

SIMATIC

S7 S7-1200 Automatisierungssystem

Systemhandbuch

Vorwort	
Produktübersicht	1
Neue Funktionen	2
STEP 7 Programmiersoftware	3
Einbau	4
PLC-Grundlagen	5
Gerätekonfiguration	6
Programmierkonzepte	7
Anweisungen	8
Erweiterte Anweisungen	9
Technologieanweisungen	10
Kommunikation	11
Webserver	12
Kommunikationsprozessor und Modbus-TCP	13
TeleService- Kommunikation (SMTP-E- Mail)	14
Online- und Diagnose-Tools	15
Technische Daten	A
Berechnung der Leistungsbilanz	B
Bestellinformationen	C
Geräteaustausch und Ersatzteilkompatibilität	D

Rechtliche Hinweise

Warnhinweiskonzept


Dieses Handbuch enthält Hinweise, die Sie zu Ihrer persönlichen Sicherheit sowie zur Vermeidung von Sachschäden beachten müssen. Die Hinweise zu Ihrer persönlichen Sicherheit sind durch ein Warndreieck hervorgehoben, Hinweise zu alleinigen Sachschäden stehen ohne Warndreieck. Je nach Gefährdungsstufe werden die Warnhinweise in abnehmender Reihenfolge wie folgt dargestellt.

 GEFAHR

bedeutet, dass Tod oder schwere Körperverletzung eintreten wird , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 WARNUNG
--

bedeutet, dass Tod oder schwere Körperverletzung eintreten kann , wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

 VORSICHT

bedeutet, dass eine leichte Körperverletzung eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

ACHTUNG

bedeutet, dass Sachschaden eintreten kann, wenn die entsprechenden Vorsichtsmaßnahmen nicht getroffen werden.

Beim Auftreten mehrerer Gefährdungsstufen wird immer der Warnhinweis zur jeweils höchsten Stufe verwendet. Wenn in einem Warnhinweis mit dem Warndreieck vor Personenschäden gewarnt wird, dann kann im selben Warnhinweis zusätzlich eine Warnung vor Sachschäden angefügt sein.

Qualifiziertes Personal

Das zu dieser Dokumentation zugehörige Produkt/System darf nur von für die jeweilige Aufgabenstellung **qualifiziertem Personal** gehandhabt werden unter Beachtung der für die jeweilige Aufgabenstellung zugehörigen Dokumentation, insbesondere der darin enthaltenen Sicherheits- und Warnhinweise. Qualifiziertes Personal ist auf Grund seiner Ausbildung und Erfahrung befähigt, im Umgang mit diesen Produkten/Systemen Risiken zu erkennen und mögliche Gefährdungen zu vermeiden.

Bestimmungsgemäßer Gebrauch von Siemens-Produkten

Beachten Sie Folgendes:

 WARNUNG
--

Siemens-Produkte dürfen nur für die im Katalog und in der zugehörigen technischen Dokumentation vorgesehenen Einsatzfälle verwendet werden. Falls Fremdprodukte und -komponenten zum Einsatz kommen, müssen diese von Siemens empfohlen bzw. zugelassen sein. Der einwandfreie und sichere Betrieb der Produkte setzt sachgemäßen Transport, sachgemäße Lagerung, Aufstellung, Montage, Installation, Inbetriebnahme, Bedienung und Instandhaltung voraus. Die zulässigen Umgebungsbedingungen müssen eingehalten werden. Hinweise in den zugehörigen Dokumentationen müssen beachtet werden.

Marken

Alle mit dem Schutzrechtsvermerk ® gekennzeichneten Bezeichnungen sind eingetragene Marken der Siemens AG. Die übrigen Bezeichnungen in dieser Schrift können Marken sein, deren Benutzung durch Dritte für deren Zwecke die Rechte der Inhaber verletzen kann.

Haftungsausschluss

Wir haben den Inhalt der Druckschrift auf Übereinstimmung mit der beschriebenen Hard- und Software geprüft. Dennoch können Abweichungen nicht ausgeschlossen werden, so dass wir für die vollständige Übereinstimmung keine Gewähr übernehmen. Die Angaben in dieser Druckschrift werden regelmäßig überprüft, notwendige Korrekturen sind in den nachfolgenden Auflagen enthalten.

Vorwort

Zweck des Handbuchs

Die Familie S7-1200 umfasst verschiedene speicherprogrammierbare Steuerungen (SPS), mit denen Sie eine breite Palette von Automatisierungsaufgaben lösen können. Durch das kompakte Design, die Möglichkeit der Erweiterung, den günstigen Preis und einen leistungsstarken Befehlssatz eignet sich die S71200 hervorragend für eine Vielzahl von Steuerungsanwendungen. Die Ausführungen der S7-1200 und die vielfältigen STEP 7-Programmiermöglichkeiten (Seite 37) unter Windows bieten Ihnen eine extrem hohe Flexibilität beim Umsetzen Ihrer Automatisierungslösungen.

Dieses Handbuch bietet Informationen zum Installieren und Programmieren von S7-1200 Steuerungen und wendet sich an Ingenieure, Programmierer und Wartungspersonal mit allgemeinen Kenntnissen über Automatisierungssysteme.

Erforderliche Grundkenntnisse

Damit Sie mit diesem Handbuch arbeiten können, benötigen Sie allgemeine Kenntnisse im Bereich der Automatisierung und der speicherprogrammierbaren Steuerungen.

Umfang des Handbuchs

Dieses Handbuch beschreibt die folgenden Produkte:

- STEP 7 Basic und Professional (Seite 37)
- S7-1200 CPU Firmware Release V4.5

Eine vollständige Liste der S71200 Produkte finden Sie in den technischen Daten (Seite 1253).

Zertifizierung, CE-Kennzeichen, C-Tick und andere Zulassungen

Ausführliche Informationen finden Sie in den technischen Daten (Seite 1253).

Service und Support

Zusätzlich zu unserem Dokumentations-Angebot bietet Siemens Ihnen im Internet technisches Know-how auf der Kundensupport-Website (<https://support.industry.siemens.com/cs/ww/de>) an.

Falls Sie technische Fragen haben, eine Schulung benötigen oder S7-Produkte bestellen wollen, wenden Sie sich bitte an Ihre Siemens-Vertretung. Das technisch geschulte Vertriebspersonal verfügt über sehr spezifische Kenntnisse zu Einsatzmöglichkeiten und Prozessen sowie zu den verschiedenen Siemens-Produkten und kann Ihnen deshalb am schnellsten und besten weiterhelfen, wenn Probleme auftreten.

Dokumentation und Information

S7-1200 und STEP 7 bieten eine Vielzahl von Dokumentationen und anderen Quellen mit technischen Informationen.

- Das Systemhandbuch S7-1200 Automatisierungssystem bietet spezielle Informationen zu Funktionsweise, Programmierung und technischen Daten der gesamten S7-1200-Produktfamilie.
Das Systemhandbuch steht elektronisch (im PDF-Format) zur Verfügung. Sie können dieses und andere elektronische Handbücher auf der Siemens-Website Industry Online-Support (<https://support.industry.siemens.com/cs/ww/de>) herunterladen oder anzeigen. Das Systemhandbuch ist außerdem auf der Dokumenten-CD verfügbar, die zum Lieferumfang jeder S7-1200 CPU gehört.
- Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Konzeptinformationen und spezielle Hinweise zur Funktionsweise und Funktionalität des Programmierpakets sowie zur grundlegenden Funktionsweise der SIMATIC CPUs.
- Die Siemens-Website Industry Online-Support (<https://support.industry.siemens.com/cs/ww/de>) bietet Zugriff auf die elektronischen Versionen (PDF) des SIMATIC-Dokumentationssatzes, der das Systemhandbuch und das Informationssystem von STEP 7 umfasst. Die vorhandenen Dokumente finden Sie unter dem Link Produkt-Support. Über diesen Zugriff auf die Online-Dokumentation können Sie auch Themen aus verschiedenen Dokumenten per Drag & Drop selbst anordnen und so eigene benutzerspezifische Handbücher anlegen. Auch Aktualisierungen früher veröffentlichter Systemhandbücher sind unter Siemens Industry Online-Support erhältlich. Die Online-Dokumentation rufen Sie über den Link "mySupport" auf der linken Seite auf dieser Website auf. Klicken Sie dann auf die Option "Dokumentation". Um die Funktion der Dokumentation unter mySupport nutzen zu können, müssen Sie sich als registrierter Benutzer anmelden.
- Die Website Siemens Industry Online-Support bietet außerdem FAQs und andere hilfreiche Dokumente für S7-1200 und STEP 7.
- Im technischen Forum unter Service & Support (<https://support.industry.siemens.com/tf/ww/de/?Language=de&onlyInternet=False>) können Sie außerdem Produktdiskussionen verfolgen oder sich daran beteiligen. Folgende Foren bieten Ihnen die Möglichkeit, mit verschiedenen Produktexperten in Kontakt zu treten.
 - Forum für S7-1200 (<https://support.industry.siemens.com/tf/ww/de/threads/236?title=simatic-s7-1200&skip=0&take=10&orderBy=LastPostDate+desc>)
 - Forum für STEP 7 Basic (<https://support.industry.siemens.com/tf/ww/de/threads/241?title=step-7-tia-portal&skip=0&take=10&orderBy=LastPostDate+desc>)

Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Systemen, Maschinen und Netzwerken unterstützen.

Um Anlagen, Systeme, Maschinen und Netzwerke gegen Cyber-Bedrohungen zu sichern, ist es erforderlich, ein ganzheitliches Industrial Security-Konzept zu implementieren (und kontinuierlich aufrechtzuerhalten), das dem aktuellen Stand der Technik entspricht. Die Produkte und Lösungen von Siemens formen einen Bestandteil eines solchen Konzepts.

Die Kunden sind dafür verantwortlich, unbefugten Zugriff auf ihre Anlagen, Systeme, Maschinen und Netzwerke zu verhindern. Diese Systeme, Maschinen und Komponenten sollten

nur mit dem Unternehmensnetzwerk oder dem Internet verbunden werden, wenn und soweit dies notwendig ist und nur wenn entsprechende Schutzmaßnahmen (z. B. Firewalls und/oder Netzwerksegmentierung) ergriffen wurden.

Weiterführende Informationen zu möglichen Schutzmaßnahmen im Bereich Industrial Security finden Sie unter (<https://www.siemens.com/industrialsecurity>).

Die Produkte und Lösungen von Siemens werden ständig weiterentwickelt, um sie noch sicherer zu machen. Siemens empfiehlt ausdrücklich, Produkt-Updates anzuwenden, sobald sie zur Verfügung stehen und immer nur die aktuellen Produktversionen zu verwenden. Die Verwendung veralteter oder nicht mehr unterstützter Versionen kann das Risiko von Cyber-Bedrohungen erhöhen.

Um stets über Produkt-Updates informiert zu sein, abonnieren Sie den Siemens Industrial Security RSS Feed unter (<https://www.siemens.com/industrialsecurity>).

Aufrechterhaltung der Betriebssicherheit Ihrer Anlage

Aufrechterhaltung der Betriebssicherheit Ihrer Anlage

Hinweis

Wichtiger Hinweis zur Aufrechterhaltung der Betriebssicherheit Ihrer Anlage

Anlagen mit sicherheitsrelevanten Funktionen unterliegen besonderen Anforderungen hinsichtlich der bedienerseitigen Betriebssicherheit. Selbst Lieferanten müssen während der Produktüberwachung besondere Maßnahmen einhalten. Deshalb informieren wir Sie in persönlichen Benachrichtigungen über Produktentwicklungen und Leistungsmerkmale, die aus einer Sicherheitsperspektive für den Betrieb von Systemen relevant sind (oder sein könnten).

Indem Sie die entsprechenden Benachrichtigungen abonnieren, stellen Sie sicher, dass Sie stets auf dem neuesten Stand sind und gegebenenfalls Änderungen an Ihrem System vornehmen können.

Melden Sie sich beim Industry Online-Support an. Gehen Sie zu den folgenden Links und klicken Sie an der Seite mit der rechten Maustaste auf "E-Mail bei Aktualisierung":

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

Inhaltsverzeichnis

	Vorwort	3
1	Produktübersicht	27
1.1	Einführung in den S7-1200 PLC.....	27
1.2	Erweiterungsfähigkeit der CPU	31
1.3	Grundlegende HMI-Panels.....	32
2	Neue Funktionen	35
3	STEP 7 Programmiersoftware	37
3.1	Systemvoraussetzungen.....	37
3.2	Einfaches Arbeiten mit unterschiedlichen Ansichten	39
3.3	Kompatibilität zwischen STEP 7 und der S7-1200.....	40
4	Einbau	43
4.1	Richtlinien für den Einbau von S71200 Geräten	43
4.2	Leistungsbilanz	45
4.3	Vorgehensweisen zum Einbau und Ausbau.....	46
4.3.1	Befestigungsmaße für S7-1200 Geräte	46
4.3.2	Einbau und Ausbau der CPU	49
4.3.3	Ein- und Ausbau eines SBs, CBs oder BBs	52
4.3.4	Einbau und Ausbau eines SMs	54
4.3.5	Einbau und Ausbau eines CMs oder CPs.....	56
4.3.6	Ausbau und Einbau des S7-1200 Klemmenblocks	57
4.3.7	Einbau und Ausbau des Erweiterungskabels	58
4.4	Verdrahtungsrichtlinien	60
5	PLC-Grundlagen	67
5.1	Ausführung des Anwenderprogramms	67
5.1.1	Betriebszustände der CPU.....	71
5.1.2	Verarbeitung des Zyklus im Betriebszustand RUN.....	74
5.1.3	Organisationsbausteine (OBs)	75
5.1.3.1	Programmzyklus-OB.....	75
5.1.3.2	Anlauf-OB	76
5.1.3.3	Verzögerungsalarm-OB	76
5.1.3.4	Weckalarm-OB	77
5.1.3.5	Prozessalarm-OB	78
5.1.3.6	Zeitfehler-OB	79
5.1.3.7	Diagnosefehler-OB	80
5.1.3.8	OB "Ziehen oder Stecken von Modulen"	82
5.1.3.9	Baugruppenträger- oder Stationsfehler-OB ("Rack or station failure OB").....	82
5.1.3.10	Uhrzeit-OB.....	83
5.1.3.11	Zustands-OB	84

5.1.3.12	Aktualisierungs-OB	84
5.1.3.13	Profil-OB	84
5.1.3.14	OB MC-Servo- und MC-Interpolator.....	85
5.1.3.15	MC-PreServo	85
5.1.3.16	MC-PostServo	86
5.1.3.17	Prioritäten und Warteschlange für die Ausführung von Ereignissen	86
5.1.4	Überwachen und Konfigurieren der Zykluszeit	90
5.1.5	CPU-Speicher	92
5.1.5.1	System- und Taktmerker.....	94
5.1.6	Diagnosepuffer	96
5.1.7	Echtzeituhr	97
5.1.8	Konfigurieren der Ausgänge für den Wechsel von RUN in STOP	97
5.2	Datenspeicher, Speicherbereiche, E/A und Adressierung	98
5.2.1	Zugriff auf die Daten der S7-1200.....	98
5.3	Verarbeitung von Analogwerten.....	104
5.4	Datentypen.....	105
5.4.1	Datentypen Bool, Byte, Word und DWord	107
5.4.2	Ganzzahlige Datentypen	108
5.4.3	Gleitpunktzahl/Realzahl-Datentypen	108
5.4.4	Uhrzeit- und Datums-Datentypen	109
5.4.5	Zeichen- und Zeichenfolge-Datentypen	110
5.4.6	Datentyp ARRAY	112
5.4.7	Datentyp Struktur	113
5.4.8	PLC-Datentyp	114
5.4.9	Pointer-Datentyp "Variant".....	114
5.4.10	Zugriff auf eine "Slice" eines Variablentyps	114
5.4.11	Zugriff auf eine Variable mit einer AT-Überlagerung	116
5.5	Memory Card verwenden.....	117
5.5.1	Memory Card in die CPU stecken	118
5.5.2	Anlaufparameter der CPU vor dem Kopieren des Projekts auf die Memory Card konfigurieren	121
5.5.3	Übertragungskarte.....	122
5.5.4	Programmkarte.....	124
5.5.5	Memory Card zum Schützen vertraulicher PLC-Konfigurationsdaten nutzen.....	129
5.5.6	Firmware-Update	131
5.6	Vorgehensweise bei verlorenem Passwort	134
6	Gerätekonfiguration	135
6.1	Einfügen einer CPU	136
6.2	Konfiguration aus einer angeschlossenen CPU laden.....	137
6.3	Module zur Konfiguration hinzufügen	139
6.4	Konfigurationssteuerung.....	141
6.4.1	Vorteile und Nutzung der Konfigurationssteuerung.....	141
6.4.2	Den zentralen Aufbau und optionale Module konfigurieren.....	141
6.4.3	Beispiel für die Konfigurationssteuerung.....	148
6.5	Ändern eines Geräts.....	152
6.6	Konfigurieren des CPU-Betriebs.....	153

6.6.1	Übersicht	153
6.6.2	Filterzeiten für Digitaleingänge einrichten	155
6.6.3	Impulsabgriff	156
6.7	Mehrsprachigen Support konfigurieren.....	158
6.8	Schutz & Security	160
6.8.1	PLC-Security-Einstellungen im Security-Assistenten festlegen.....	160
6.8.2	Schutz vertraulicher PLC-Konfigurationsdaten	161
6.8.3	Zugriffsschutz für die CPU.....	163
6.8.4	Verbindungsmechanismen konfigurieren.....	165
6.8.4.1	Zugriffsmechanismus für entfernte Partner festlegen.....	165
6.8.4.2	Sichere PG/PC-Kommunikation und HMI-Kommunikation aktivieren und Zertifikate erstellen	166
6.8.5	Externer Ladespeicher	167
6.8.6	Knowhow-Schutz.....	167
6.8.7	Kopierschutz.....	168
6.9	Modulparameter konfigurieren	169
6.10	CPU für die Kommunikation konfigurieren	171
6.11	Uhrzeitsynchronisation	172
7	Programmierkonzepte	175
7.1	Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät.....	175
7.2	Strukturieren Ihres Anwenderprogramms	176
7.3	Verwendung von Bausteinen zum Strukturieren Ihres Programms	178
7.3.1	Organisationsbaustein (OB)	178
7.3.2	Funktion (FC).....	180
7.3.3	Funktionsbaustein (FB)	181
7.3.4	Datenbaustein (DB).....	182
7.3.5	Anlegen wiederverwendbarer Codebausteine.....	184
7.3.6	Übergabe von Parametern an Bausteine	184
7.4	Datenkonsistenz	187
7.5	Programmiersprache.....	188
7.5.1	Kontaktplan (KOP)	189
7.5.2	Funktionsplan (FUP).....	190
7.5.3	SCL.....	190
7.5.3.1	SCL-Programmiereditor	190
7.5.3.2	SCL-Ausdrücke und -Operationen	192
7.5.3.3	Indexierte Adressierung mit den Anweisungen PEEK und POKE.....	195
7.5.4	EN und ENO in KOP, FUP und SCL	197
7.6	Laden der Programmelemente	199
7.7	Synchronisieren der Online-CPU und des Offline-Projekts	203
7.8	Laden von der Online-CPU.....	205
7.8.1	Vergleich der Online-CPU mit der Offline-CPU.....	205
7.9	Debugging und Testen des Programms.....	205
7.9.1	Daten in der CPU beobachten und steuern.....	205
7.9.2	Beobachtungstabellen und Forcetabellen	206
7.9.3	Querverweis zum Anzeigen der Verwendung	207

7.9.4	Aufrufstruktur zur Prüfung der Aufrufhierarchie	208
8	Anweisungen	209
8.1	Bitverknüpfungen	209
8.1.1	Bitverknüpfungsanweisungen	209
8.1.2	Setz- und Rücksetzoperationen	212
8.1.3	Operationen Steigende Flanke und Fallende Flanke	214
8.2	Funktionsweise der Zeiten	217
8.3	Funktionsweise der Zähler	225
8.4	Funktionsweise von Vergleichen	230
8.4.1	Vergleichsoperationen	230
8.4.2	IN_Range (Wert innerhalb Bereich) und OUT_Range (Wert außerhalb Bereich)	231
8.4.3	OK (Gültigkeit prüfen) und NOT_OK (Ungültigkeit prüfen)	232
8.4.4	Variant- und Array-Vergleichsoperationen	232
8.4.4.1	Gleich- und Ungleich-Vergleichsoperationen	232
8.4.4.2	Null-Vergleichsoperationen	233
8.4.4.3	IS_ARRAY (Auf ARRAY prüfen)	234
8.5	Arithmetische Funktionen	235
8.5.1	CALCULATE (Berechnen)	235
8.5.2	Anweisungen Addieren, Subtrahieren, Multiplizieren und Dividieren	236
8.5.3	MOD (Divisionsrest einer Division)	237
8.5.4	NEG (Zweierkomplement erstellen)	238
8.5.5	INC (Inkrementieren) und DEC (Dekrementieren)	239
8.5.6	ABS (Absolutwert bilden)	239
8.5.7	MIN (Minimum abrufen) und MAX (Maximum abrufen)	240
8.5.8	LIMIT (Grenzwert setzen)	241
8.5.9	Exponential-, Logarithmus- und Trigonometrieanweisungen	242
8.6	Anweisungen zum Übertragen von Daten	244
8.6.1	MOVE (Wert kopieren), MOVE_BLK (Bereich kopieren), UMOVE_BLK (Bereich ununterbrechbar kopieren) und MOVE_BLK_VARIANT (Bereich kopieren)	244
8.6.2	Deserialize	248
8.6.3	Serialize	250
8.6.4	FILL_BLK (Speicher mit Bitmuster belegen) und UFILL_BLK (Speicher ununterbrechbar mit Bitmuster belegen)	253
8.6.5	SWAP (Anordnung ändern)	255
8.6.6	LOWER_BOUND: (Untere ARRAY-Grenze auslesen)	255
8.6.7	UPPER_BOUND: (Obere ARRAY-Grenze auslesen)	257
8.6.8	Anweisungen Speicher lesen / in Speicher schreiben	258
8.6.8.1	PEEK und POKE (nur SCL)	258
8.6.8.2	Anweisungen Big- und Little-Endian-Format lesen und schreiben (SCL)	260
8.6.9	Variant-Anweisungen	262
8.6.9.1	VariantGet (VARIANT Variablenwert lesen)	262
8.6.9.2	VariantPut (VARIANT Variablenwert schreiben)	263
8.6.9.3	CountOfElements (Anzahl ARRAY-Elemente abfragen)	264
8.6.10	Anweisungen in älteren Systemen	265
8.6.10.1	Anweisungen FieldRead (Feld lesen) und FieldWrite (Feld schreiben)	265
8.6.11	SCATTER	267
8.6.12	SCATTER_BLK	271
8.6.13	GATHER	276
8.6.14	GATHER_BLK	280

