
funcionamiento básico, 1357, 1400
SB 1231 AI 1 x 16 bits, 1398
Tabla de selección de filtros de la SB 1231, 1401
Tabla de selección de filtros de termopar para el SM 1231, 1357
Tabla de selección de filtros de termopar para la SB 1231, 1400
Tabla de selección de termopares del SM 1231, 1357
TIA Portal, Vista del portal y Vista del proyecto, 39
Tiempo
T_ADD (sumar tiempos), 326
T_SUB (restar tiempos), 326
Tiempo de ciclo
Configuración, 92
vigilar, 1211
Vista general, 91
Tiempo de espera, 944
Tiempo de filtro, 155
Tiempo de filtro de entrada, 155
Tiempo de filtro de entrada digital, 155
Tiempo de supervisión de respuesta (instrucción RE_TRIGR), 303
Tiempo excedido entre caracteres, 953
Tiempos de ejecución, 1257, 1269, 1281, 1294, 1308
Time
DTL (tipo de datos Data and Time Long), 111
Tipo de datos Time, 110
TOD (tipo de datos Time of Day), 110
TimeTransformationRule para horario de verano, 330
Tipos de datos, 106
Bool, byte, palabra o palabra doble, 107
Caracteres y cadenas, 112
Editor de tipo de datos PLC, 115
Matrices, 114
Real, LReal (real en coma flotante), 109
Struc, 115
Time, Date, TOD (hora del día), DTL (fecha y hora largo), 110
USInt, SInt, UInt, Int, UDInt, Dint (entero), 108
Variant (puntero), 116
Tipos de enumeración en páginas web definidas por el usuario, 901
TM_MAIL (transmitir e-mail), 1189
TMAIL_C, 682
Topología
Anillo, 765
Topología en anillo, 765
Transición de RUN a STOP, 98

Transmisión de datos, iniciar, 973, 1120
TRCV, 652
  ID de conexión, 616
TRCV (recibir datos vía Ethernet (TCP))
  Configuración de parámetros, 732
  Modo ad hoc, 615
TRCV_C
  Modo ad hoc, 615
TRCV_C (recibir datos vía Ethernet (TCP)), 631
  Configuración, 597
  ID de conexión, 616
  Parámetros de la conexión, 618
TRCV_C Legacy (recibir datos vía Ethernet (TCP)), 644
TRUNC (truncar a entero), 288
TS Adapter, 31
TSAP (Transport Service Access Points), 599
  Configurar los parámetros generales, 730, 800
  Definición, 614
  Instrucciones para la asignación a dispositivos, 613
  TSAPs y números de puerto restringidos, 724
TSAPs y números de puerto restringidos, 724
TSEND, 652
  ID de conexión, 616
TSEND_C (enviar datos vía Ethernet (TCP)), 631
  Configuración, 597
  Configuración de instrucciones, 731
  ID de conexión, 616
  Parámetros de la conexión, 618
TSEND_C Legacy (enviar datos vía Ethernet (TCP)), 644
TSEND_C y TRCV_C
  Versiones, 631
  Versiones Legacy, 644
TURCV (recibir datos vía Ethernet (UDP)), 707
  Configuración, 597
  Parámetros de la conexión, 618
TUSEND (enviar datos vía Ethernet (UDP)), 707
  Configuración, 597
  Parámetros, 618

U

UDP
  Configuración de la conexión, 597
  Parámetros, 618
UFILL_BLK (rellenar área sin interrupciones), 253
UMOVE_BLK (copiar área sin interrupciones), 244
UPPER_BOUND (leer límite superior de ARRAY), 256

V

VAL_STRG (convertir valor numérico en cadena de caracteres), 335
Valores booleanos o de bit, 100
Valores de retorno
  Instrucciones de la comunicación abierta, 723
  Instrucciones PtP, 959
Valores de retorno de instrucciones PtP, 959
Valores de retorno de las instrucciones de la comunicación abierta, 723
Valores de retorno en el tiempo de ejecución de la recepción, 976, 1122
Variable
  Operación de forzado permanente, 1222
  Segmento, 116
  Superposición, 117
  Vigilancia de estado o valor, 1215
Variables, observar y forzar desde el servidor web, 875
VARIANT_TO_DB_ANY (convertir VARIANT en DB_ANY), 293
VariantGet (Leer valor de una variable VARIANT), 261
VariantPut (Escribir valor en una variable VARIANT), 262
Velocidad de ejecución de las instrucciones, 1257, 1269, 1281, 1294, 1308
Velocidad de transferencia, 944
Ventilación, 44
Versiones de instrucciones, 631, 644, 651, 667, 997, 1016, 1086, 1128, 1141, 1158
Vida útil de la SIMATIC Memory Card, 129
Vida útil de los relés, 1252
Vigilar
  carga de la memoria, 1211
  Tiempo de ciclo, 1211
Vista del portal, 39
Vista del proyecto, 39, 40
Vista topológica, 39
Visualización, dispositivos HMI, 32
Visualizar las direcciones MAC e IP, 608

W

WChar (tipo de datos Word Character), 112
WHILE, SCL, 313
WR_LOC_T (escribir hora local), 328
WR_SYS_T (ajustar la hora), 328
WRIT_DBL (escribir en un bloque de datos de la memoria de carga), 518

WRITE_BIG (Escribir datos en formato Big-Endian), 259
WRITE_LITTLE (Escribir datos en formato Little-Endian), 259
WRREC (escribir registro), 370, 385
WString (tipo de datos Word String), 112
WWW (sincronización de páginas web definidas por el usuario), 909

X

XON/XOFF, 946
XOR (operación lógica O-exclusiva), 317

Z

Zona de disipación, 44, 47