8.7	Umwandlungsoperationen.....	285
8.7.1	CONV (Wert umwandeln).....	285
8.7.2	Umwandlungsanweisungen in SCL.....	286
8.7.3	ROUND (Zahl runden) und TRUNC (Ganzzahl erzeugen).....	289
8.7.4	CEIL und FLOOR (Aus Gleitpunktzahl nächsthöhere und nächstniedere Ganzzahl erzeugen).....	290
8.7.5	SCALE_X (Skalieren) und NORM_X (Normieren).....	291
8.7.6	Variant-Umwandlungsanweisungen	294
8.7.6.1	VARIANT_TO_DB_ANY (VARIANT in DB_ANY konvertieren)	294
8.7.6.2	DB_ANY_TO_VARIANT (DB_ANY in VARIANT konvertieren)	295
8.8	Programmsteuerungsoperationen	296
8.8.1	Anweisungen JMP (Springen bei VKE = 1), JMPN (Springen bei VKE = 0) und Label (Sprungmarke)	296
8.8.2	JMP_LIST (Sprungliste definieren)	297
8.8.3	SWITCH (Sprungverteilung)	298
8.8.4	RET (Rückgabewert)	300
8.8.5	ENDIS_PW (CPU-Passwort aktivieren/deaktivieren)	301
8.8.6	RE_TRIGR (Zyklusüberwachungszeit neu starten)	304
8.8.7	STP (Programm beenden)	305
8.8.8	Anweisungen GET_ERROR und GET_ERROR_ID (Fehler lokal abrufen und Fehler-ID lokal abrufen)	305
8.8.9	RUNTIME (Programmlaufzeit messen)	309
8.8.10	Programmsteuerungsanweisungen in SCL	310
8.8.10.1	Übersicht über die Programmsteuerungsanweisungen in SCL	310
8.8.10.2	IF-THEN-Anweisung	311
8.8.10.3	CASE-Anweisung	312
8.8.10.4	FOR-Anweisung	313
8.8.10.5	WHILE-DO-Anweisung.....	314
8.8.10.6	REPEAT-UNTIL-Anweisung.....	315
8.8.10.7	CONTINUE-Anweisung	316
8.8.10.8	EXIT-Anweisung.....	317
8.8.10.9	GOTO-Anweisung	317
8.8.10.10	RETURN-Anweisung	318
8.9	Wortverknüpfung	318
8.9.1	Verknüpfungsoperationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER).....	318
8.9.2	INV (Einerkomplement erstellen).....	319
8.9.3	Anweisungen DECO (Decodieren) and ENCO (Encodieren).....	320
8.9.4	Anweisungen SEL (Selektieren), MUX (Multiplexen) und DEMUX (Demultiplexen).....	321
8.10	Schieben und Rotieren	324
8.10.1	Anweisungen SHL (Rechts schieben) und SHL (Links schieben)	324
8.10.2	Anweisungen ROR (Rechts rotieren) und ROL (Links rotieren).....	325
9	Erweiterte Anweisungen	327
9.1	Datums-, Uhrzeit- und Uhrfunktionen	327
9.1.1	Datums- und Uhrzeitanweisungen	327
9.1.2	Uhrzeitfunktionen.....	330
9.1.3	Datenstruktur TimeTransformationRule	332
9.1.4	SET_TIMEZONE (Zeitzone setzen)	333
9.1.5	RTM (Betriebsstundenzähler).....	334
9.2	Zeichenketten- und Zeichenanweisungen.....	336

9.2.1	Datentyp String.....	336
9.2.2	S_MOV (Zeichenkette verschieben)	337
9.2.3	Anweisungen für die Zeichenkettenkonvertierung	337
9.2.3.1	Anweisungen S_CONV, STRG_VAL und VAL_STRG (In/von Zeichenkette und Zahlenwert umwandeln).....	337
9.2.3.2	Anweisungen Strg_TO_Chars und Chars_TO_Strg (In/aus Zeichenkette und Array aus CHAR umwandeln).....	346
9.2.3.3	ATH und HTA (In/aus ASCII-Zeichenkette und Hexadezimalzahl umwandeln)	348
9.2.4	Zeichenkettenanweisungen	350
9.2.4.1	MAX_LEN (Maximale Länge einer Zeichenkette)	351
9.2.4.2	LEN (Länge einer Zeichenkette ermitteln)	351
9.2.4.3	CONCAT (Zeichenketten verketten).....	352
9.2.4.4	Anweisungen LEFT, RIGHT und MID (Teilzeichenketten in einer Zeichenkette lesen).....	353
9.2.4.5	DELETE (Zeichen in einer Zeichenkette löschen)	354
9.2.4.6	INSERT (Zeichen in einer Zeichenkette einfügen)	355
9.2.4.7	REPLACE (Zeichen in einer Zeichenkette ersetzen)	356
9.2.4.8	FIND (Zeichen in einer Zeichenkette finden)	357
9.2.5	Informationen zur Laufzeit	358
9.2.5.1	GetSymbolName (Namen einer Variable am Eingangsparameter auslesen)	358
9.2.5.2	GetSymbolPath (Zusammengesetzten globalen Namen der Eingangsparameterversorgung abfragen).....	361
9.2.5.3	GetInstanceName (Namen der Baustein-Instanz auslesen)	364
9.2.5.4	GetInstancePath (Zusammengesetzten globalen Namen der Baustein-Instanz abfragen) ...	367
9.2.5.5	GetBlockName (Name des Bausteins auslesen)	369
9.3	Dezentrale E/A (PROFINET, PROFIBUS oder AS-i)	371
9.3.1	Anweisungen für die dezentrale E/A	371
9.3.2	RDREC und WRREC (Datensatz lesen/schreiben)	373
9.3.3	GETIO (Prozessabbild lesen)	376
9.3.4	SETIO (Prozessabbild übertragen).....	377
9.3.5	GETIO_PART (Prozessabbildbereich lesen).....	378
9.3.6	SETIO_PART (Prozessabbildbereich übertragen)	379
9.3.7	RALRM (Alarm empfangen)	380
9.3.8	D_ACT_DP (DP-Slaves deaktivieren/aktivieren)	383
9.3.9	STATUS-Parameter für RDREC, WRREC und RALRM	388
9.3.10	Andere	393
9.3.10.1	DPRD_DAT und DPWR_DAT (Konsistente Daten lesen/schreiben).....	393
9.3.10.2	RCVREC (Datensatz empfangen).....	396
9.3.10.3	PRVREC (Datensatz bereitstellen).....	398
9.3.10.4	DPNRM_DG (Diagnosedaten eines DP-Slaves lesen)	401
9.4	PROFenergy	403
9.5	Alarme	404
9.5.1	Anweisungen ATTACH und DETACH (OB und Alarmereignis einander zuweisen/ Zuweisung aufheben)	404
9.5.2	Weckalarme	407
9.5.2.1	SET_CINT (Weckalarm parametrieren).....	407
9.5.2.2	QRY_CINT (Weckalarmparameter abfragen)	409
9.5.3	Uhrzeitalarme	410
9.5.3.1	SET_TINTL (Uhrzeitalarm festlegen).....	411
9.5.3.2	CAN_TINT (Uhrzeitalarm löschen)	412
9.5.3.3	ACT_TINT (Uhrzeitalarm aktivieren)	413
9.5.3.4	QRY_TINT (Status des Uhrzeitalarms abfragen)	413

9.5.4	Verzögerungsalarme	414
9.5.5	Anweisungen DIS_AIRT und EN_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen verzögern/aktivieren)	417
9.6	Alarme	418
9.6.1	Gen_UsrMsg (Anwenderdiagnosemeldungen erzeugen)	418
9.7	Diagnose (PROFINET oder PROFIBUS)	420
9.7.1	Diagnoseanweisungen	420
9.7.2	RD_SINFO (Startinformation des aktuellen OBs auslesen)	421
9.7.3	LED (LED-Status lesen)	430
9.7.4	Get_IM_Data (Identifikations- und Wartungsdaten lesen)	431
9.7.5	Get_Name (Namen eines PROFINET IO-Device lesen)	433
9.7.6	GetStationInfo (IP- oder MAC-Adresse eines PROFINET IO-Device lesen)	439
9.7.7	Anweisung DeviceStates	449
9.7.7.1	Beispiele für die Konfiguration von DeviceStates	450
9.7.8	Anweisung ModuleStates	454
9.7.8.1	Beispiele für die Konfiguration von ModuleStates	455
9.7.9	GET_DIAG (Diagnoseinformationen lesen)	459
9.7.10	GetSMCInfo (Informationen über die Memory Card auslesen)	466
9.7.11	Diagnoseereignisse für die dezentrale Peripherie	469
9.8	Impuls	470
9.8.1	CTRL_PWM (Impulsdauermodulation)	470
9.8.2	CTRL_PTO (Impulsfolge)	472
9.8.3	Funktionsweise der Impulsausgänge	475
9.8.4	Konfigurieren eines Impulskanals für PWM oder PTO	477
9.9	Rezepte und Datenprotokolle	482
9.9.1	Rezepte	482
9.9.1.1	Übersicht über Rezepte	482
9.9.1.2	Beispiel für ein Rezept	483
9.9.1.3	Programmanweisungen zum Übertragen von Rezeptdaten	487
9.9.1.4	Beispielprogramm für Rezepte	491
9.9.2	Datenprotokolle	493
9.9.2.1	Datensatzstruktur der Datenprotokolle	494
9.9.2.2	Programmanweisungen zum Steuern von Datenprotokollen	495
9.9.2.3	Arbeiten mit Datenprotokollen	509
9.9.2.4	Grenzwert für die Größe von Datenprotokolldateien	511
9.9.2.5	Beispielprogramm für Datenprotokolle	514
9.10	Datenbausteinststeuerung	518
9.10.1	CREATE_DB (Datenbaustein erstellen)	518
9.10.2	Anweisungen READ_DBL und WRIT_DBL (Datenbaustein im Ladespeicher lesen/schreiben)	522
9.10.3	ATTR_DB (Attribute eines Datenbausteins lesen)	525
9.10.4	DELETE_DB (Datenbaustein löschen)	527
9.11	Adressverarbeitung	528
9.11.1	GEO2LOG (Aus dem Steckplatz die Hardwarekennung ermitteln)	528
9.11.2	LOG2GEO (Steckplatz über Hardwarekennung ermitteln)	530
9.11.3	IO2MOD (Aus einer E/A-Adresse die Hardwarekennung ermitteln)	531
9.11.4	RD_ADDR (E/A-Adressen über Hardwarekennung ermitteln)	532
9.11.5	Systemdatentyp GEOADDR	534
9.12	Gemeinsame Fehlercodes für die erweiterten Anweisungen	535

9.13	Handhabung von Dateien	535
9.13.1	FileReadC: Datei von Memory Card lesen	535
9.13.2	FileWriteC: Datei auf die Memory Card schreiben	538
9.13.3	FileDelete: Datei auf der Memory Card löschen	540
10	Technologieanweisungen.....	543
10.1	Zählen (schnelle Zähler).....	543
10.1.1	Anweisung CTRL_HSC_EXT (Schnellen Zähler steuern).....	544
10.1.1.1	Übersicht über die Anweisung.....	544
10.1.1.2	Beispiel.....	545
10.1.1.3	Systemdatentypen (SDT) der Anweisung CTRL_HSC_EXT.....	548
10.1.2	Betrieb des schnellen Zählers	553
10.1.2.1	Synchronisierungsfunktion.....	553
10.1.2.2	Gate-Funktion.....	554
10.1.2.3	Erfassungsfunktion	556
10.1.2.4	Vergleichsfunktion	558
10.1.2.5	Anwendungen	558
10.1.3	Konfigurieren eines schnellen Zählers.....	560
10.1.3.1	Zählarten.....	561
10.1.3.2	Betriebsphase	562
10.1.3.3	Anfangswerte	565
10.1.3.4	Eingangsfunktionen.....	566
10.1.3.5	Ausgangsfunktion.....	567
10.1.3.6	Alarmereignisse	567
10.1.3.7	Anschlussbelegung des Hardwareeingangs	568
10.1.3.8	Anschlussbelegung des Hardwareausgangs	570
10.1.3.9	HSC-Eingangsadressen.....	570
10.1.3.10	Hardwareerkennung	571
10.1.4	Frühere Anweisung CTRL_HSC (Schnellen Zähler steuern).....	571
10.1.4.1	Übersicht über die Anweisung.....	571
10.1.4.2	Verwenden von CTRL_HSC	573
10.1.4.3	Aktueller Zählwert des HSC	573
10.2	Bewegungssteuerung	574
10.2.1	Übersicht über die Bewegungssteuerung.....	574
10.2.2	Hardwarekomponenten für die Bewegungssteuerung.....	574
10.2.3	Bewegungssteuerungsanweisungen	575
10.2.3.1	Übersicht MC-Anweisungen	575
10.2.3.2	MC_Power (Achse freigeben/sperrn).....	576
10.2.3.3	MC_Reset (Fehler bestätigen).....	577
10.2.3.4	MC_Home (Referenzpunktfahrt der Achse durchführen)	578
10.2.3.5	MC_Halt (Achse pausieren)	578
10.2.3.6	MC_MoveAbsolute (Achse absolut positionieren).....	579
10.2.3.7	MC_MoveRelative (Achse relativ positionieren).....	579
10.2.3.8	MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen).....	580
10.2.3.9	MC_MoveJog (Achse im Tippbetrieb bewegen).....	580
10.2.3.10	MC_CommandTable (Achssteuerungsbefehle als Bewegungsfolge ausführen).....	581
10.2.3.11	MC_WriteParam (Parameter eines Technologieobjekts schreiben)	581
10.2.3.12	Anweisung MC_ReadParam (Parameter des Technologieobjekts lesen).....	582
10.2.3.13	MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern)	582
10.2.4	Weitere Informationen zur Bewegungssteuerung mit S7-1200	583
10.3	PID-Regelung	584

10.3.1	PID-Funktionalität	584
10.3.2	PID-Anweisungen	584
10.3.2.1	PID-Funktionalität	584
10.3.2.2	Anweisung PID_Compact	585
10.3.2.3	Anweisung PID_3Step	586
10.3.2.4	Anweisung PID_Temp.....	587
10.3.3	Weitere Informationen zum PID-Regler der S7-1200.....	587
11	Kommunikation	589
11.1	Überblick	589
11.2	Secure Communication und Legacy-Kommunikation	591
11.3	Kommunikationsprotokolle und Ports für die Ethernet-Kommunikation	593
11.4	Asynchrone Kommunikationsverbindungen.....	595
11.5	PROFINET.....	598
11.5.1	Netzwerkverbindung erstellen.....	600
11.5.2	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren	600
11.5.3	IP-Adressen zuweisen.....	603
11.5.3.1	IP-Adressen für Programmier- und Netzwerkgeräte zuweisen	603
11.5.3.2	Ermitteln der IP-Adresse Ihres Programmiergeräts.....	606
11.5.3.3	Online eine IP-Adresse zu einer CPU zuweisen	606
11.5.3.4	IP-Adresse für eine CPU in Ihrem Projekt konfigurieren	608
11.5.4	Testen des PROFINET-Netzwerks	612
11.5.5	Ermitteln der Ethernet-Adresse (MAC-Adresse) der CPU	613
11.5.6	NTP-Synchronisation (Network Time Protocol, NTP) konfigurieren	615
11.5.7	Anlaufzeit, Benennung und Adresszuweisung von PROFINET-Geräten	616
11.5.8	Offene Benutzerkommunikation.....	617
11.5.8.1	Protokolle	617
11.5.8.2	TCP und ISO on TCP.....	618
11.5.8.3	Kommunikationsdienste und verwendete Portnummern.....	619
11.5.8.4	Ad-hoc-Modus	620
11.5.8.5	Verbindungs-IDs für Anweisungen für die offene Benutzerkommunikation	621
11.5.8.6	Parameter für die PROFINET-Verbindung.....	624
11.5.8.7	Konfigurieren eines DNS.....	631
11.5.8.8	OUC-Verbindung im TIA Portal V17 konfigurieren	632
11.5.8.9	Anweisungen TSEND_C und TRCV_C.....	635
11.5.8.10	Anweisungen TSEND_C und TRCV_C in älteren Systemen.....	649
11.5.8.11	Anweisungen TCON, TDISCON, TSEND und TRCV	656
11.5.8.12	TCONSettings.....	667
11.5.8.13	Ältere Anweisungen TCON, TDISCON, TSEND und TRCV	673
11.5.8.14	Anweisung T_RESET (bestehende Verbindung beenden und neu aufbauen).....	682
11.5.8.15	Anweisung T_DIAG (Status einer Verbindung prüfen und Informationen lesen)	684
11.5.8.16	Anweisung TMAIL_C (Email über die Ethernet-Schnittstelle der CPU senden).....	688
11.5.8.17	UDP.....	711
11.5.8.18	TUSEND und TURCV	711
11.5.8.19	T_CONFIG	717
11.5.8.20	Gemeinsame Parameter für Anweisungen	727
11.5.9	Kommunikation mit einem Programmiergerät	728
11.5.9.1	Hardware-Kommunikationsverbindung herstellen	729
11.5.9.2	Konfigurieren der Geräte	729
11.5.9.3	IP-Adressen zuweisen.....	730

11.5.9.4	Testen Ihres PROFINET-Netzwerks.....	730
11.5.10	Kommunikation HMI/PLC.....	731
11.5.10.1	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten	732
11.5.11	Kommunikation PLC/PLC.....	732
11.5.11.1	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten	733
11.5.11.2	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren	734
11.5.11.3	Sende- und Empfangsparameter konfigurieren	734
11.5.12	CPU und PROFINET IO-Gerät konfigurieren	737
11.5.12.1	PROFINET IO-Gerät hinzufügen.....	737
11.5.12.2	CPUs und Gerätenamen zuweisen	738
11.5.12.3	IP-Adressen zuweisen.....	738
11.5.12.4	IO-Zykluszeit konfigurieren	739
11.5.13	CPU und PROFINET I-Device konfigurieren	740
11.5.13.1	I-Device-Funktionalität	740
11.5.13.2	Eigenschaften und Vorteile des I-Device	741
11.5.13.3	Merkmale eines I-Device	741
11.5.13.4	Datenaustausch zwischen über- und untergeordnetem E/A-System	744
11.5.13.5	I-Device konfigurieren	747
11.5.14	Shared Devices	749
11.5.14.1	Shared-Device-Funktion	749
11.5.14.2	Beispiel: Ein Shared Device konfigurieren (GSD-Konfiguration)	752
11.5.14.3	Beispiel: Ein I-Device als Shared Device konfigurieren	757
11.5.15	Medienredundanzprotokoll (MRP)	766
11.5.15.1	Medienredundanz bei Ringtopologien	769
11.5.15.2	Einsetzen des Medienredundanzprotokolls (MRP)	770
11.5.15.3	Medienredundanz konfigurieren.....	773
11.5.16	S7-Routing.....	776
11.5.16.1	S7-Routing zwischen CPU und CP-Schnittstellen	777
11.5.16.2	S7-Routing zwischen zwei CP-Schnittstellen	778
11.5.17	SNMP deaktivieren	778
11.5.17.1	SNMP deaktivieren	779
11.5.18	Diagnose	781
11.5.19	Anweisungen für die dezentrale Peripherie.....	781
11.5.20	Diagnoseanweisungen	781
11.5.21	Diagnoseereignisse für die dezentrale Peripherie	781
11.6	PROFIBUS.....	782
11.6.1	Kommunikationsdienste der PROFIBUS-CMs	783
11.6.2	Verweis auf die Benutzerhandbücher für PROFIBUS-CMs.....	784
11.6.3	DP-Master und -Slave konfigurieren	785
11.6.3.1	CM 1243-5 (DP-Master) und DP-Slave hinzufügen.....	785
11.6.3.2	Konfigurieren der logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten	785
11.6.3.3	PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen.....	786
11.6.4	Anweisungen für die dezentrale Peripherie.....	787
11.6.5	Diagnoseanweisungen	787
11.6.6	Diagnoseereignisse für die dezentrale Peripherie	788
11.7	ASi	788
11.7.1	AS-i-Master und -Slavegeräte konfigurieren	789
11.7.1.1	AS-i-Mastermodul CM 1243-2 und AS-i-Slave hinzufügen.....	789
11.7.1.2	Logische Netzwerkverbindungen zwischen zwei AS-i-Geräten konfigurieren.....	790
11.7.1.3	Eigenschaften des AS-i-Masters CM1243-2 konfigurieren	790
11.7.1.4	Einem AS-i-Slave eine AS-i-Adresse zuweisen	791

11.7.2	Datenaustausch zwischen dem Anwenderprogramm und AS-i-Slaves	793
11.7.2.1	STEP 7 Basic konfigurieren	793
11.7.2.2	Slaves mit STEP 7 konfigurieren	795
11.7.3	Anweisungen für die dezentrale Peripherie	797
11.7.4	Mit AS-i-Online-Werkzeugen arbeiten	797
11.8	S7-Kommunikation	799
11.8.1	GET und PUT (Aus remoter CPU auslesen und schreiben)	799
11.8.2	S7-Verbindung erstellen	803
11.8.3	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren	804
11.8.4	Zuweisung von Verbindungsparametern für GET/PUT	804
11.8.4.1	Verbindungsparameter	805
11.8.4.2	S7-Verbindung von CPU zu CPU konfigurieren	807
11.9	Was tun, wenn kein Zugriff auf die CPU über die IP-Adresse möglich ist	812
11.10	OPC UA-Server	813
11.10.1	Konfiguration des OPC UA-Servers	813
11.10.1.1	Aktivierung des OPC UA-Servers	814
11.10.1.2	Verhalten des OPC UA-Servers im Betrieb	814
11.10.1.3	Einstellungen für den OPC UA-Server	816
11.10.1.4	S7-1200 als OPC UA-Server verwenden	817
11.10.2	Sicherheit des OPC UA-Servers	819
11.10.2.1	Unterstützte Sicherheitsrichtlinien	820
11.10.2.2	Vertrauenswürdige Clients	822
11.10.2.3	Benutzer-Authentifizierung	822
11.10.3	Schnittstelle des OPC UA-Servers	824
11.10.3.1	Unterstützte Datentypen	824
11.10.3.2	PLC-Darstellung	825
11.10.3.3	Ladbare Serverschnittstellen	827
11.10.4	OPC UA-Diagnosepuffer	829
11.10.4.1	OPC UA-Grenzwerte erreicht	831
11.10.4.2	Entferntes Lesen des Diagnosepuffers bei OPC UA	833
11.10.4.3	OPC UA-Sicherheitsereignisse	833
11.10.4.4	Falsche OPC UA-Verwendung	839
11.10.4.5	Zusammenfassende Meldungen für OPC UA	841
11.10.5	OPC UA-Methodenaufrufe	841
11.10.5.1	Nützliche Informationen über Servermethoden	841
11.10.5.2	Grenzbedingungen für die Verwendung von Servermethoden	845
12	Webserver	847
12.1	Webserver aktivieren	850
12.2	Konfigurieren von Webserver-Benutzern	852
12.3	Über den PC auf die Webseiten zugreifen	855
12.4	Über ein mobiles Gerät auf die Webseiten zugreifen	856
12.5	Verwenden eines CP-Moduls für den Zugriff auf die Webseiten	858
12.6	Herunterladen und Installieren eines Sicherheitszertifikats	858
12.7	Standard-Webseiten	861
12.7.1	Aufbau der Standard-Webseiten	861
12.7.2	Basisseiten	862
12.7.3	Anmeldung und Benutzerrechte	863

12.7.4	Einleitung	866
12.7.5	Start	867
12.7.6	Diagnose	868
12.7.7	Diagnosepuffer	870
12.7.8	Modulinformationen	872
12.7.9	Kommunikation	876
12.7.10	Variablenstatus	879
12.7.11	Beobachtungstabellen	881
12.7.12	Online-Sicherung	883
12.7.13	Datenprotokollseite	886
12.7.14	Anwenderdateien	888
12.7.15	API für Datenprotokolle und Anwenderdateien	893
12.7.16	Dateibrowser	893
12.8	Benutzerdefinierte Webseiten	894
12.8.1	HTML-Seiten anlegen	895
12.8.2	Vom S7-1200 Webserver unterstützte AWP-Befehle	896
12.8.2.1	Variablen lesen	898
12.8.2.2	Variablen schreiben	899
12.8.2.3	Sondervariablen lesen	901
12.8.2.4	Sondervariablen schreiben	903
12.8.2.5	Alias für einen Variablenverweis nutzen	904
12.8.2.6	Enum-Typen definieren	905
12.8.2.7	CPU-Variablen mit einem Enum-Typ referenzieren	905
12.8.2.8	Fragmente erstellen	907
12.8.2.9	Fragmente importieren	908
12.8.2.10	Definitionen verbinden	909
12.8.2.11	Handhabung von Variablennamen mit Sonderzeichen	909
12.8.3	Verwendung von benutzerdefinierten Webseiten konfigurieren	911
12.8.4	Konfigurieren der Einstiegsseite	913
12.8.5	WWW-Anweisung für benutzerdefinierte Webseiten programmieren	913
12.8.6	Programmbausteine in die CPU laden	915
12.8.7	Zugriff auf die benutzerdefinierten Webseiten	915
12.8.8	Einschränkungen bei benutzerdefinierten Webseiten	916
12.8.9	Beispiel für eine benutzerdefinierte Webseite	917
12.8.9.1	Webseite zum Beobachten und Steuern einer Windturbine	917
12.8.9.2	Steuerungsdaten lesen und anzeigen	919
12.8.9.3	Enum-Typ verwenden	920
12.8.9.4	Benutzereingaben in die Steuerung schreiben	921
12.8.9.5	Sondervariablen schreiben	922
12.8.9.6	Referenz: HTML-Code der Webseite "Remote Wind Turbine Monitor"	922
12.8.9.7	Konfiguration der Beispiel-Webseite in STEP 7	927
12.8.10	Benutzerdefinierte Webseiten in mehreren Sprachen einrichten	928
12.8.10.1	Ordnerstruktur anlegen	928
12.8.10.2	Sprachumschaltung programmieren	929
12.8.10.3	STEP 7 für die Verwendung einer mehrsprachigen Seitenstruktur konfigurieren	931
12.8.11	Erweiterte Steuerung von benutzerdefinierten Webseiten	932
12.8.12	Web-API	935
12.8.12.1	Web-API	935
12.8.12.2	Unterstützte Web-API-Methoden	936
12.9	Einschränkungen	937
12.9.1	Verwendung von JavaScript	938

12.9.2	Eingeschränkte Funktionen, wenn Cookies in den Internetoptionen nicht erlaubt sind	938
12.9.3	Regeln für die Eingabe von Variablennamen und Werten	939
12.9.4	Datenprotokolle im CSV-Format in nicht amerikanische/englische Versionen von Microsoft Excel importieren	939
13	Kommunikationsprozessor und Modbus-TCP	941
13.1	Mit den seriellen Kommunikationsschnittstellen arbeiten.....	941
13.2	Abschließen eines RS485-Busanschlussteckers	942
13.3	Punkt-zu-Punkt-Kommunikation (PtP)	943
13.3.1	PtP, frei programmierbare Kommunikation	944
13.3.2	3964(R)-Kommunikation.....	946
13.3.3	Konfigurieren der frei programmierbaren PtP-Kommunikation.....	947
13.3.3.1	Steuerung der Flusskontrolle	949
13.3.3.2	Sendeparameter konfigurieren	950
13.3.3.3	Empfangsparameter konfigurieren	952
13.3.4	Konfigurieren der 3964(R)-Kommunikation	960
13.3.4.1	Konfigurieren der 3964(R)-Kommunikationsports	960
13.3.4.2	Konfigurieren von 3964(R)-Priorität und Protokollparametern	961
13.3.5	Punkt-zu-Punkt-Anweisungen	963
13.3.5.1	Gemeinsame Parameter für Punkt-zu-Punkt-Operationen.....	963
13.3.5.2	Port_Config (Kommunikationsparameter dynamisch konfigurieren).....	965
13.3.5.3	Send_Config (Parameter für die serielle Kommunikation dynamisch konfigurieren).....	968
13.3.5.4	Receive_Config (Parameter für den seriellen Empfang dynamisch konfigurieren)	970
13.3.5.5	P3964_Config (3964(R)-Protokoll konfigurieren)	975
13.3.5.6	Send_P2P (Sendepufferdaten übertragen)	977
13.3.5.7	Receive_P2P (Meldungsempfang aktivieren).....	980
13.3.5.8	Receive_Reset (Empfangspuffer löschen)	982
13.3.5.9	Signal_Get (RS-232-Signale abfragen)	983
13.3.5.10	Signal_Set (RS-232-Signale festlegen)	984
13.3.5.11	Get_Features	985
13.3.5.12	Set_Features.....	986
13.3.6	Programmieren der PtP-Kommunikation	987
13.3.6.1	Abfragearchitektur.....	988
13.3.7	Beispiel: Punkt-zu-Punkt-Kommunikation.....	989
13.3.7.1	Kommunikationsmodul konfigurieren	990
13.3.7.2	Betriebsarten RS422 und RS485	993
13.3.7.3	STEP 7-Programm programmieren.....	995
13.3.7.4	Terminalemulator konfigurieren	997
13.3.7.5	Beispielprogramm ausführen	997
13.4	Kommunikation über die universelle serielle Schnittstelle (USS)	998
13.4.1	Version der USS-Anweisungen auswählen	1000
13.4.2	Voraussetzungen für den Einsatz des USS-Protokolls	1002
13.4.3	USS-Anweisungen.....	1004
13.4.3.1	USS_Port_Scan (Kommunikation über USS-Netzwerk bearbeiten).....	1004
13.4.3.2	USS_Drive_Control (Daten mit Antrieb tauschen).....	1006
13.4.3.3	USS_Read_Param (Parameter aus dem Antrieb auslesen).....	1009
13.4.3.4	USS_Write_Param (Parameter im Antrieb ändern)	1010
13.4.4	USS-Zustandscodes.....	1012
13.4.5	Allgemeine Voraussetzungen für die USS-Antriebseinrichtung	1014
13.4.6	Beispiel: Allgemeine USS-Antriebsverbindung und -einrichtung	1014

13.5	Modbus-Kommunikation.....	1017
13.5.1	Überblick zur Kommunikation mittels Modbus RTU und Modbus TCP.....	1017
13.5.2	Modbus TCP.....	1020
13.5.2.1	Übersicht.....	1020
13.5.2.2	Version der Modbus TCP-Anweisungen auswählen	1021
13.5.2.3	Modbus TCP-Anweisungen.....	1022
13.5.2.4	Beispiele für Modbus TCP	1084
13.5.3	Modbus RTU	1089
13.5.3.1	Übersicht.....	1089
13.5.3.2	Version der Modbus RTU-Anweisungen auswählen	1091
13.5.3.3	Maximale Anzahl unterstützter Modbus-Slaves	1091
13.5.3.4	Modbus RTU-Anweisungen	1092
13.5.3.5	Modbus RTU Beispiele	1113
13.6	PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)	1117
13.6.1	Ältere Punkt-zu-Punkt-Anweisungen.....	1117
13.6.1.1	PORT_CFG (Kommunikationsparameter dynamisch konfigurieren)	1117
13.6.1.2	SEND_CFG (Sendekonfiguration)	1119
13.6.1.3	RCV_CFG (Empfangskonfiguration).....	1120
13.6.1.4	SEND_PTP (Sendepufferdaten übertragen).....	1125
13.6.1.5	RCV_PTP (Empfangsmeldungen aktivieren).....	1128
13.6.1.6	RCV_RST (Empfangspuffer löschen)	1129
13.6.1.7	SGN_GET (RS232-Signale abfragen)	1130
13.6.1.8	SGN_SET (RS-232-Signale einstellen).....	1131
13.7	USS-Kommunikation in älteren Systemen (nur CM/CB 1241)	1132
13.7.1	Version der USS-Anweisungen auswählen	1133
13.7.2	Voraussetzungen für den Einsatz des USS-Protokolls	1135
13.7.3	Ältere USS-Anweisungen.....	1137
13.7.3.1	Anweisung USS_PORT (Kommunikation über USS-Netzwerk bearbeiten)	1137
13.7.3.2	Anweisung USS_DRV (Daten mit Antrieb tauschen)	1139
13.7.3.3	Anweisung USS_RPM (Parameter aus dem Antrieb auslesen)	1142
13.7.3.4	Anweisung USS_WPM (Parameter im Antrieb ändern).....	1143
13.7.4	Alte USS-Statuscodes	1145
13.7.5	Allgemeine Voraussetzungen für die Antriebseinrichtung mit der alten Anweisung USS ..	1147
13.8	Modbus TCP-Kommunikation in älteren Systemen	1147
13.8.1	Übersicht.....	1147
13.8.2	Version der Modbus TCP-Anweisungen auswählen	1147
13.8.3	Ältere Modbus TCP-Anweisungen	1148
13.8.3.1	MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren).....	1148
13.8.3.2	MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren)	1154
13.8.4	Ältere Modbus TCP-Beispiele	1160
13.8.4.1	Beispiel: MB_SERVER für mehrere TCP-Verbindungen in älteren Systemen	1160
13.8.4.2	Beispiel: MB_CLIENT 1 in älteren Systemen: Mehrere Anforderungen mit gemeinsamer TCP-Verbindung	1161
13.8.4.3	Beispiel: MB_CLIENT 2 in älteren Systemen: Mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen	1162
13.8.4.4	Beispiel: MB_CLIENT 3 in älteren Systemen: Schreibenanforderung für das Prozessabbild der Ausgänge	1163
13.8.4.5	Beispiel: MB_CLIENT 4 in älteren Systemen: Mehrere Anforderungen koordinieren	1163
13.9	Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241).....	1164
13.9.1	Übersicht.....	1164

13.9.2	Version der Modbus RTU-Anweisungen auswählen	1165
13.9.3	Ältere Modbus RTU-Anweisungen	1166
13.9.3.1	MB_COMM_LOAD (Port am PtP-Modul für Modbus RTU konfigurieren)	1166
13.9.3.2	MB_MASTER (Über den PtP-Port als Modbus RTU-Master kommunizieren)	1169
13.9.3.3	MB_SLAVE (Über den PtP-Port als Modbus RTU-Slave kommunizieren)	1175
13.9.4	Beispiel für Modbus RTU in älteren Systemen.....	1181
13.9.4.1	Beispiel: Beispielprogramm für einen Modbus RTU-Master in älteren Systemen	1181
13.9.4.2	Beispiel: Beispielprogramm für einen Modbus RTU-Slave in älteren Systemen.....	1183
13.10	Industrial Remote Communication (IRC)	1185
13.10.1	Übersicht über Telecontrol-CPs	1185
13.10.2	Anschluss an ein GSM-Netz	1188
13.10.3	Anwendungen des CP 1242-7	1189
13.10.4	Weitere Eigenschaften des CP 1242-7.....	1191
13.10.5	Weitere Informationen	1191
13.10.6	Zubehör	1191
13.10.7	Konfigurationsbeispiele für Telecontrol.....	1192
14	TeleService-Kommunikation (SMTP-E-Mail)	1197
14.1	Anweisung TM_Mail (E-Mail senden).....	1197
15	Online- und Diagnose-Tools.....	1205
15.1	Status-LEDs.....	1205
15.2	Online-Verbindung mit einer CPU herstellen.....	1209
15.3	Einem PROFINET IO-Gerät online einen Namen zuweisen.....	1210
15.4	Einstellen der IP-Adresse und der Uhrzeit.....	1212
15.5	Firmware aktualisieren.....	1212
15.6	Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten festlegen oder löschen.....	1213
15.7	Rücksetzen auf Werkseinstellungen	1214
15.8	Prüfen eines Moduls auf Defekte (Servicedaten speichern)	1215
15.9	Formatierung einer SIMATIC Memory Card über STEP 7	1217
15.10	Bedienpanel für die Online-CPU.....	1218
15.11	Überwachung von Zykluszeit und Speicherauslastung	1219
15.12	Diagnoseereignisse in der CPU anzeigen.....	1219
15.13	Vergleichen von Offline- und Online-CPUs.....	1221
15.14	Durchführen eines Online/Offline-Topologievergleichs	1221
15.15	Werte in der CPU beobachten und steuern.....	1223
15.15.1	Online gehen, um die Werte in der CPU zu beobachten.....	1223
15.15.2	Zustand im Programmiereditor anzeigen	1224
15.15.3	Erfassen einer Momentaufnahme der Online-Werte eines DBs zum Wiederherstellen von Werten	1225
15.15.4	Werte in der CPU über die Beobachtungstabelle beobachten und steuern	1226
15.15.4.1	Variablen mit Trigger beobachten oder steuern.....	1228
15.15.4.2	Ausgänge im Betriebszustand STOP freischalten	1229
15.15.5	Werte in der CPU forcen	1229
15.15.5.1	Arbeiten mit der Forcetabelle	1229

15.15.5.2	Funktionsweise der Forcefunktion	1230
15.16	Laden im Betriebszustand RUN	1232
15.16.1	Voraussetzungen für "Laden im Betriebszustand RUN"	1233
15.16.2	Ändern des Programms im Betriebszustand RUN	1234
15.16.3	Ausgewählte Bausteine laden	1235
15.16.4	Einen einzelnen ausgewählten Baustein mit einem Übersetzungsfehler in einem anderen Baustein laden	1236
15.16.5	Bestehende Bausteine im Betriebszustand RUN ändern und ins Zielsystem laden	1237
15.16.6	Systemreaktion bei fehlgeschlagenem Ladevorgang	1240
15.16.7	Sicherheitsaspekte beim Laden im Betriebszustand RUN	1240
15.17	CPU-Daten bei Auslösebedingungen verfolgen und aufzeichnen	1242
15.18	Ermitteln der Art eines Drahtbruchs über ein Modul SM 1231	1244
15.19	Sichern und Wiederherstellen einer CPU	1247
15.19.1	Optionen zum Sichern und Wiederherstellen	1247
15.19.2	Sichern einer Online-CPU	1248
15.19.3	Wiederherstellen einer CPU	1251
A	Technische Daten	1253
A.1	Siemens-Website für Online-Support	1253
A.2	Allgemeine technische Daten	1253
A.3	Anschlussbelegung PROFINET-Schnittstellenport X1	1262
A.4	CPU 1211C	1263
A.4.1	Allgemeine technische Daten und Leistungsmerkmale	1263
A.4.2	Von der CPU 1211C unterstützte Zeiten, Zähler und Codebausteine	1265
A.4.3	Digitale Eingänge und Ausgänge	1269
A.4.4	Analoge Eingänge	1271
A.4.4.1	Schrittantwort der integrierten analogen Eingänge der CPU	1271
A.4.4.2	Abtastzeit der integrierten analogen Ports der CPU	1271
A.4.4.3	Messbereiche der analogen Eingänge für Spannung (CPUs)	1272
A.4.5	Schaltpläne der CPU 1211C	1273
A.5	CPU 1212C	1276
A.5.1	Allgemeine technische Daten und Leistungsmerkmale	1276
A.5.2	Von der CPU 1212C unterstützte Zeiten, Zähler und Codebausteine	1278
A.5.3	Digitale Eingänge und Ausgänge	1281
A.5.4	Analoge Eingänge	1283
A.5.4.1	Schrittantwort der integrierten analogen Eingänge der CPU	1284
A.5.4.2	Abtastzeit der integrierten analogen Ports der CPU	1284
A.5.4.3	Messbereiche der analogen Eingänge für Spannung (CPUs)	1284
A.5.5	Schaltpläne der CPU 1212C	1285
A.6	CPU 1214C	1288
A.6.1	Allgemeine technische Daten und Leistungsmerkmale	1288
A.6.2	Von der CPU 1214C unterstützte Zeiten, Zähler und Codebausteine	1290
A.6.3	Digitale Eingänge und Ausgänge	1293
A.6.4	Analoge Eingänge	1295
A.6.4.1	Schrittantwort der integrierten analogen Eingänge der CPU	1296
A.6.4.2	Abtastzeit der integrierten analogen Ports der CPU	1296
A.6.4.3	Messbereiche der analogen Eingänge für Spannung (CPUs)	1296
A.6.5	Schaltpläne der CPU 1214C	1297

A.7	CPU 1215C	1300
A.7.1	Allgemeine technische Daten und Leistungsmerkmale.....	1300
A.7.2	Von der CPU 1215C unterstützte Zeiten, Zähler und Codebausteine	1302
A.7.3	Digitale Eingänge und Ausgänge.....	1306
A.7.4	Analogeingänge und -ausgänge	1308
A.7.4.1	Schrittantwort der integrierten analogen Eingänge der CPU.....	1308
A.7.4.2	Abtastzeit der integrierten analogen Ports der CPU	1308
A.7.4.3	Messbereiche der analogen Eingänge für Spannung (CPUs)	1309
A.7.4.4	Technische Daten der Analogausgänge.....	1309
A.7.5	Schaltpläne der CPU 1215C.....	1310
A.8	CPU 1217C	1314
A.8.1	Allgemeine technische Daten und Leistungsmerkmale.....	1314
A.8.2	Von der CPU 1217C unterstützte Zeiten, Zähler und Codebausteine	1316
A.8.3	Digitale Eingänge und Ausgänge.....	1320
A.8.4	Analogeingänge und -ausgänge	1323
A.8.4.1	Technische Daten der analogen Eingänge.....	1323
A.8.4.2	Schrittantwort der integrierten analogen Eingänge der CPU.....	1324
A.8.4.3	Abtastzeit der integrierten analogen Ports der CPU	1324
A.8.4.4	Messbereiche der analogen Eingänge für Spannung (CPUs)	1325
A.8.4.5	Technische Daten der Analogausgänge.....	1325
A.8.5	Schaltpläne der CPU 1217C.....	1326
A.8.6	CPU 1217C Differential Eingang (DI) Detail und Anwendungsbeispiel.....	1328
A.8.7	CPU 1217C Differential Ausgang (DO) Detail und Anwendungsbeispiel.....	1329
A.9	Digitale Signalmodule (SMs)	1330
A.9.1	Technische Daten für das digitale Eingangsmodul SM 1221	1330
A.9.2	Technische Daten für das digitale Ausgangsmodul SM 1222 mit 8 Ausgängen.....	1332
A.9.3	Technische Daten für das digitale Ausgangsmodul SM 1222 mit 16 Ausgängen.....	1333
A.9.4	Technische Daten für das digitale Eingabe-/Ausgabemodul SM 1223 (V DC)	1339
A.9.5	Technische Daten für das digitale Eingabe-/Ausgabemodul SM 1223 (V AC)	1345
A.10	Analoge Signalmodule (SMs)	1347
A.10.1	Technische Daten des SM 1231 Analogeingabemoduls	1347
A.10.2	Technische Daten des SM 1232 Analogausgabemoduls.....	1352
A.10.3	Technische Daten des SM 1234 Analogein-/Analogausgabemoduls	1354
A.10.4	Schrittantwort der analogen Eingänge	1357
A.10.5	Abtastzeit und Aktualisierungszeiten der Analogeingänge	1357
A.10.6	Messbereiche der analogen Eingänge für Spannung und Strom (SB und SM).....	1358
A.10.7	Messbereiche der analogen Ausgänge für Spannung und Strom (SB und SM).....	1359
A.11	Thermoelement- und RTD-Signalmodule (SMs).....	1360
A.11.1	SM 1231 Thermoelement	1360
A.11.1.1	Grundlegende Funktionsweise eines Thermoelements.....	1362
A.11.1.2	Auswahltabellen für das SM 1231 Thermoelement	1363
A.11.2	SM 1231 RTD	1366
A.11.2.1	Auswahltabellen für das SM 1231 RTD	1369
A.12	Technologiemodule	1372
A.12.1	SM 1278 4xIO-Link-Master SM	1372
A.12.1.1	SM 1278 4xIO-Link-Master - Überblick	1375
A.12.1.2	Anschluss	1377
A.12.1.3	Parameter/Adressbereich.....	1379
A.12.1.4	Alarm-, Fehler- und Systemmeldungen.....	1382

A.13	Digitale Signalboards (SBs).....	1385
A.13.1	Technische Daten des SB 1221 200 kHz Digitaleingabe	1385
A.13.2	Technische Daten des SB 1222 200 kHz Digitalausgabe	1387
A.13.3	Technische Daten des SB 1223 200 kHz Digitalein-/Digitalausgabe	1390
A.13.4	Technische Daten für das SB 1223 mit 2 x 24-V-DC-Eingang / 2 x 24-V-DC-Ausgang	1393
A.14	Analoge Signalboards (SBs).....	1396
A.14.1	Technische Daten des SB 1231 1 Analogeingang	1396
A.14.2	Technische Daten des SB 1232 1 Analogausgabe	1398
A.14.3	Messbereiche der analogen Eingänge und Ausgänge	1400
A.14.3.1	Schrittantwort der analogen Eingänge	1400
A.14.3.2	Abtastzeit und Aktualisierungszeiten der Analogeingänge	1400
A.14.3.3	Messbereiche der analogen Eingänge für Spannung und Strom (SB und SM).....	1400
A.14.3.4	Messbereiche der analogen Ausgänge für Spannung und Strom (SB und SM).....	1401
A.14.4	Thermoelement-Signalboards (SBs).....	1403
A.14.4.1	Technische Daten des SB 1231 1 Analogeingang Thermoelement	1403
A.14.4.2	Grundlegende Funktionsweise eines Thermoelements.....	1404
A.14.5	RTD-Signalboards (SBs)	1408
A.14.5.1	Technische Daten des SB 1231 1 Analogeingang RTD	1408
A.14.5.2	Auswahltabellen für das SB 1231 RTD	1411
A.15	BB 1297 Batterieboard	1413
A.16	Kommunikationsschnittstellen	1414
A.16.1	PROFIBUS.....	1414
A.16.1.1	CM 1242-5 PROFIBUS DP-SLAVE	1414
A.16.1.2	Anschlussbelegung der Sub-D-Buchse des CM 1242-5	1416
A.16.1.3	CM 1243-5 PROFIBUS DP-Master	1416
A.16.1.4	Anschlussbelegung der Sub-D-Buchse des CM 1243-5	1418
A.16.2	CP 1242-7.....	1418
A.16.2.1	CP 1242-7 GPRS	1419
A.16.2.2	GSM/GPRS-Antenne ANT794-4MR	1420
A.16.2.3	Flachantenne ANT794-3M	1421
A.16.3	CM 1243-2 AS-i-Master	1422
A.16.3.1	Technische Daten des AS-i Master CM 1243-2.....	1422
A.16.3.2	Elektrischer Anschluss des AS-i-Masters	1423
A.16.4	RS232, RS422 und RS485.....	1424
A.16.4.1	Technische Daten des CB 1241 RS485	1424
A.16.4.2	Technische Daten des CM 1241 RS232	1427
A.16.4.3	Technische Daten des CM 1241 RS422/485	1428
A.17	TeleService (TS-Adapter und TS-Adaptermodul)	1429
A.18	SIMATIC Memory Cards	1430
A.19	Eingangssimulatoren	1430
A.20	S7-1200 Potentiometermodul	1432
A.21	Steckleitung für Erweiterungsmodule.....	1433
A.22	Zugehörige Produkte	1434
A.22.1	PM 1207 Stromversorgungsmodul	1434
A.22.2	CSM 1277 Compact Switch Module.....	1434
A.22.3	CM CANopen-Modul	1434
A.22.4	Kommunikationsmodul RF120C	1435

A.22.5	SM 1238 Energy Meter.....	1435
A.22.6	SIWAREX Wägeelektronik	1436
B	Berechnung der Leistungsbilanz	1437
C	Bestellinformationen	1441
C.1	CPU-Module.....	1441
C.2	Signalmodule (SMs), Signalboards (SBs) und Batterieboards (BBs)	1441
C.3	Kommunikation	1443
C.4	Fehlersichere CPUs und Signalmodule	1444
C.5	Sonstige Module	1445
C.6	Memory Cards	1445
C.7	Grundlegende HMI-Geräte	1445
C.8	Ersatzteile und sonstige Hardware.....	1446
C.9	Programmiersoftware	1452
C.10	OPC UA-Lizenzen	1453
D	Geräteaustausch und Ersatzteilkompatibilität	1455
D.1	Eine CPU mit Schutz vertraulicher PLC-Konfigurationsdaten ersetzen.....	1455
D.2	Ersatz einer CPU V3.0 durch eine CPU V4.x	1456
D.3	S7-1200 bis V3.0 - Ersatzklemmenblöcke	1462
	Index.....	1465

Produktübersicht

1.1 Einführung in den S7-1200 PLC

Die Steuerung S7-1200 bietet Ihnen die erforderliche Flexibilität und Leistung zur Steuerung einer breiten Palette von Geräten für Ihre Automatisierungslösungen. Durch den kompakten Aufbau, die flexible Konfiguration und einen leistungsstarken Befehlssatz eignet sich die S7-1200 hervorragend für eine große Bandbreite von Steuerungsanwendungen.

Die CPU verbindet die folgenden und weitere Elemente in einem kompakten Gehäuse zu einer leistungsstarken Steuerung:

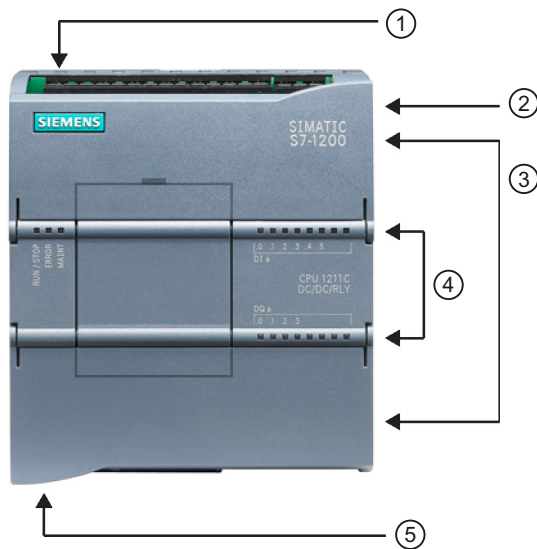
- einen Mikroprozessor
- ein integriertes Netzteil
- Ein- und Ausgangstromkreise
- Integriertes PROFINET
- Hochgeschwindigkeitsein-/ausgänge für die Bewegungssteuerung

Nachdem Sie Ihr Programm geladen haben, enthält die CPU die erforderliche Logik, damit Sie die Geräte in Ihrer Anwendung beobachten und steuern können. Die CPU beobachtet Eingänge und ändert Ausgänge anhand der Befehle Ihres Anwenderprogramms, das Boolesche Verknüpfungen, Zähl- und Zeitfunktionen, komplexe arithmetische Operationen, Bewegungssteuerung und Kommunikation mit anderen intelligenten Geräten umfassen kann.

Die CPU verfügt über einen PROFINET-Port zur Kommunikation über ein PROFINET-Netzwerk. Für die Kommunikation über Netzwerke und verschiedene Protokolle stehen weitere Module zur Verfügung:

- PROFIBUS
- GPRS
- LTE
- WAN mit integrierten Sicherheitsfunktionen (Firewall, VPN)
- RS485
- RS232
- RS422
- IEC 60870
- DNP3
- USS
- MODBUS

1.1 Einführung in den S7-1200 PLC



- ① Stromanschluss
- ② Steckplatz für eine Memory Card unter der oberen Abdeckklappe
- ③ Steckbarer Klemmenblock für die Anwenderverdrahtung (hinter den Abdeckklappen)
- ④ Status-LEDs für die integrierten E/A
- ⑤ PROFINET-Anschluss (auf der Unterseite der CPU)

Verschiedene Sicherheitsfunktionen schützen den Zugriff auf die CPU und das Steuerungsprogramm:

- Passwortschutz (Seite 163), mit dem der Zugriff auf die CPU-Funktionen nach Bedarf eingerichtet werden kann.
- Knowhow-Schutz (Seite 167), um den Code in einem bestimmten Baustein zu verbergen.
- Kopierschutz (Seite 168), um Ihr Programm mit einer bestimmten Memory Card oder CPU zu verknüpfen.
- Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161)
- Sichere Kommunikation für PG/PC und HMI (Seite 166)

Tabelle 1-1 Vergleich der CPU-Varianten

Merkmal		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C	
Abmessungen (mm)		90 x 100 x 75			110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Anwenderspeicher	Arbeitsspeicher	50 kB	75 kB	100 kB	125 kB	150 kB	
	Ladespeicher	1 MB	2 MB	4 MB			
	Remanent	14 kB					
Lokale integrierte E/A	Digital	6 Eingänge/ 4 Ausgänge	8 Eingänge/ 6 Ausgänge	14 Eingänge/ 10 Ausgänge			
	Analog	2 Eingänge			2 Eingänge/2 Ausgänge		
Größe des Prozessabbilds	Eingänge (E)	1024 Byte					
	Ausgang (A)	1024 Byte					
Merker (M)		4096 Byte		8192 Byte			
Erweiterung: Signalmodul (SM)		Keine	2	8			

Merkmal		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Signalboard (SB), Batterieboard (BB) oder Kommunikationsboard (CB)		1				
Kommunikationsmodule (CM) (Anbau links)		3				
Schnelle Zähler	Summe	Bis zu 6 konfiguriert für die Verwendung integrierter oder SB-Eingänge				
	1 MHz	-				Eb.2 bis Eb.5
	100/180 kHz	Ea.0 bis Ea.5				
	30/20 kHz	--	Ea.6 bis Ea.7	Ea.6 bis Eb.5		Ea.6 bis Eb.1
	200 kHz ³					
Impulsausgänge ²	Summe	Bis zu 4 konfiguriert für die Verwendung integrierter oder SB-Ausgänge				
	1 MHz	--				Aa.0 bis Aa.3
	100 kHz	Aa.0 bis Aa.3				Aa.4 bis Ab.1
	20 kHz	--	Aa.4 bis Aa.5	Aa.4 bis Ab.1		--
Memory Card		SIMATIC Memory Card (optional)				
Datenprotokolle	Anzahl	Maximal 8 gleichzeitig geöffnet				
	Größe	500 MB pro Datenprotokoll, oder wie begrenzt durch maximal verfügbaren Ladespeicher				
Pufferung Echtzeituhr		20 Tage, typ./min. 12 Tage bei 40 Grad C (wartungsfreier Hochleistungskondensator)				
PROFINET Ethernet-Kommunikationsport		1			2	
Ausführungszeit arithm. Operationen		2,3 µs/Operation				
Ausführungszeit Boolesche Operationen		0,08 µs/Operation				

¹ Die niedrigere Geschwindigkeit ist gültig, wenn der HSC für den A/B-Zählerbetrieb konfiguriert ist.

² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

³ Bis zu 200 kHz sind mit dem SB 1221 DI x 24 V DC 200 kHz und SB 1221 DI 4 x 5 V DC 200 kHz verfügbar.

Die verschiedenen CPU-Ausführungen bieten eine Vielfalt an Leistungsmerkmalen und Funktionen, damit Sie effektive Lösungen für verschiedenste Anwendungen erstellen können. Ausführliche Informationen zu bestimmten CPUs finden Sie in den technischen Daten (Seite 1253).

1.1 Einführung in den S7-1200 PLC

Tabelle 1-2 Von der S7-1200 unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung					
Bausteine	Typ	OB, FB, FC, DB					
	Größe	CPU-Variante	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
		Codebausteine	50 KB	64 KB	64 KB	64 KB	64 KB
		Verknüpfte ¹ Datenbausteine	50 KB	75 KB	100 KB	125 KB	150 KB
	Nicht verknüpfte ² Datenbausteine	256 KB	256 KB	256 KB	256 KB	256 KB	
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)					
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ³					
Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.						
OBs	Programmzyklus	Mehrere					
	Anlauf	Mehrere					
	Verzögerungsalarme	4 (1 pro Ereignis)					
	Weckalarme	4 (1 pro Ereignis)					
	Prozessalarme	50 (1 pro Ereignis)					
	Zeitfehleralarme	1					
	Diagnosefehleralarme	1					
	Ziehen oder Stecken von Modulen	1					
	Fehler bei Baugruppenträger oder Station	1					
	Uhrzeitalarm	Mehrere					
	Zustand	1					
	Update	1					
	Profil	1					
	Zeiten	Typ	IEC				
Anzahl		Nur durch die Speicherkapazität begrenzt					
Speicherung		Struktur im DB, 16 Bytes pro Zeit					
Zähler	Typ	IEC					
	Anzahl	Nur durch die Speicherkapazität begrenzt					
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Byte • Int, UInt: 6 Byte • DInt, UDInt: 12 Byte 					

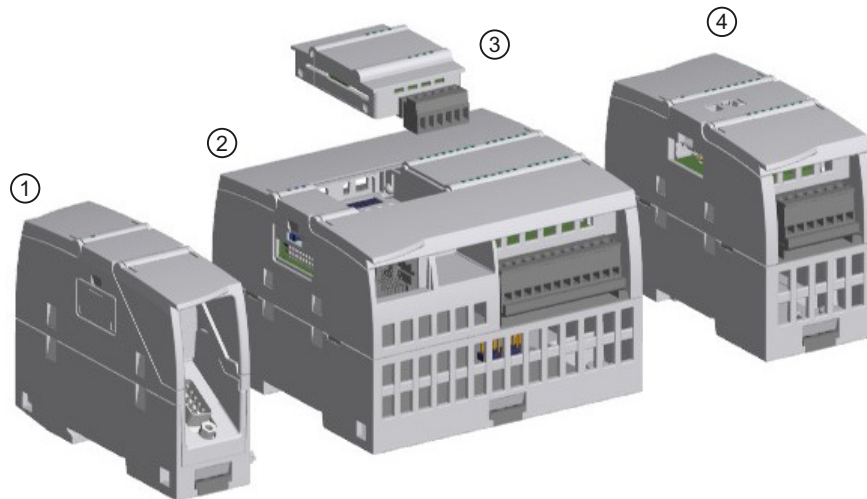
¹ In Arbeitsspeicher und Ladespeicher gespeichert. Kann die Größe des freien Arbeits- oder Ladespeichers nicht überschreiten.

² Nur im Ladespeicher gespeichert

³ Sicherheitsprogramme nutzen zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

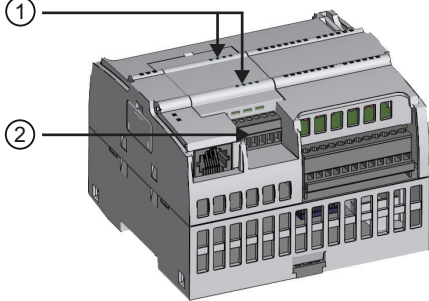
1.2 Erweiterungsfähigkeit der CPU

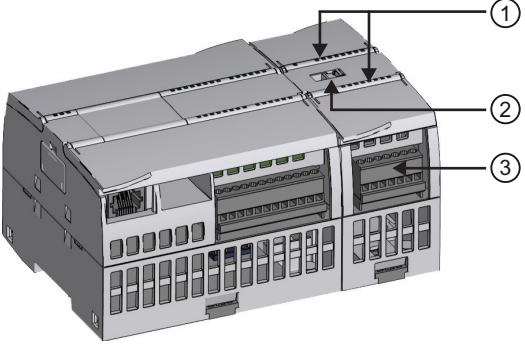
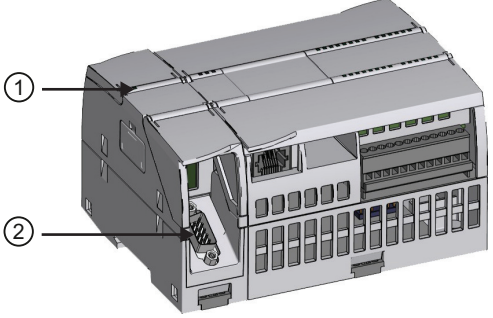
Die Produktfamilie S7-1200 bietet eine Vielzahl von Modulen und steckbaren Boards zur Erweiterung der CPU um zusätzliche E/A oder andere Kommunikationsprotokolle. Ausführliche Informationen zu bestimmten Modulen finden Sie in den technischen Daten (Seite 1253).



- ① Kommunikationsmodul (CM) oder Kommunikationsprozessor (CP) (Seite 1414)
- ② CPU (CPU 1211C (Seite 1263), CPU 1212C (Seite 1276), CPU 1214C (Seite 1288), CPU 1215C (Seite 1300), CPU 1217C (Seite 1314))
- ③ Signalboard (SB) (digitales SB (Seite 1385), analoges SB (Seite 1396)), Kommunikationsboard (CB) (Seite 1424) oder Batterieboard (BB) CPU (CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C, CPU 1217C) (Seite 1413)
- ④ Signalmodul (SM) (digitales SM (Seite 1330), analoges SM (Seite 1347), Thermoelement-SM (Seite 1360), RTD-SM (Seite 1366), Technologie-SM) (Seite 1372)

Tabelle 1-3 S7-1200 Erweiterungsmodule

Art des Moduls	Beschreibung
<p>Die CPU unterstützt ein steckbares Erweiterungsboard:</p> <ul style="list-style-type: none"> • Ein Signalboard (SB) stellt zusätzliche E/A für Ihre CPU bereit. Das SB wird auf der Vorderseite der CPU angeschlossen. • Mit einem Kommunikationsboard (CB) können Sie Ihre CPU um einen Kommunikationsanschluss erweitern. • Ein Batterieboard (BB) gewährleistet die langfristige Pufferung der Echtzeituhr. 	 <ul style="list-style-type: none"> ① Status-LEDs am SB ② Steckbarer Klemmenblock für die Anwenderverdrahtung

Art des Moduls	Beschreibung
<p>Signalmodule (SMs) erweitern die CPU um zusätzliche Funktionalität. SMs werden an der rechten Seite der CPU angeschlossen.</p> <ul style="list-style-type: none"> • Digitale E/A • Analoge E/A • RTD und Thermoelement • SM 1278 IO-Link-Master • SM 1238 Energy Meter (https://support.industry.siemens.com/cs/ww/de/view/109483435) 	<div style="text-align: right;">  </div> <ul style="list-style-type: none"> ① Status-LEDs ② Schiebelasche Busstecker ③ Steckbarer Klemmenblock für die Anwenderverdrahtung
<p>Kommunikationsmodule (CMs) und Kommunikationsprozessoren (CPs) erweitern die CPU um verschiedene Kommunikationsmöglichkeiten, z. B. PROFIBUS oder RS232/RS485 (für PtP, Modbus oder USS) oder AS-i-Master.</p> <p>Ein CP bietet Möglichkeiten für andere Arten der Kommunikation, z. B. für den Anschluss der CPU über ein GPRS-, LTE- IEC-, DNP3- oder WDC-Netzwerk.</p> <ul style="list-style-type: none"> • Die CPU unterstützt maximal 3 CMs oder CPs. • Jedes CM bzw. jeder CP wird an der linken Seite der CPU angeschlossen (bzw. an der linken Seite eines anderen CMs oder CPs). 	<div style="text-align: right;">  </div> <ul style="list-style-type: none"> ① Status-LEDs ② Kommunikationsstecker

1.3 Grundlegende HMI-Panels

Die SIMATIC HMI Basic Panels bieten Geräte mit Touchscreen für grundlegende Aufgaben im Bereich Bedienen und Beobachten. Alle Panels weisen die Schutzklasse IP65 auf und sind nach CE, UL, cULus und NEMA 4x zertifiziert.

Die erhältlichen Basic HMI Panels (Seite 1445) sind nachstehend beschrieben:

- KTP400 Basic: 4"-Touchscreen mit 4 konfigurierbaren Tasten, Auflösung 480 x 272 und 800 Variablen
- KTP700 Basic: 7"-Touchscreen mit 8 konfigurierbaren Tasten, Auflösung 800 x 480 und 800 Variablen
- KTP700 Basic DP: 7" Touchscreen mit 8 konfigurierbaren Tasten, Auflösung 800 x 480 und 800 Variablen

- KTP900 Basic: 9"-Touchscreen mit 8 konfigurierbaren Tasten, Auflösung 800 x 480 und 800 Variablen
- KTP1200 Basic: 12" Touchscreen mit 10 konfigurierbaren Tasten, Auflösung 800 x 480 und 800 Variablen
- KTP 1200 Basic DP: 12" Touchscreen mit 10 konfigurierbaren Tasten, Auflösung 800 x 400 und 800 Variablen

Neue Funktionen

Die folgenden Funktionen sind in der Version 4.5 neu hinzugekommen:

- S7-1200 OPC UA (Seite 813) Erweiterung:
 - Servermethoden-Aufrufe (Remote Procedure Calls)
 - Strukturierte Datentypen und Array-Datentypen
 - Verbesserte Diagnose
- Neue Anweisungen:
 - Die Anweisung GetSMCInfo ruft Informationen über die gesteckte SIMATIC Memory Card ab
 - Kompakte Anweisungen zum Lesen/Schreiben von Dateien: FileReadC (Seite 535), FileWriteC (Seite 538) und FileDelete (Seite 540)
- OUC-Verbindungstypen TCP, Iso-on_TCP und UDP
- Offene Benutzerkommunikation: unterstützt jetzt TCON_Settings (Seite 667)
- Webserver: Unterstützung moderner API und Zertifikathandhabung
- PROFINET unterstützt das Media Redundancy Protocol (MRP) (Seite 766) als "Client" und als "Manager" (CPU 1215C und CPU 1217C)
- Verbesserte DataLog-Funktionalität einschließlich Feld für die Synchronisierung des Zeitstempels mit der S7-1500
- Neue Übersicht der Kommunikationsprotokolle und Ports (Seite 593) für die Ethernet-Kommunikation
- Erweiterte Sicherheit:
 - Verwendung von X.509-Zertifikaten und TLS (Transport Layer Security) für sichere PG/PC- und HMI-Kommunikation (Seite 166)
 - Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161)
 - Security-Assistent (Seite 160) im TIA Portal zur Unterstützung der Konfigurierung für Secure Communication und Schutzfunktionen
 - Fortgesetzte Unterstützung der Legacy-Kommunikation ergänzend zu Secure Communication (Seite 591)
 - Erweiterte Verschlüsselung für die Passwörter der CPU-Zugriffsstufe (Seite 163) mit einer Voreinstellung für den vollständigen Schutz der CPU.
 - Möglichkeit der Nutzung einer SIMATIC Memory Card zum Festlegen oder Ändern des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten (Seite 129)
- Größerer Remanenzspeicher für S7-1200 CPUs von 10 KB bis 14 KB

- CCC-Zulassung für S7-1200 Produkte
Nicht alle S7-1200 Varianten können nach diesen Normen (Seite 1253) zertifiziert werden, und der Zertifizierungszustand kann sich ohne Ankündigung ändern. Es liegt in der Verantwortung des Anwenders, geltende Zertifizierungen anhand der auf dem Produkt angebrachten Zulassungen zu ermitteln.
- Wichtige Informationen zur Aufrechterhaltung der Betriebssicherheit Ihrer Anlage (Seite 5).

Austausch der CPU V3.0 durch eine CPU V4.x.x

Wenn Sie eine S7-1200 CPU V3.0 durch eine S7-1200 CPU V4.x.x ersetzen, beachten Sie die dokumentierten Unterschiede (Seite 1456) zwischen den Versionen und die erforderlichen Maßnahmen.

STEP 7 Programmiersoftware

STEP 7 bietet eine bedienerfreundliche Umgebung zum Entwickeln, Bearbeiten und Beobachten der Logik zur Steuerung Ihrer Anwendung. Sie bietet auch die Werkzeuge zum Konfigurieren aller Geräte in Ihrem Projekt, wie PLC- und HMI-Geräte. Damit Sie die Informationen finden, die Sie benötigen, verfügt STEP 7 über eine umfangreiche Online-Hilfe.

STEP 7 bietet Standardprogrammiersprachen, mit denen Sie das Steuerungsprogramm für Ihre Anwendung bequem und effizient entwickeln können.

- KOP (Kontaktplan) (Seite 189) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen.
- FUP (Funktionsplan) (Seite 190) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra basiert.
- SCL (Structured Control Language) (Seite 190) ist eine textbasierte, höhere Programmiersprache.

Wenn Sie einen Codebaustein anlegen, müssen Sie die Programmiersprache für den Baustein auswählen. Ihr Anwenderprogramm kann mit Codebausteinen arbeiten, die in einer dieser Programmiersprachen angelegt wurden.

Hinweis

STEP 7 ist die Softwarekomponente für Programmierung und Konfiguration im TIA-Portal. Das TIA-Portal umfasst neben STEP 7 auch WinCC zum Entwerfen und Ausführen von Runtime-Prozessvisualisierung, und es bietet Online-Hilfe für WinCC ebenso wie für STEP 7.

Für die neuen Funktionen in S7-1200 V4.5 ist STEP 7 Professional V17 erforderlich.

3.1 Systemvoraussetzungen

Sie müssen STEP 7 mit Administratorrechten installieren.

Tabelle 3-1 Systemvoraussetzungen

Hardware/Software	Voraussetzungen
Prozessortyp	Intel® Core™ i3-6100U, 2,30 GHz oder mehr
RAM	8 GB
Freier Festplattenspeicher	20 GB auf Systemlaufwerk C:\

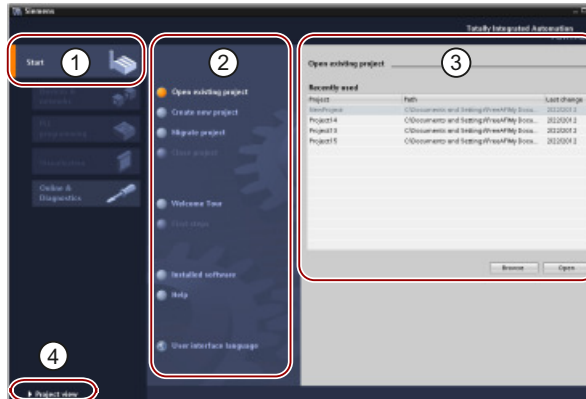
Hardware/Software	Voraussetzungen
Betriebssysteme	<p>Sie können STEP 7 mit den folgenden Betriebssystemen verwenden:</p> <ul style="list-style-type: none"> • Windows 7 (64 Bit): <ul style="list-style-type: none"> – Windows 7 Home Premium SP1 ** – Windows 7 Professional SP1 – Windows 7 Enterprise SP1 – Windows 7 Ultimate SP1 • Windows 10 (64 Bit): <ul style="list-style-type: none"> – 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 LTSC – Windows 10 IoT Enterprise 2015 LTSC – Windows 10 IoT Enterprise 2016 LTSC • Windows Server (64 Bit) <ul style="list-style-type: none"> – Windows Server 2012 R2 StdE (vollständige Installation) – Windows Server 2016 Standard (vollständige Installation)
Grafikkarte	32 MB RAM-Speicher 24-Bit-Farbtiefe
Bildschirmauflösung	1024 x 768
Netzwerk	100 MBit/s Ethernet oder schneller für die Kommunikation zwischen STEP 7 und der CPU

* Einschließlich aller geltenden Sicherheits-Updates. * Ausführlichere Informationen zu Betriebssystemen finden Sie in der Hilfe zu Microsoft Windows oder der Microsoft-Website.

** Nur für Basic-Versionen

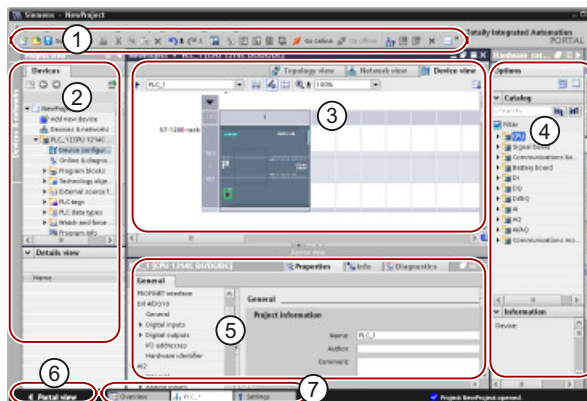
3.2 Einfaches Arbeiten mit unterschiedlichen Ansichten

STEP 7 stellt eine benutzerfreundliche Umgebung bereit, in der Sie die Steuerungslogik entwickeln, die HMI-Visualisierung konfigurieren und die Netzwerkkommunikation einrichten können. Zur Steigerung Ihrer Produktivität bietet STEP 7 zwei unterschiedliche Ansichten des Projekts: eine tätigkeitsorientierte Anzahl von Portalen für die einzelnen Funktionen (Portalansicht) und eine projektorientierte Ansicht der Elemente im Projekt (Projektansicht). Sie entscheiden, in welcher Ansicht Sie am effizientesten arbeiten können. Per Mausklick können Sie zwischen der Portalansicht und der Projektansicht wechseln.



Portalansicht

- ① Portale für die verschiedenen Aufgaben
- ② Aufgaben für das ausgewählte Portal
- ③ Auswahlpanel zur gewählten Aktion
- ④ Änderungen an der Projektansicht



Projektansicht

- ① Menüs und Funktionsleiste
- ② Projektnavigator
- ③ Arbeitsbereich
- ④ Taskcards
- ⑤ Inspektorfenster
- ⑥ Änderungen an der Portalansicht
- ⑦ Editorleiste

Da sich alle Komponenten an einer Stelle befinden, haben Sie schnellen Zugriff auf jeden Bereich Ihres Projekts. Der Arbeitsbereich besteht aus drei Ansichten in Registern:

- Gerätesicht: Zeigt das Gerät an, das Sie hinzugefügt oder ausgewählt haben, und dessen zugehörige Module
- Netzsicht: Zeigt die CPUs und Netzwerkverbindungen in Ihrem Netzwerk an
- Topologiesicht: Zeigt die PROFINET-Topologie des Netzwerks einschließlich Geräten, passiven Komponenten, Ports, Verschaltungen und Portdiagnose an

In jeder Ansicht können Sie auch Konfigurationsaufgaben durchführen. Das Inspektorfenster zeigt die Eigenschaften und weitere Informationen für das im Arbeitsbereich ausgewählte Objekt an. Für die verschiedenen von Ihnen gewählten Objekte zeigt das Inspektorfenster

jeweils die konfigurierbaren Eigenschaften. Das Inspektorfenster verfügt außerdem über Register, unter denen Diagnoseinformation und weitere Meldungen angezeigt werden.

In der Editorleiste werden alle derzeit geöffneten Editoren angezeigt. Mit der Editorleiste arbeiten Sie so schneller und effizienter. Zum Umschalten zwischen geöffneten Editoren klicken Sie einfach auf den gewünschten Editor. Sie können auch zwei Editoren gleichzeitig anzeigen und diese vertikal oder horizontal anordnen. Dadurch sind "Drag&Drop"-Operationen zwischen Editoren möglich.

Das STEP 7 Informationssystem bietet eine umfangreiche Online-Hilfe für alle Konfigurations-, Programmier- und Überwachungswerkzeuge von STEP 7. Dort finden Sie über dieses Handbuch hinausgehende ausführliche Erklärungen.

3.3 Kompatibilität zwischen STEP 7 und der S7-1200

STEP 7 V17 unterstützt die Konfiguration und Programmierung der S7-1200 V4.5 CPU.

Sie können Projekte für frühere Versionen von S7-1200 CPUs V4.x aus STEP 7 V13 oder höher in eine S7-1200 CPU V4.5 laden. Ihre Konfiguration und das Programm sind begrenzt auf den Umfang der Funktionen und Anweisungen, der von der Vorgängerversion der S7-1200 CPU und Ihrer Version von STEP 7 unterstützt wurde.

Dank dieser Rückwärtskompatibilität können Sie Programme, die Sie zuvor für Vorgängerversionen entwickelt und programmiert hatten, auf Varianten der S7-1200 CPU V4.5 weiterhin ausführen.

WARNUNG

Risiken beim Kopieren und Einfügen von Programmlogik aus älteren Versionen von STEP 7

Kopieren von Programmlogik aus einer Vorgängerversion von STEP 7 kann unvorhersehbares Verhalten bei der Programmausführung oder Fehler beim Übersetzen verursachen. In den verschiedenen Versionen von STEP 7 werden Programmelemente unterschiedlich implementiert. Beim Übersetzen werden die Unterschiede nicht immer erkannt, wenn Sie die Änderungen durch Einfügen aus einer Vorgängerversion in STEP 7 V15 vorgenommen haben. Die Ausführung unberechenbarer Programmlogik kann schwere oder lebensgefährliche Verletzungen verursachen, wenn Sie das Programm nicht korrigieren.

Aktualisieren Sie bei der Verwendung von Programmlogik aus einer Vorgängerversion von STEP 7 immer das gesamte Projekt auf die neueste Version von STEP 7. Sie können Programmlogik nach Bedarf kopieren, ausschneiden, einfügen und bearbeiten. In STEP 7 V16 oder höher können Sie ein Projekt aus STEP 7 V13 oder höher öffnen. STEP 7 führt anschließend die notwendigen Kompatibilitätsumwandlungen durch und aktualisiert das Programm korrekt. Solche Umwandlungen und Korrekturen im Zusammenhang mit der Aktualisierung sind für die ordnungsgemäße Übersetzung und Ausführung des Programms notwendig. Wenn Ihr Projekt älter als STEP 7 V13 ist, müssen Sie das Projekt schrittweise auf STEP 7 V17 aktualisieren (Seite 1455).

Sie können keine Projekte einer S7-1200 CPU V1.0, V2.0 oder V3.0 in eine S7-1200 CPU V4.x laden. Unter Ersatz einer CPU V3.0 durch eine CPU V4.x (Seite 1456) finden Sie Richtlinien zum Aktualisieren älterer Projekte auf ein Projekt, das Sie in das Zielsystem laden können.

Hinweis**Projekte mit Varianten der S7-1200 CPU V1.x**

Sie können ein STEP 7-Projekt, das S7-1200 CPUs V1.x enthält, in STEP 7 V15.1 nicht öffnen. Um Ihr bestehendes Projekt verwenden zu können, müssen Sie Ihr Projekt in STEP 7 V13 (mit jeder beliebigen Aktualisierung) öffnen und die S7-1200 CPUs V1.x in V2.0 oder höher konvertieren. Dann können Sie das gespeicherte Projekt mit den konvertierten CPUs in STEP 7 V15.1 öffnen.

Kompatibilität im Hinblick auf Secure Communication

Die Funktionen für Secure Communication (Seite 160) ab V4.5 der S7-1200 CPU benötigen STEP 7 V17 oder höher.

Es kann die Situation geben, dass Sie S7-1200 CPUs der Version 4.5 nutzen, jedoch nicht mit STEP 7 V17.

Einzelheiten zur Kommunikation zwischen einer S7-1200 CPU V4.5, Clients und HMI-Geräten finden Sie unter Sichere PG/PC-Kommunikation und HMI-Kommunikation aktivieren und Zertifikate erstellen (Seite 166).

Siehe auch

Neue Funktionen (Seite 35)

Einbau

4.1 Richtlinien für den Einbau von S71200 Geräten

Die S7-1200 Geräte wurden so ausgelegt, dass sie einfach einzubauen sind. Sie können eine S7-1200 entweder in einer Schalttafel oder auf einer Standard-Hutschiene einbauen; die S7-1200 kann horizontal oder vertikal eingebaut werden. Die kompakte Größe der S7-1200 macht eine effiziente Platzausnutzung möglich.


Das SIMATIC S7-1200 System ist anhand von Normen für elektrische Geräte als offenes Betriebsmittel klassifiziert. Sie müssen die S7-1200 in einem Gehäuse, Schaltschrank oder in einer Schaltzentrale einbauen. Nur berechtigtes Personal darf Zugang zum Gehäuse, Schaltschrank oder der Schaltzentrale haben.

Bei der Installation ist eine trockene Umgebung für die S7-1200 vorzusehen. SELV/PELV-Stromkreise sollen in trockenen Umgebungen Schutz vor elektrischem Schlag bieten.

Die Installation muss die für offene Betriebsmittel in Ihrer spezifischen Standortkategorie geforderte mechanische Festigkeit, Brandklasse und Stabilität nach den geltenden elektrotechnischen und baurechtlichen Vorschriften bieten.

Verschmutzung leitfähiger Teile durch Staub, Feuchtigkeit und Luftverschmutzung kann zu Betriebsfehlern und elektrischen Fehlern in der PLC führen.

Wenn sich der PLC in einem Bereich befindet, in dem Verschmutzung von leitfähigen Teilen auftreten kann, muss der PLC durch ein Gehäuse mit entsprechender Schutzklasse geschützt werden. IP54 ist eine Schutzklasse, die im Allgemeinen für elektronische Anlagen in stark verunreinigten Umgebungen verwendet wird und möglicherweise für Ihre Anwendung geeignet ist.

	WARNUNG
<p>Nicht ordnungsgemäße Installation der S7-1200 kann zu elektrischen Fehlern oder unerwartetem Betrieb der Maschine führen.</p> <p>Elektrische Fehler oder unerwarteter Betrieb der Maschine können zu schweren oder lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p> <p>Alle Anweisungen für die Installation und Wartung einer ordnungsgemäßen Betriebsumgebung sind zu befolgen, um einen sicheren Betrieb der Geräte zu gewährleisten.</p>	

Halten Sie die S7-1200 Geräte fern von Wärme, Hochspannung und elektrischen Störungen

Als allgemeine Regel für die Anordnung von Geräten in Ihrem System gilt, dass Sie Geräte, die Hochspannung oder hohe elektrische Störungen erzeugen, von den elektronischen Niederspannungsgeräten wie der S7-1200 fernhalten.


Wenn Sie das Layout der S71200 in Ihrer Schalttafel planen, berücksichtigen Sie wärmeerzeugende Geräte und ordnen Sie die elektronischen Geräte in den kühleren Bereichen Ihres Schaltschranks an. Je weniger Sie ein elektronisches Gerät Umgebungen mit hohen Temperaturen aussetzen, desto länger ist die Betriebsdauer des Geräts.

4.1 Richtlinien für den Einbau von S71200 Geräten

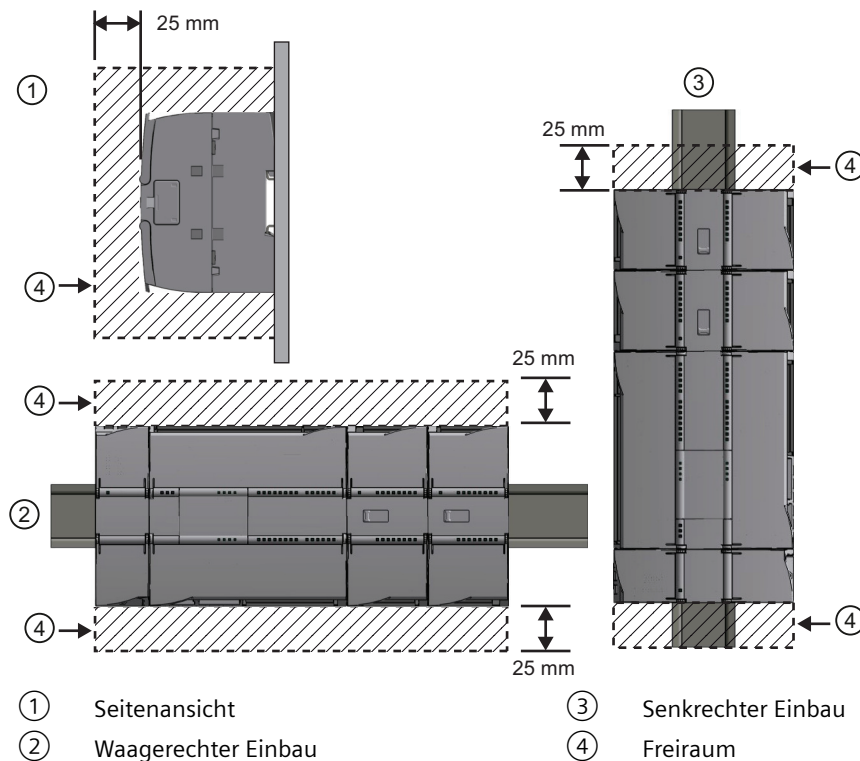
Berücksichtigen Sie auch, wie Sie die Verdrahtung der Geräte in der Schalttafel verlegen. Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in derselben Kabeltrasse wie AC-Versorgungsleitungen und schnellschaltende DC-Starkstromleitungen zu verlegen.

Lassen Sie genügend Abstand für Kühlung und Verdrahtung

Die S71200 Geräte sind für natürliche Wärmeabfuhr durch Konvektion ausgelegt. Lassen Sie deshalb oberhalb und unterhalb der Geräte jeweils mindestens 25 mm Platz, um die Wärmeabfuhr zu gewährleisten. Achten Sie ferner darauf, dass zwischen der Modulfront und der Innenseite des Gehäuses eine Tiefe von mindestens 25 mm bleibt.

	VORSICHT
<p>Bei vertikalem Einbau ist die maximal zulässige Umgebungstemperatur um 10 °C niedriger.</p>	
<p>Richten Sie ein vertikal eingebautes S71200 System wie in der folgenden Abbildung gezeigt aus.</p>	
<p>Stellen Sie sicher, dass das S7-1200 System richtig montiert wird.</p>	

Wenn Sie das Layout für Ihr S71200 System planen, lassen Sie genügend Abstand für die Verdrahtung und die Kommunikationskabelanschlüsse.




4.2 Leistungsbilanz

Ihre CPU besitzt eine interne Spannungsversorgung, die neben der CPU die Signalmodule, Signalboards, Kommunikationsmodule und andere 24-V-DC-Verbraucher speist.


In den Technischen Daten (Seite 1253) finden Sie Informationen zur 5-V-DC-Leistungsbilanz Ihrer CPU und zum 5-V-DC-Leistungsbedarf der Signalmodule, Signalboards und Kommunikationsmodule. Mit Hilfe der Informationen unter "Berechnen der Leistungsbilanz" (Seite 1437) können Sie berechnen, wieviel Leistung (bzw. Strom) die CPU für Ihre Konfiguration liefern kann.

Die CPU verfügt auch über eine 24-V-DC-Geberversorgung, die 24 V DC für Eingänge, für die Relaispulen der Signalmodule und für sonstige Verbraucher liefert. Wenn der 24-V-DC-Leistungsbedarf die Leistung der Geberversorgung übersteigt, müssen Sie eine externe 24-V-DC-Spannungsversorgung an Ihr System anschließen. In den Technischen Daten (Seite 1253) finden Sie die Leistungsbilanz für die 24-V-DC-Geberversorgung für Ihre jeweilige CPU.

Wenn Sie eine externe 24-V-DC-Spannungsversorgung benötigen, müssen Sie darauf achten, dass die Spannungsversorgung nicht parallel zur Geberversorgung der CPU angeschlossen ist. Die beste Störfestigkeit erreichen Sie, wenn Sie die Leitungen (M) der jeweiligen Spannungsversorgungen miteinander verbinden.

	WARNUNG
<p>Wenn Sie parallel zur 24-V-DC-Geberversorgung eine externe 24-V-DC-Spannungsquelle anschließen, kann es sein, dass die beiden Spannungsquellen sich beim Aufbauen der geeigneten Ausgangsspannung gegenseitig beeinträchtigen.</p> <p>Als Folge kann sich die Lebensdauer verkürzen bzw. eine oder beide Spannungsquellen können sofort ausfallen. Unvorhersehbarer Betrieb kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p> <p>Die DC-Geberversorgung und eine externe Spannungsquelle müssen die Spannung an unterschiedlichen Punkten liefern.</p>	

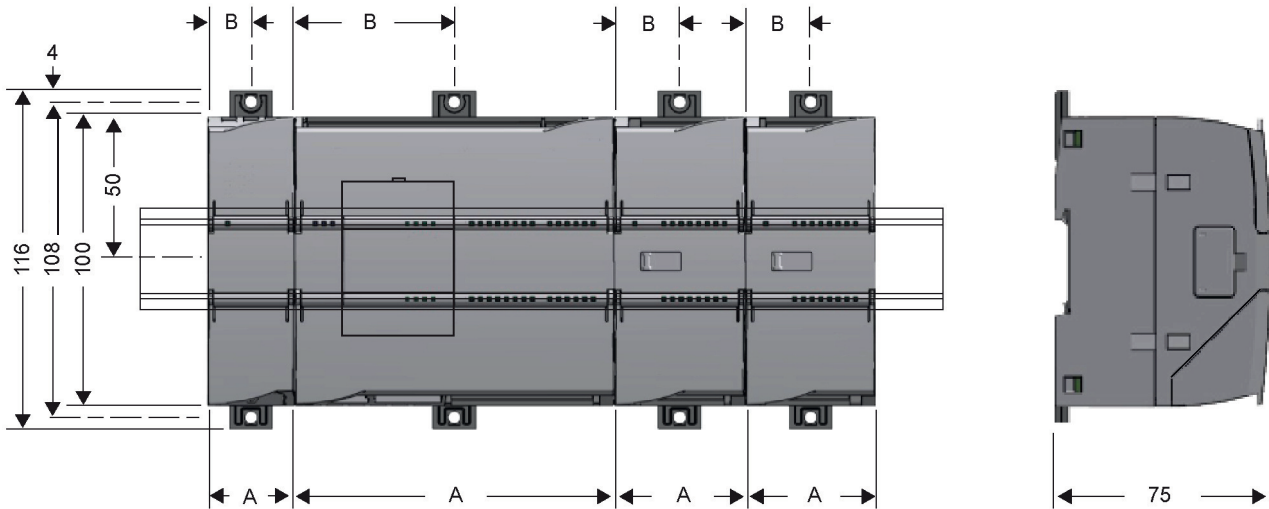
Einige der 24-V-DC-Eingangsports des S7-1200 Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Die folgenden Stromkreise sind beispielsweise miteinander verbunden, sofern sie in den Datenblättern als "nicht potentialgetrennt" angegeben sind: die 24-V-DC-Versorgung der CPU, der Leistungseingang für die Relaispule eines SM oder die Versorgung eines nicht potentialgetrennten Analogeingangs. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

	WARNUNG
<p>Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betrieb des Zielsystems und angeschlossener Geräte führen kann.</p> <p>Die Nichteinhaltung dieser Richtlinien kann Schäden oder unvorhersehbaren Betrieb verursachen, was zu lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.</p> <p>Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem S7-1200 System an dasselbe Bezugspotential an.</p>	

4.3 Vorgehensweisen zum Einbau und Ausbau

4.3.1 Befestigungsmaße für S7-1200 Geräte

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



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

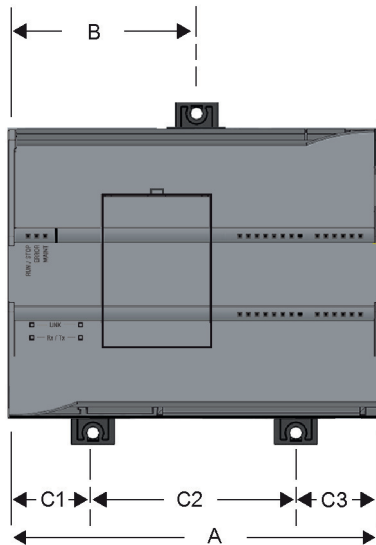


Tabelle 4-1 Abmessungen für die Montage (mm)

S71200 Geräte		Breite A (mm)	Breite B (mm)	Breite C (mm)
CPU	CPU 1211C und CPU 1212C	90	45	--
	CPU 1214C	110	55	--
	CPU 1215C	130	65 (oben)	Unten: C1: 32,5 C2: 65 C3: 32,5
	CPU 1217C	150	75	Unten: C1: 37,5 C2: 75 C3: 37,5
Signalmodule	8 oder 16 digitale E/A 2, 4 oder 8 analoge E/A Thermoelement, 4 oder 8 E/A RTD, 4 E/A SM 1278 IO Link-Master	45	22,5	--
	8 digitale Ausgänge x Relais (Umschaltung)	70	35	--
	16 analoge E/A RTD, 8 E/A	70	35	--
	SM 1238 Energy Meter	45	22,5	--
Kommunikationschnittstellen	CM 1241 RS232 und CM 1241 RS422/485 CM 1243-5 PROFIBUS-Master und CM 1242-5 PROFIBUS-Slave CM 1242-2 AS-i-Master CP 1242-7 GPRS V2 CP 1243-7 LTE-US CP 1243-7 LTE-EU CP 1243-1 CP 1243-8 IRC RF120C	30	15	--
	TS (TeleService) Adapter IE Advanced ¹ TS (TeleService)-Adapter IE Basic ¹ TS-Adapter TS-Modul	30 30	15 15	-- --

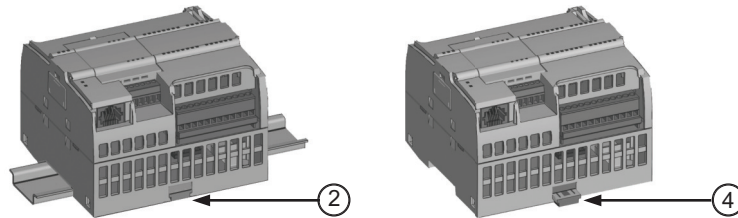
¹ Bevor Sie den TS (TeleService)-Adapter IE Advanced oder IE Basic einbauen, müssen Sie zunächst den TS-Adapter und ein TS-Modul anschließen. Die Gesamtbreite ("Breite A") beträgt 60 mm.

Alle CPUs, SMs, CMs und CPs können auf der DIN-Schiene oder im Schaltschrank montiert werden. Verwenden Sie die Hutschielenklemmen für die Befestigung des Geräts auf der Hutschiene. Diese Klemmen rasten auch in einer ausgezogenen Position ein, um den Einbau des Geräts in einer Schalttafel zu ermöglichen. Das Innenmaß der Bohrung für die Hutschielenklemmen am Gerät beträgt 4,3 mm.

Ober- und unterhalb des Geräts muss ein Freiraum von 25 mm für die Luftzirkulation als Schutz vor Überhitzung eingehalten werden.

Einbau und Ausbau der S71200 Geräte

Die CPU kann auf einfache Weise auf einer Standard-Hutschiene oder in einer Schalttafel eingebaut werden. Für die Befestigung des Geräts auf der Hutschiene werden passende Hutschienenklemmen mitgeliefert. Diese Klemmen können auch in einer ausgezogenen Position einrasten und ermöglichen dann das Festschrauben des Geräts in einer Schalttafel.



- | | |
|---|--|
| ① | ③ |
| ① Hutschienenmontage | ③ Schalttafelmontage |
| ② Standard-Hutschienenklemme in verriegelter Position | ④ Klemme in ausgezogener Position für die Schalttafelmontage |

Bevor Sie ein elektrisches Gerät einbauen oder ausbauen, müssen Sie sicherstellen, dass die Spannungsversorgung der Geräte ausgeschaltet ist. Achten Sie außerdem darauf, dass auch alle angeschlossenen Geräte ausgeschaltet sind.

⚠ WARNUNG

Wenn Sie die S7-1200 oder daran angeschlossene Geräte in eingeschaltetem Zustand ein- oder ausbauen, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte unerwartet arbeiten.

Failure to disable all power to the S7-1200 and related equipment during installation or removal procedures could result in death, severe personal injury and/or property damage due to electric shock or unexpected equipment operation.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S71200 CPUs abgeschaltet ist.

Achten Sie immer darauf, dass Sie das richtige Modul bzw. das richtige Gerät verwenden, wenn Sie ein S7-1200 Gerät einbauen bzw. auswechseln.

 **WARNUNG**

Falscher Einbau eines S7-1200 Moduls kann zu unvorhersehbarer Funktionsweise des Programms der S7-1200 führen.

Failure to replace an S7-1200 device with the same model, orientation, or order could result in death, severe personal injury and/or property damage due to unexpected equipment operation.

Wechseln Sie ein S7-1200 Gerät immer mit der gleichen Ausführung aus, richten Sie das Gerät korrekt aus und bauen Sie es an der richtigen Stelle ein.

 **WARNUNG**

Trennen Sie keine Geräte, wenn eine entflammbare oder brennbare Atmosphäre vorliegt.

Das Trennen von Geräten in einer entflammbaren oder brennbaren Atmosphäre kann ein Feuer oder eine Explosion verursachen, was zu Tod, schweren Verletzungen und/oder Sachschaden führen kann.

Achten Sie immer auf angemessene Sicherheitsmaßnahmen, wenn eine entflammbare oder brennbare Atmosphäre vorliegt.

Hinweis

Elektrostatische Entladungen können das Gerät oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit dem Gerät arbeiten.

4.3.2 Einbau und Ausbau der CPU

Sie können die CPU in einer Schalttafel oder auf einer Standard-Hutschiene einbauen.

Hinweis

Schließen Sie die Kommunikationsmodule an die CPU an und bauen Sie alle Module gemeinsam ein. Installieren Sie die Signalmodule erst nach dem Einbau der CPU.

Beachten Sie beim Einbau der Geräte auf der Hutschiene oder in einer Schalttafel Folgendes:

- Achten Sie beim Einbau in eine Hutschiene darauf, dass sich die obere Hutschieneklemme der CPU und der angeschlossenen CMs in der eingerasteten (inneren) Stellung und die untere Hutschieneklemme in der ausgezogenen Stellung befindet.
- Bringen Sie nach dem Einbau der Geräte auf der Hutschiene die unteren Hutschieneklemmen in die eingerastete Stellung, um die Geräte auf der Hutschiene zu befestigen.
- Achten Sie bei der Schalttafelmontage darauf, dass sich die Hutschieneklemmen in der ausgezogenen Stellung befinden.

Um die CPU in eine Schalttafel einzubauen, gehen Sie folgendermaßen vor:

1. Bringen Sie die Bohrungen (M4) mit den in den Abmessungen in der Tabelle Abmessungen für die Montage (mm) (Seite 46) gezeigten Vorgaben an.
2. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind.
3. Ziehen Sie die Montageklemmen aus dem Modul heraus. Achten Sie darauf, dass sich die Hutschieneklemmen oben und unten an der CPU in der ausgezogenen Position befinden.
4. Schrauben Sie das Modul mit einer Zylinderkopfschraube M4 sowie einem Federring und einer flachen Unterlegscheibe fest. Verwenden Sie keine Senkkopfschraube.

Hinweis

Welcher Schraubentyp erforderlich ist, hängt von der Art des Materials ab, auf dem das Modul montiert wird. Ziehen Sie die Schraube mit dem entsprechenden Drehmoment fest, bis der Federring flachgedrückt ist. Ziehen Sie die Schrauben nicht mit übermäßigem Drehmoment fest. Verwenden Sie keine Senkkopfschrauben.

Hinweis

In Umgebungen, in denen starke Schwingungen auftreten, oder bei vertikalem Einbau der CPU kann es nützlich sein, die CPU mit DIN-Schieneklemmen auf der Hutschiene zu sichern. Montieren Sie einen Endhalter (8WA1808 oder 8WA1805) auf der DIN-Schiene, um sicherzustellen, dass die Module miteinander verbunden bleiben. In Umgebungen, in denen starke Schwingungen auftreten, bietet die Schalttafelmontage der CPU einen besseren Schutz vor Schwingungen.

Tabelle 4-2 Einbau der CPU auf einer DIN-Schiene

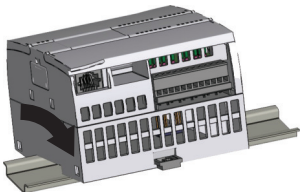
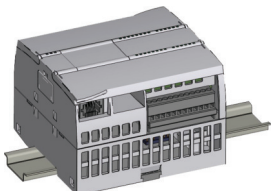
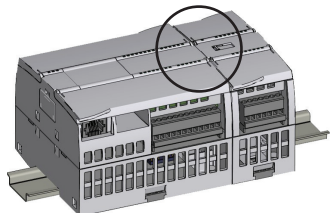
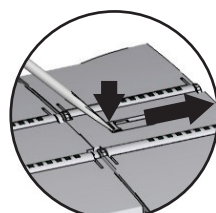
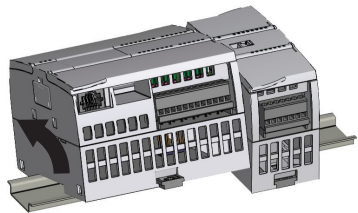
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Montieren Sie die Hutschiene. Verschrauben Sie die Hutschiene in Abständen von jeweils 75 mm mit der Schalttafel. 2. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 3. Hängen Sie die CPU oben an der Hutschiene ein. 4. Ziehen Sie die Hutschieneklemme auf der Unterseite der CPU heraus, damit die CPU über die Schiene passt. 5. Schwenken Sie die CPU nach unten in ihre Einbauposition auf der Schiene. 6. Drücken Sie die Klemmen herunter, so dass die CPU fest auf der Schiene sitzt.
	

Tabelle 4-3 Ausbau der CPU auf einer DIN-Schiene

Aufgabenstellung	Vorgehensweise
 	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Trennen Sie die E/A-Steckverbinder, die Verdrahtung und Kabel von der CPU (Seite 57). 3. Bauen Sie die CPU und die zugehörigen Kommunikationsmodule gemeinsam aus. Alle Signalmodule müssen eingebaut bleiben. 4. Wenn ein SM an die CPU angeschlossen ist, ziehen Sie den Busstecker zurück: <ul style="list-style-type: none"> – Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an. – Drücken Sie nach unten, um den Klemmenblock von der CPU zu lösen. – Schieben Sie die Lasche ganz nach rechts. 5. Nehmen Sie die CPU ab: <ul style="list-style-type: none"> – Ziehen Sie die Hutschieneklemme heraus, damit die CPU von der Schiene gelöst wird. – Schwenken Sie die CPU nach oben von der Schiene herunter und nehmen Sie die CPU aus dem System heraus.
	

4.3.3 Ein- und Ausbau eines SBs, CBs oder BBs

Tabelle 4-4 Einbau eines SBs, CBs oder BBs 1297

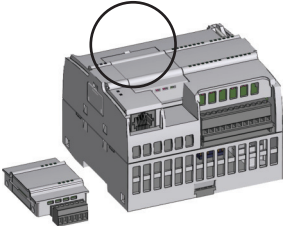
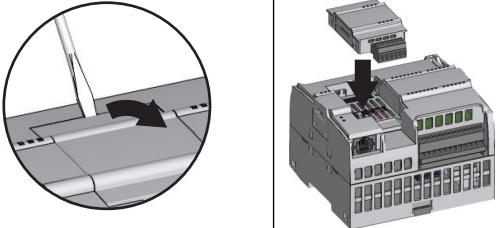
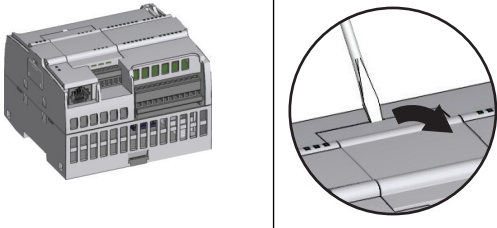
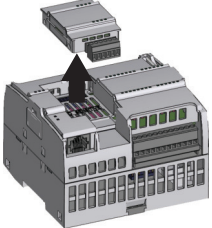
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Nehmen Sie die obere und untere Abdeckung des Klemmenblocks von der CPU ab. 3. Führen Sie einen Schraubendreher in den Schlitz oben auf der CPU an der Hinterseite der Abdeckung ein. 4. Hebeln Sie die Abdeckung vorsichtig gerade nach oben aus ihrer Halterung und nehmen Sie sie von der CPU ab.
	<ol style="list-style-type: none"> 5. Setzen Sie das Modul gerade von oben in seine Montageposition in der CPU ein. 6. Drücken Sie das Modul fest herunter, bis es einrastet. 7. Setzen Sie die Klemmenabdeckungen wieder ein.

Tabelle 4-5 Ausbau eines SBs, CBs oder BBs 1297

Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Nehmen Sie die obere und untere Abdeckung des Klemmenblocks von der CPU ab. 3. Nehmen Sie den Steckverbinder des Signalboards ab (sofern vorhanden), indem Sie ihn mit einem Schraubendreher leicht wegdrücken.
	<ol style="list-style-type: none"> 4. Setzen Sie einen Schraubendreher in den Schlitz auf der Oberseite des Moduls ein. 5. Hebeln Sie das Modul vorsichtig aus seiner Halterung und nehmen Sie es von der CPU ab. 6. Nehmen Sie das Modul ohne Zuhilfenahme eines Schraubendrehers gerade von oben aus seiner Montageposition aus der CPU heraus. 7. Bringen Sie die Abdeckung der CPU wieder an. 8. Setzen Sie die Klemmenabdeckungen wieder ein.

Einsetzen oder Austauschen der Batterie des Batterieboards BB 1297

Das BB 1297 benötigt eine Batterie des Typs CR1025. Die Batterie ist nicht im Lieferumfang des BB 1297 enthalten und muss separat erworben werden. Zum Einbauen oder Austauschen der Batterie gehen Sie folgendermaßen vor:

1. Setzen Sie eine neue Batterie im BB 1297 mit der positiven Seite der Batterie nach oben und der negativen Seite neben der gedruckten Beschaltungsplatte ein.
2. Das BB 1297 kann jetzt in die CPU eingesetzt werden. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind, und gehen Sie zur Installation des BB 1297 wie oben beschrieben vor.

So ersetzen Sie die Batterie des BBs 1297:

1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. Bauen Sie das BB 1297 wie oben beschrieben aus der CPU aus.
2. Entnehmen Sie die alte Batterie vorsichtig mit Hilfe eines kleinen Schraubendrehers. Schieben Sie die Batterie unter der Klemme heraus.
3. Setzen Sie eine neue Batterie CR1025 mit der positiven Seite der Batterie nach oben und der negativen Seite neben der gedruckten Beschaltungsplatte ein.
4. Bauen Sie das Batterieboard BB 1297 wieder wie oben beschrieben ein.



WARNUNG

Die Installation einer nicht spezifizierten Batterie im BB 1297 oder der Anschluss einer nicht spezifizierten Batterie an den Schaltkreis auf andere Weise kann zu Brand oder Bauteilbeschädigung und unvorhersehbarem Betrieb von Maschinen führen.

Brand oder unvorhersehbarer Betrieb von Maschinen können lebensgefährliche Verletzungen oder Sachschäden verursachen.

Verwenden Sie für die Pufferung der Echtzeituhr nur die spezifizierte Batterie CR1025.

4.3.4 Einbau und Ausbau eines SMs

Tabelle 4-6 Einbau eines SMs

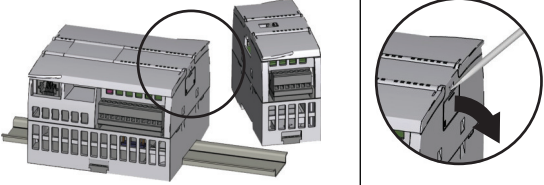
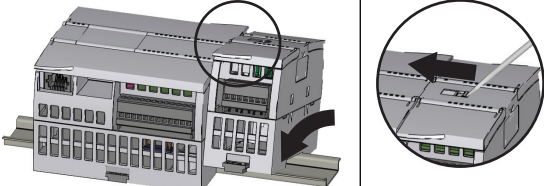
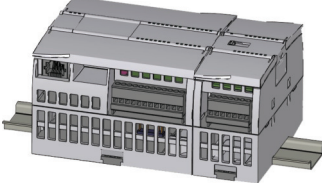
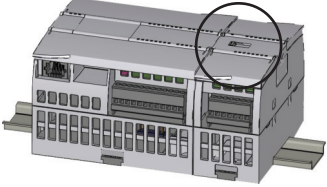
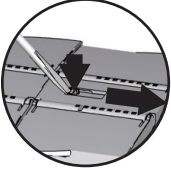
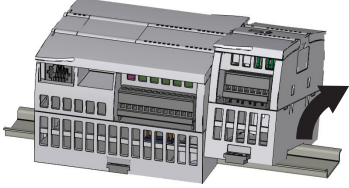
Aufgabenstellung	Vorgehensweise
	<p>Bauen Sie Ihr SM nach der Montage der CPU ein.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Nehmen Sie die Abdeckung für den Anschluss an der rechten Seite der CPU ab: <ul style="list-style-type: none"> – Führen Sie einen Schraubendreher in den Schlitz über der Abdeckung ein. – Hebeln Sie die Abdeckung vorsichtig oben aus ihrer Halterung und nehmen Sie sie ab. 3. Bewahren Sie die Abdeckung zur späteren Verwendung auf.
	<p>SM an die CPU anschließen:</p> <ol style="list-style-type: none"> 1. Positionieren Sie das SM neben der CPU. 2. Hängen Sie das SM oben an der Hutschiene ein. 3. Ziehen Sie die untere Hutschieneklemme heraus, damit das SM über die Schiene passt. 4. Schwenken Sie das SM nach unten in die Position neben der CPU und drücken Sie die untere Klemme herunter, so dass das SM fest auf der Hutschiene sitzt.
	<p>Der Busstecker stellt dadurch die mechanische und elektrische Verbindung für das SM her.</p> <ol style="list-style-type: none"> 1. Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des SM an. 2. Schieben Sie die Lasche ganz nach links, um den Busstecker in die CPU zu schieben. <p>Gehen Sie genauso vor, um ein Signalmodul an einem Signalmodul einzubauen.</p>

Tabelle 4-7 Ausbau eines SMs

Aufgabenstellung	Vorgehensweise
	<p>Sie können jedes SM ausbauen, ohne die CPU oder andere SMs ausbauen zu müssen.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Ziehen Sie die E/A-Steckverbinder und die Verdrahtung vom SM (Seite 57) ab. 3. Ziehen Sie den Busstecker zurück. <ul style="list-style-type: none"> – Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des SM an. – Drücken Sie nach unten, um den Klemmenblock von der CPU zu lösen. – Schieben Sie die Lasche ganz nach rechts. <p>Ist auf der rechten Seite ein weiteres SM eingebaut, wiederholen Sie diese Vorgehensweise für das SM.</p>
	
	<p>Nehmen Sie das SM ab:</p> <ol style="list-style-type: none"> 1. Ziehen Sie die untere Hutschienenklemme heraus, damit das SM von der Schiene gelöst wird. 2. Schwenken Sie das SM nach oben und von der Schiene herunter. Nehmen Sie das SM aus dem System heraus. 3. Bringen Sie ggf. eine Abdeckung am Busstecker der CPU an, um Verschmutzung zu vermeiden. <p>Gehen Sie genauso vor, um ein Signalmodul von einem Signalmodul auszubauen.</p>

4.3.5 Einbau und Ausbau eines CMs oder CPs

Schließen Sie die Kommunikationsmodule an die CPU an und bauen Sie alle Module gemeinsam ein. Dies wird unter Einbau und Ausbau der CPU (Seite 49) gezeigt.

Tabelle 4-8 Einbau eines CMs oder CPs

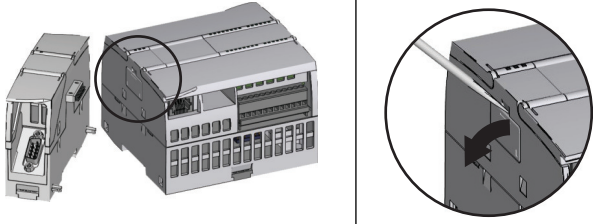
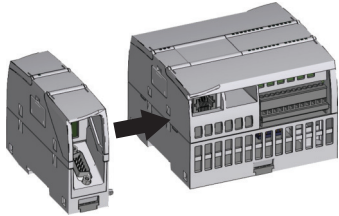
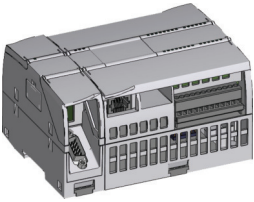
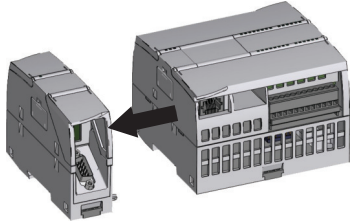
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Schließen Sie das CM an die CPU an und bauen Sie alle Module gemeinsam auf der Hutschiene oder in der Schalttafel ein. 3. Nehmen Sie die Busabdeckung an der linken Seite der CPU ab: <ul style="list-style-type: none"> – Führen Sie einen Schraubendreher in den Schlitz über der Busabdeckung ein. – Hebeln Sie die Abdeckung vorsichtig oben aus ihrer Halterung. 4. Nehmen Sie die Busabdeckung ab. Bewahren Sie die Abdeckung zur späteren Verwendung auf. 5. CM oder CP an die CPU anschließen: <ul style="list-style-type: none"> – Richten Sie den Busstecker und die Stifte des CMs zu den Bohrungen in der CPU aus. – Drücken Sie die Bauteile fest zusammen, bis die Stifte einrasten. 6. Bauen Sie die CPU und den CP auf einer DIN-Schiene oder in eine Schalttafel ein.
	

Tabelle 4-9 Ausbau eines CMs oder CPs

Aufgabenstellung	Vorgehensweise
	<p>Bauen Sie die CPU und das CM gemeinsam aus der Standard-Hutschiene bzw. der Schattafel aus.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Ziehen Sie die E/A-Steckverbinder, die Verdrahtung und die Kabel von der CPU und den CMs ab. 3. Bringen Sie bei der Hutschienenmontage die unteren Hutschienenklemmen an der CPU und den CMs in die ausgezogene Stellung. 4. Bauen Sie die CPU und die CMs aus der Standard-Hutschiene bzw. der Schattafel aus. 5. Halten Sie die CPU und die CMs fest und ziehen Sie sie auseinander.
	

ACHTUNG

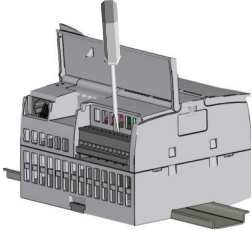
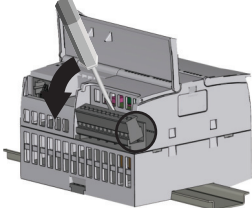
Trennen Sie die Module ohne Einsatz eines Werkzeugs.

Verwenden Sie zum Trennen der Module kein Werkzeug, da sonst die Teile beschädigt werden können.

4.3.6 Ausbau und Einbau des S7-1200 Klemmenblocks

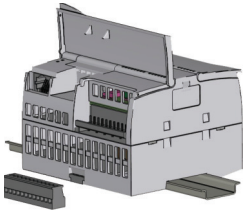
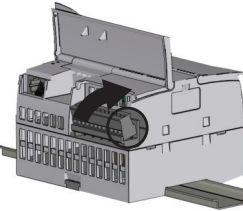
Die CPU, das SB und SM verfügen über abnehmbare Steckverbinder, um die Verdrahtung zu vereinfachen.

Tabelle 4-10 Ausbau des Klemmenblocks

Aufgabenstellung	Vorgehensweise
	<p>Bereiten Sie das System für den Ausbau des Klemmenblocks vor, indem Sie die Spannung der CPU ausschalten und die Abdeckung oberhalb des Klemmenblocks öffnen.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Suchen Sie auf der Oberseite des Klemmenblocks den Schlitz für die Spitze des Schraubendrehers. 3. Führen Sie einen Schraubendreher in den Schlitz ein.
	<ol style="list-style-type: none"> 4. Hebeln Sie die Oberseite des Klemmenblocks vorsichtig aus der Halterung in der CPU. Es ist ein Schnappgeräusch zu hören, wenn sich der Klemmenblock löst. 5. Fassen Sie den Klemmenblock mit der Hand und ziehen Sie ihn aus der CPU heraus.

4.3 Vorgehensweisen zum Einbau und Ausbau

Tabelle 4-11 Einbau des Klemmenblocks

Aufgabenstellung	Vorgehensweise
	<p>Bereiten Sie die Komponenten für den Einbau des Klemmenblocks vor, indem Sie die Spannung der CPU ausschalten und die Abdeckung für den Steckverbinder öffnen.</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Richten Sie ihn zu den Stiften im Gerät aus. 3. Richten Sie die Verdrahtungsseite des Klemmenblocks im Stecksockel richtig aus. 4. Drücken Sie den Klemmenblock mit einer Drehbewegung fest nach unten, bis er einrastet.
	<p>Prüfen Sie, ob der Klemmenblock richtig ausgerichtet ist und fest sitzt.</p>

4.3.7 Einbau und Ausbau des Erweiterungskabels

Das S7-1200 Erweiterungskabel sorgt für zusätzliche Flexibilität beim Konfigurieren des Layouts Ihres S7-1200 Systems. Pro CPU-System dürfen Sie ein Erweiterungskabel verwenden. Sie bauen das Erweiterungskabel entweder zwischen der CPU und dem ersten SM oder zwischen zwei SMs ein.

Tabelle 4-12 Ein- und Ausbau des Steckers des Erweiterungskabels

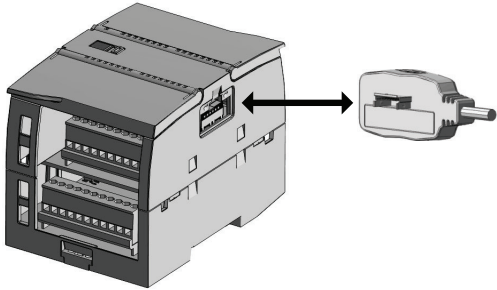
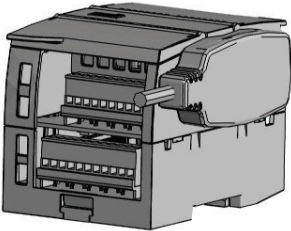
Aufgabenstellung	Vorgehensweise
	<p>Stecker einbauen:</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Drücken Sie den Steckverbinder in den Busanschluss auf der rechten Seite des Signalmoduls oder der CPU. <p>Stecker ausbauen:</p> <ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Ziehen Sie den Stecker vom Signalmodul oder von der CPU ab.
	

Tabelle 4-13 Einbau der Buchse des Erweiterungskabels

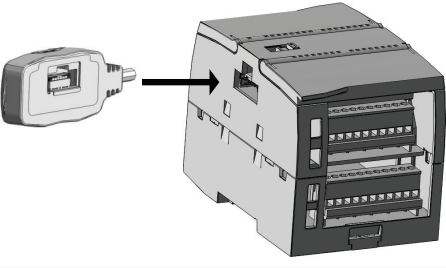
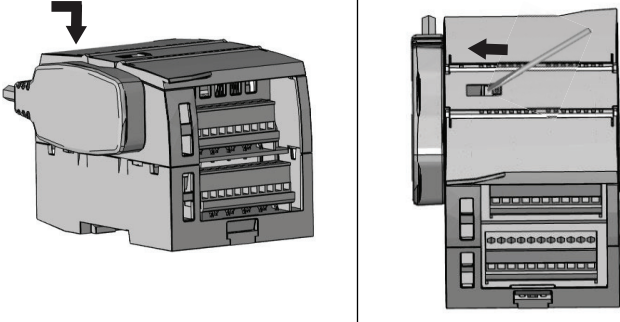
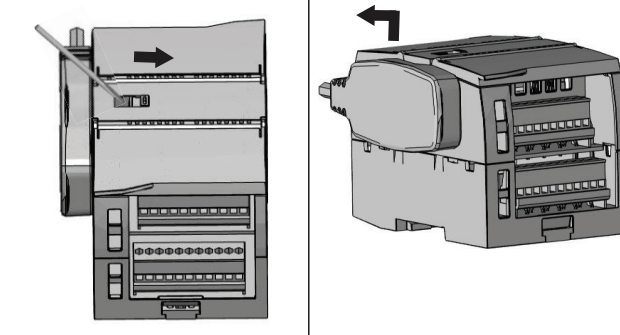
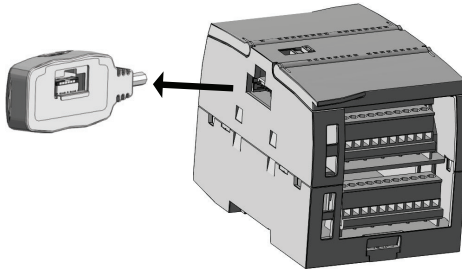
Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Stecken Sie die Buchse auf den Busanschluss an der linken Seite des Signalmoduls. 3. Schieben Sie den Hakenfortsatz der Buchse in das Gehäuse am Busanschluss und drücken Sie vorsichtig, damit der Haken einrastet.
	<ol style="list-style-type: none"> 4. Befestigen Sie die Steckverbindung: <ul style="list-style-type: none"> – Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an. – Schieben Sie die Lasche ganz nach links. <p>Damit der Steckverbinder einrastet, müssen Sie die Lasche des Steckverbinders ganz nach links schieben. Die Steckverbinderlasche muss vollständig einrasten.</p>

Tabelle 4-14 Ausbau der Buchse des Erweiterungskabels

Aufgabenstellung	Vorgehensweise
	<ol style="list-style-type: none"> 1. Stellen Sie sicher, dass die CPU und alle S7-1200 Geräte von der elektrischen Leistung getrennt sind. 2. Lösen Sie den Steckverbinder: <ul style="list-style-type: none"> – Setzen Sie einen Schraubendreher an der Lasche an der Oberseite des Signalmoduls an. – Drücken Sie ihn leicht nach unten und schieben Sie die Lasche ganz nach rechts. 3. Heben Sie den Steckverbinder leicht an, damit sich der Haken löst.
	<ol style="list-style-type: none"> 4. Nehmen Sie die Buchse ab.

Hinweis**Einbau des Erweiterungskabels in schwingungsbelasteten Umgebungen**

Wird das Erweiterungskabel an Module angeschlossen, die sich bewegen oder nicht fest montiert sind, kann sich die Einrastverbindung am Stecker des Kabels im Lauf der Zeit lockern.

Um für zusätzliche Zugentlastung zu sorgen, fixieren Sie in diesem Fall das steckerseitige Kabelende mit einem Kabelbinder auf der DIN-Schiene (oder an einem anderen Träger).

Vermeiden Sie übermäßige Kraftanwendung, wenn das Kabel während des Einbaus gezogen werden muss. Stellen Sie nach beendetem Einbau sicher, dass die Verbindung zwischen Kabel und Modul sich an der korrekten Position befindet.


4.4 Verdrahtungsrichtlinien

Die ordnungsgemäße Erdung und Verdrahtung aller elektrischen Geräte ist wichtig für den optimalen Betrieb Ihres Systems und für die zusätzliche Störfestigkeit Ihrer Anwendung und der S7-1200. Die Schaltpläne der S7-1200 finden Sie in den technischen Daten (Seite 1253).

Voraussetzung

Bevor Sie ein elektrisches Gerät erden oder verdrahten, müssen Sie sicherstellen, dass die Spannungsversorgung der Geräte ausgeschaltet ist. Achten Sie außerdem darauf, dass auch alle angeschlossenen Geräte ausgeschaltet sind.

Stellen Sie sicher, dass Sie bei der Verdrahtung der S7-1200 und aller angeschlossenen Geräte alle geltenden und verbindlichen Normen befolgen. Beachten Sie bei Einbau und Betrieb der Geräte die entsprechenden nationalen und regionalen Vorschriften. Erfragen Sie bei den Behörden vor Ort die Normen und Vorschriften, die für Ihren speziellen Fall zu befolgen sind.

 WARNUNG
--


<p>Wenn Sie die S7-1200 oder daran angeschlossene Geräte in eingeschaltetem Zustand einbauen oder verdrahten, kann es passieren, dass Sie einen elektrischen Schlag bekommen oder die Geräte unerwartet arbeiten.</p>
--

<p>Ist die Spannungsversorgung der S7-1200 und aller daran angeschlossenen Geräte während des Einbaus bzw. Ausbaus von Geräten nicht abgeschaltet, so kann dies aufgrund von elektrischem Schlag oder unerwartetem Betrieb der Geräte zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p>

<p>Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S71200 abgeschaltet ist.</p>
--

Denken Sie beim Planen von Erdung und Verdrahtung Ihres S7-1200 Systems immer an die Sicherheit. Elektronische Steuerungsgeräte wie die S7-1200 können ausfallen und dadurch unerwarteten Betrieb der gesteuerten oder beobachteten Geräte hervorrufen. Deshalb sollten

Sie Sicherheitseinrichtungen implementieren, die von der S7-1200 unabhängig sind und vor möglichen Personen- und/oder Sachschäden schützen.


 WARNUNG
<p>Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen.</p> <p>Unvorhersehbarer Betrieb kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p> <p>Sorgen Sie daher für eine NOTAUSfunktion, elektromechanische oder andere redundante Sicherheitseinrichtungen, die von Ihrer S7-1200 unabhängig sind.</p>

Richtlinien für die Potentialtrennung

Die Grenzwerte der AC-Spannungsversorgung und die E/A-Grenzen zu AC-Stromkreisen der S7-1200 sind dafür konzipiert und zugelassen, sichere elektrische Trennung zwischen AC-Netzspannungen und Niederspannungskreisen zu bieten. Je nach Norm umfassen diese Grenzen doppelte oder verstärkte Isolierung bzw. grundlegende plus zusätzliche Isolierung. Komponenten, die diese Grenzen kreuzen, z.B. optische Koppler, Kondensatoren, Transformatoren und Relais, haben die Zulassung, sichere elektrische Trennung zu bieten. Nur für AC-Netzspannung ausgelegte Stromkreise umfassen die sicherheitsbezogene Potentialtrennung zu anderen Stromkreisen. Die Potentialtrennungsgrenzen zwischen 24-V-DC-Stromkreisen sind rein funktional, und Sie sollten sich bezüglich der Sicherheit nicht auf diese Grenzen verlassen.

Gebersorgungsausgang, Kommunikationsstromkreise und Stromkreise der internen Logik einer S7-1200 mit integrierter AC-Spannungsversorgung werden als Sicherheitskleinspannung (SELV) nach EN 61131-2 gespeist.

Um die sichere Eigenschaft der Niederspannungskreise der S7-1200 zu erhalten, müssen externe Anschlüsse an Kommunikationsports, analoge Stromkreise sowie sämtliche 24-V-DC-Nennspannungsversorgung und alle E/A-Stromkreise aus zugelassenen Quellen gespeist werden, die die Anforderungen nach verschiedenen Normen für SELV, PELV, Klasse 2, spannungsbegrenzte oder leistungsbegrenzte Quellen erfüllen.

 WARNUNG
<p>Der Einsatz nicht potentialgetrennter oder einfach isolierter Versorgungen für die Niederspannungskreise aus AC-Leitungen kann zu gefährlichen Spannungen in Kreisen führen, die als berührungssicher gelten, z.B. Kommunikationskreise oder Niederspannungsgeberverdrahtung.</p> <p>Unerwartet hohe Spannungen können einen elektrischen Schlag verursachen, der zu lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.</p> <p>Verwenden Sie nur solche Hochspannungs-/Niederspannungswandler, die als Quellen berührungssicherer Kreise mit begrenzter Spannung zugelassen sind.</p>

Richtlinien für die Erdung der S7-1200

Am besten erden Sie Ihre Anwendung, indem Sie darauf achten, dass alle gemeinsamen Anschlüsse und alle Erdanschlüsse Ihrer S7-1200 und aller angeschlossenen Geräte an einer einzigen Stelle geerdet werden. Diese Stelle muss direkt mit der Systemerde verbunden werden.

Erdleitungen sollten möglichst kurz sein und einen großen Aderquerschnitt, z. B. 2 mm² (14 AWG) haben.

Beachten Sie beim Auswählen von Erdungspunkten die entsprechenden Sicherheitsvorschriften, und stellen Sie die einwandfreie Funktion von stromkreisunterbrechenden Schutzeinrichtungen sicher.

Richtlinien für die Verdrahtung der S71200

Wenn Sie die Verdrahtung Ihrer S7-1200 planen, richten Sie einen Einzeltrennschalter ein, der gleichzeitig die Spannung der Spannungsversorgung für die S7-1200 CPU, die Spannung aller Eingangskreise und die Spannung aller Ausgangskreise trennt. Sorgen Sie für Überstromschutz, z. B. durch eine Sicherung oder einen Schutzschalter, um Fehlerstrom in der Versorgungsverdrahtung zu begrenzen. Sie können zusätzlichen Schutz durch Sicherungen oder andere Strombegrenzungen in den einzelnen Ausgangskreisen implementieren.

Versehen Sie blitzschlaggefährdete Leitungen mit einem geeigneten Überspannungsschutz. Weitere Informationen finden Sie im Abschnitt "Allgemein" der technischen Daten unter Störfestigkeit gegen Stoßspannungen (Seite 1253).

Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in derselben Kabeltrasse wie AC-Versorgungsleitungen und schnellschaltende DC-Starkstromleitungen zu verlegen. Verlegen Sie Leitungen immer paarweise, den Neutral- oder Nullleiter zusammen mit dem Phasenleiter oder der Signalleitung.

Verlegen Sie Leitungen so kurz wie möglich und achten Sie darauf, dass der Leitungsquerschnitt dem benötigten Strom entspricht.

Leitungen und Kabel müssen für eine Temperatur von 30 °C höher als die Umgebungstemperatur der S7-1200 ausgelegt sein (Beispiel: Leitungen, die bei einer Umgebungstemperatur von 55 °C für mindestens 85 °C ausgelegt sind). Ermitteln Sie andere Verdrahtungsarten und Materialanforderungen anhand der spezifischen Bemessungswerte der Stromkreise und Ihrer Installationsumgebung.

Verwenden Sie als optimalen Schutz vor Störfestigkeit geschirmte Leitungen. Typischerweise erreichen Sie durch Erdung des Schirms an der S7-1200 die besten Ergebnisse. Sie müssen den Schirm von Kommunikationskabeln mit S7-1200 Kommunikationssteckergehäusen über Steckverbinder erden, die leitend mit dem Kabelschirm verbunden sind, oder indem Sie den Schirm des Kommunikationskabels an separate Erde anschließen. Erden Sie andere Kabelschirme mittels Klemmen oder Kupferband um den Schirm, um eine große Oberfläche mit dem Erdungspunkt zu verbinden.

Wenn Sie einen Eingangskreis verdrahten, der durch eine externe Spannungsversorgung versorgt wird, implementieren Sie ein Überstromschutzgerät in dem Eingangskreis. Bei Kreisen, die durch die 24-V-DC-Geberversorgung der S7-1200 versorgt werden, ist kein externer Schutz erforderlich, weil die Geberversorgung bereits strombegrenzt ist.

Alle S7-1200 Module verfügen über steckbare Klemmenblöcke für die Anwenderverdrahtung. Achten Sie darauf, dass der Klemmenblock fest sitzt und die Leitungen fest im Klemmenblock verdrahtet sind, damit es nicht zu losen Anschlüssen kommt.

Die S7-1200 arbeitet mit Potentialtrennungsgrenzen, wodurch das Entstehen unerwünschter Ströme in Ihrer Anlage verhindert wird. Wenn Sie die Verdrahtung für Ihr System planen, berücksichtigen Sie diese Potentialtrennungsgrenzen. Im Kapitel Technische Daten (Seite 1314) finden Sie die Werte für die zur Verfügung gestellte Potentialtrennung und die Anordnung der Potentialtrennungsgrenzen. Für AC-Netzspannung ausgelegte Stromkreise umfassen die sicherheitsbezogene Potentialtrennung zu anderen Stromkreisen. Die Potentialtrennungsgrenzen zwischen 24-V-DC-Stromkreisen sind rein funktional, und Sie sollten sich bezüglich der Sicherheit nicht auf diese Grenzen verlassen.

Eine Zusammenfassung der Verdrahtungsregeln für die S7-1200 CPUs, SMs und SBs finden Sie nachstehend:

Tabelle 4-15 Verdrahtungsregeln für S7-1200-CPU, -SMs und -SBs

Verdrahtungsregeln für ...	CPU- und SM-Anschluss		SB-Anschluss
Verbindungstechnik	Push-in-Klemme	Schraube	Schraube
Anschließbare Leiterquerschnitte für Standardadern	2 mm ² bis 0,3 mm ² (14 AWG bis 22 AWG)		1,3 mm ² bis 0,3 mm ² (16 AWG bis 22 AWG)
Anzahl Adern pro Anschluss	1 oder Kombination aus 2 Adern in einem Doppelmantelkabel bis 2 mm ² (gesamt)		1 oder Kombination aus 2 Adern bis 1,3 mm ² (gesamt)
Abisolierlänge	Verwendung von Aderendhülsen für eine sichere elektrische Verbindung	6,4 mm	6,3 bis 7 mm
Anzugsdrehmoment* (maximal)	-/-	0,56 Nm	0,33 Nm
Werkzeug	Schlitzschraubendreher 2,5 bis 3,0 mm		

* Achten Sie darauf, die Schrauben nicht zu überdrehen, um den Anschluss nicht zu beschädigen.

Hinweis

Mit Aderendhülsen auf Litzenleitern reduzieren Sie die Gefahr von Kurzschlüssen durch herausstehende Adern. Aderendhülsen, die länger als die empfohlene Abisolierlänge sind, müssen einen Isolierkragen aufweisen, um Kurzschlüsse durch seitliche Bewegungen der Leiter zu verhindern. Die Begrenzungen der Querschnittsfläche von blanken Leitern gelten auch für Aderendhülsen.

Siehe auch

Technische Daten (Seite 1253)

Richtlinien für Lampenlasten

Lampenlasten, einschließlich von LED-Lampen, schädigen Relaiskontakte aufgrund des hohen Einschaltstoßstroms. Dieser Stoßstrom ist nominal 10 bis 15 Mal so hoch wie der stationäre Strom einer Wolframlampe. Für Lampenlasten, die während der Lebensdauer der Anwendung sehr häufig geschaltet werden, wird ein austauschbares Koppelrelais oder ein Stoßstrombegrenzer empfohlen.

Richtlinien für induktive Lasten

Verwenden Sie bei induktiven Lasten Schutzbeschaltungen, um den Spannungsanstieg beim Ausschalten eines Steuerungsausgangs zu begrenzen. Schutzbeschaltungen schützen die Ausgänge vor frühzeitigem Ausfall aufgrund der Hochspannungstransiente, die auftritt, wenn der Stromfluss durch eine induktive Last unterbrochen wird.

Außerdem begrenzen Schutzbeschaltungen die elektrischen Störungen, die beim Schalten induktiver Lasten entstehen. Hochfrequente Störungen aus schlecht entstörten induktiven Lasten können den Betrieb des PLCs unterbrechen. Am effektivsten verringern Sie elektrische Störungen durch Anordnen einer externen Schutzbeschaltung parallel zur Last und in der Nähe der Last.

Die DC-Ausgänge der S7-1200 umfassen interne Schutzbeschaltungen, die für die induktiven Lasten in den meisten Anwendungen adäquat sind. Da die Relais-Ausgangskontakte der S7-1200 zum Schalten einer DC-Last oder einer AC-Last verwendet werden können, ist kein interner Schutz vorhanden.

Eine gute Lösung ist die Verwendung von Schützen und anderen induktiven Lasten, für die Schutzbeschaltungen angeboten werden, die vom Hersteller ins Lastgerät integriert oder als optionales Zubehör erhältlich sind. Einige vom Hersteller gelieferte Schutzbeschaltungen sind für Ihre Anwendung jedoch möglicherweise ungeeignet. Gegebenenfalls ist eine zusätzliche Schutzbeschaltung notwendig, um eine optimale Rauschminderung und die optimale Lebensdauer der Kontakte zu erreichen.

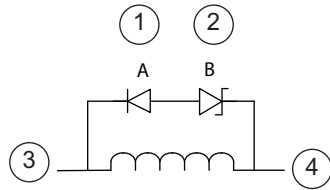
Bei AC-Lasten kann ein Metalloxidvaristor (MOV) oder ein anderer Spannungsbegrenzer mit einer parallelen RC-Beschaltung verwendet werden, diese Vorrichtungen alleine sind jedoch nicht so effektiv. Ein MOV ohne parallele RC-Beschaltung führt häufig zu erheblichen Hochfrequenzstörungen bis zur Höhe der Klemmspannung.

Eine gut beherrschte Ausschalttransiente hat eine Ringfrequenz von maximal 10 kHz, bevorzugt weniger als 1 kHz. Die Spitzenspannung für AC-Leitungen muss innerhalb von +/- 1200 V von Masse liegen. Die negative Spitzenspannung für DC-Lasten mit interner PLC-Schutzbeschaltung liegt ~40 V unterhalb der 24-V-DC-Versorgungsspannung. Eine externe Schutzbeschaltung muss die Transiente auf einen Wert innerhalb von 36 V der Versorgung begrenzen, um die interne Schutzbeschaltung zu entlasten.

Hinweis

Die Effektivität einer Schutzbeschaltung hängt von der jeweiligen Anwendung ab und muss immer für den Einzelfall geprüft werden. Achten Sie darauf, dass alle Komponenten richtig bemessen sind, und beobachten Sie die Ausschalttransiente mittels eines Oszilloskops.

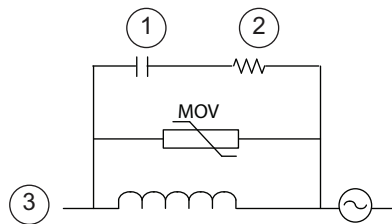
Typische Schutzbeschaltungen für DC- oder Relaisausgänge, die induktive DC-Lasten schalten



- ① Diode 1N4001 oder gleichwertig
- ② Zener-Diode 8,2 V (DC-Ausgänge),
Zener-Diode 36 V (Relaisausgänge)
- ③ Ausgang
- ④ M, 24-V-Referenz

In den meisten Anwendungen ist der zusätzliche Einsatz einer Diode (A) quer zur induktiven DC-Last geeignet, doch wenn für Ihre Anwendung schnellere Ausschaltzeiten erforderlich sind, ist der Einsatz einer Zener-Diode (B) empfehlenswert. Bemessen Sie die Zener-Diode gemäß dem Strom im Ausgangskreis.

Typische Schutzbeschaltungen für Relaisausgänge, die induktive AC-Lasten schalten



- ① Den Wert für C finden Sie in der Tabelle
- ② Den Wert für R finden Sie in der Tabelle
- ③ Ausgang

Achten Sie darauf, dass die Arbeitsspannung des Metalloxidvaristors (MOV) mindestens 20 % höher ist als die Nennspannung.

Verwenden Sie für Impulsanwendungen empfohlene impulsbemessene, nicht induktive Widerstände und Kondensatoren (üblicherweise Metallschicht). Achten Sie darauf, dass die Komponenten die Anforderungen an die durchschnittliche Leistung, Spitzenleistung und Spitzenspannung erfüllen.

Wenn Sie Ihre eigene Schutzbeschaltung entwerfen, finden Sie in der folgenden Tabelle Werte für Widerstand und Kondensator für eine Vielzahl von AC-Lasten. Diese Werte basieren auf Berechnungen mit idealen Komponentenparametern. Die Angabe I effektiv in der Tabelle bezieht sich auf den Beharrungsstrom der Last im vollständig eingeschalteten Zustand.

Tabelle 4-16 Widerstands- und Kondensatorwerte für AC-Schutzbeschaltungen

Induktive Last			Schutzbeschaltungswerte		
I effektiv	230 V AC	120 V AC	Widerstand		Kondensator
A	VA	VA	Ω	W (Nennleistung)	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

Induktive Last			Schutzbeschaltungswerte		
1	230	120	270	5	1000
2	460	240	150	10	1500

Von den Tabellenwerten erfüllte Bedingungen:

Maximaler Ausschaltschritt < 500 V

Spitzenspannung des Widerstands < 500 V

Spitzenspannung des Kondensators < 1250 V

Strom der Schutzbeschaltung < 8 % des Laststroms (50 Hz)

Strom der Schutzbeschaltung < 11 % des Laststroms (60 Hz)

Kondensator $dV/dt < 2 V/\mu s$

Impulsverlust des Kondensators: $\int (dv/dt)^2 dt < 10000 V^2/\mu s$

Resonanzfrequenz < 300 Hz

Widerstandsleistung bei max. 2 Hz Schaltfrequenz

Angenommener Leistungsfaktor von 0,3 bei typischen induktiven Lasten

Richtlinien für Differenzein- und -ausgänge

Differenzeingänge und -ausgänge verhalten sich anders als Standardein- und -ausgänge. Jeder Differenzein- und -ausgang besitzt zwei Anschlussstifte. Um festzustellen, ob ein Differenzeingang oder -ausgang ein- oder ausgeschaltet ist, muss die Spannungsdifferenz zwischen diesen beiden Anschlussstiften gemessen werden.

Beachten Sie die ausführlichen technischen Daten zur CPU 1217C in Anhang A (Seite 1314).

PLC-Grundlagen

5.1 Ausführung des Anwenderprogramms

Die CPU unterstützt die folgenden Bausteinarten für den Aufbau einer geeigneten Struktur Ihres Anwenderprogramms:

- Organisationsbausteine (OBs) legen die Struktur des Programms fest. Für einige OBs gibt es vordefiniertes Verhalten und Startereignisse, Sie können aber auch OBs mit Ihren eigenen Startereignissen anlegen.
- Funktionen (FCs) und Funktionsbausteine (FBs) enthalten den Programmcode, der den jeweiligen Aufgaben oder Parametrierungen entspricht. Jede FC bzw. jeder FB stellt eine Anzahl Ein- und Ausgangsparameter für die gemeinsame Nutzung der Daten mit dem aufrufenden Baustein bereit. Ein FB verwendet ferner einen weiteren Datenbaustein (den Instanz-DB) für die Speicherung von Datenwerten für die jeweilige Instanz des FB-Aufrufs. Sie können einen FB mehrere Male aufrufen, jedes Mal mit einem eindeutigen Instanz-DB. Aufrufe desselben FB mit unterschiedlichen Instanz-DBs wirken sich nicht auf die Datenwerte in einem der anderen Instanz-DBs aus.
- Datenbausteine (DBs) speichern Daten, die von den Programmbausteinen verwendet werden können.

Die Ausführung des Anwenderprogramms beginnt mit einem oder mehreren optionalen Anlauf-OBs, die nach dem Wechsel in den Betriebszustand RUN einmal abgearbeitet werden, gefolgt von einem oder mehreren Zyklus-OBs, die zyklisch abgearbeitet werden. Sie können einen OB auch einem Alarmereignis zuordnen, bei dem es sich um ein Standard- oder ein Fehlerereignis handeln kann. Diese OBs werden ausgeführt, wenn das entsprechende Standard- oder Fehlerereignis auftritt.

Eine Funktion (FC) oder ein Funktionsbaustein (FB) ist ein Baustein mit Programmcode, der aus einem OB oder einer anderen FC oder einem FB aufgerufen werden kann. Folgende Schachtelungstiefen sind dabei möglich:

- 16 aus Zyklus- oder Anlauf-OBs
 - 6 aus beliebigen Alarmereignis-OBs
- Hinweis: Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.


FCs sind keinem bestimmten Datenbaustein (DB) zugeordnet. FBs sind direkt mit einem DB verbunden und nutzen diesen für die Übergabe von Parametern und die Speicherung von Zwischenwerten und -ergebnissen.

Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl der einzelnen OBs, FCs, FBs und DBs ist nicht begrenzt. Die Gesamtzahl der Bausteine darf jedoch 1024 nicht überschreiten.

Jeder Zyklus umfasst das Schreiben der Ausgänge, das Lesen der Eingänge, das Bearbeiten der Anweisungen des Anwenderprogramms und die Durchführung der Hintergrundverarbeitung. Der Zyklus wird auch als Abtastzyklus oder Abtastung bezeichnet.

Ihre S7-1200 Automatisierungslösung kann aus einem zentralen Baugruppenträger mit der S7-1200 CPU und weiteren Modulen bestehen. Der Begriff "zentraler Baugruppenträger" bezieht sich auf den Hutschienen- oder Schaltschrank einbau der CPU und der zugehörigen Module. Die Module (SM, SB, BB, CB, CM oder CP) werden erkannt und erst beim Anlauf angemeldet.

- Das Stecken oder Ziehen eines Moduls im zentralen Baugruppenträger bei eingeschaltetem Gerät ist nicht möglich. Stecken oder ziehen Sie niemals ein Modul im zentralen Baugruppenträger, wenn die CPU eingeschaltet ist.

<p> WARNUNG</p> <p>Sicherheitsanforderungen zum Stecken oder Ziehen von Modulen</p> <p>Das Stecken oder Ziehen eines Moduls (SM, SB, BB, CD, CM oder CP) im zentralen Baugruppenträger bei eingeschalteter CPU kann Schäden oder unvorhersehbares Verhalten verursachen, was wiederum zu Tod, schweren Verletzungen und/oder Sachschaden führen kann.</p> <p>Trennen Sie die Spannungsversorgung von CPU und zentralem Baugruppenträger und befolgen Sie die entsprechenden Sicherheitsvorkehrungen, bevor Sie ein Modul im zentralen Baugruppenträger stecken oder ziehen.</p>
--

- Eine SIMATIC Memory Card können Sie bei eingeschalteter CPU stecken oder ziehen. Das Stecken oder Ziehen einer Memory Card im Betriebszustand RUN der CPU verursacht jedoch, dass die CPU in STOP geht.

<p>ACHTUNG</p> <p>Risiken beim Entfernen der Memory Card, wenn sich die CPU in RUN befindet</p> <p>Durch Stecken oder Ziehen einer Memory Card im Betriebszustand RUN der CPU geht die CPU in den Betriebszustand STOP, was zu Sachschaden an den gesteuerten Geräten oder im gesteuerten Prozess führen kann.</p> <p>Wenn Sie eine Memory Card stecken oder ziehen, geht die CPU sofort in den Betriebszustand STOP. Stellen Sie vor dem Stecken oder Ziehen einer Memory Card stets sicher, dass die CPU nicht aktiv eine Maschine oder einen Prozess steuert. Installieren Sie immer einen NOT-AUS-Schaltkreis für Ihre Anwendung bzw. Ihren Prozess.</p>
--

- Wenn Sie in einem dezentralen E/A-Baugruppenträger (AS-i, PROFINET oder PROFIBUS) im Betriebszustand RUN der CPU ein Modul stecken oder ziehen, generiert die CPU einen Eintrag im Diagnosepuffer, führt den OB "Ziehen oder Stecken von Modulen" aus (sofern vorhanden) und bleibt standardmäßig im Betriebszustand RUN.

Aktualisierung von Prozessabbildern und Teilprozessabbildern

Die CPU aktualisiert die lokalen digitalen und analogen Ein- und Ausgänge synchron zum Zyklus mit einem internen Speicherbereich, dem so genannten Prozessabbild. Das Prozessabbild enthält eine Momentaufnahme der physischen Ein- und Ausgänge (physische E/A von CPU, Signalboard und Signalmodulen).

Sie können die E/A so einrichten, dass sie in jedem Zyklus im Prozessabbild aktualisiert werden oder immer dann, wenn ein spezifisches Alarmereignis auftritt. Sie können die E/A auch so einrichten, dass sie aus der Aktualisierung des Prozessabbilds ausgeschlossen werden. Beispiel: Ihr Prozess benötigt bestimmte Datenwerte möglicherweise immer dann, wenn ein Ereignis wie ein Prozessalarm auftritt. Indem Sie die Aktualisierung des Prozessabbilds für diese E/A einem

Teilprozessabbild zuweisen, das Sie wiederum einem Prozessalarm-OB zuweisen, aktualisiert die CPU diese Datenwerte nicht unnötigerweise in jedem Zyklus, da Ihr Prozess die ständige Aktualisierung nicht erfordert.

Für E/A, die in jedem Zyklus aktualisiert werden, führt die CPU in jedem Zyklus die folgenden Aufgaben durch:

- Die CPU schreibt die Ausgänge aus dem Prozessabbild der Ausgänge in die physischen Ausgänge.
- Die CPU liest die physischen Eingänge unmittelbar vor der Ausführung des Anwenderprogramms und speichert die Eingangswerte im Prozessabbild der Eingänge. Diese Werte bleiben somit während der Ausführung der Anwenderanweisungen konsistent.
- Die CPU führt die Logik der Anwenderanweisungen durch und aktualisiert die Ausgangswerte im Prozessabbild der Ausgänge, statt in die tatsächlichen physischen Ausgänge zu schreiben.
Dieser Vorgang sorgt während der gesamten Ausführung der Anwenderanweisungen in dem jeweiligen Zyklus für eine konsistente Logik und verhindert ein Pendeln der physischen Ausgänge mit mehrmaligen Zustandswechseln im Prozessabbild der Ausgänge.

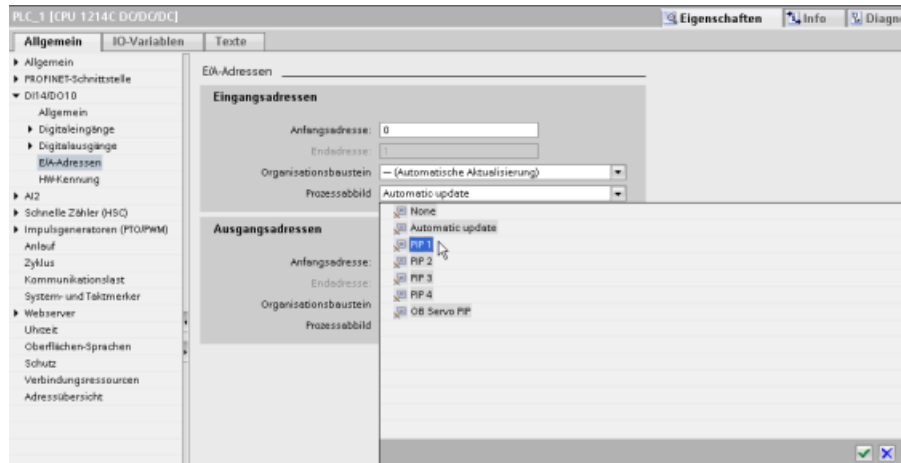
Um zu steuern, ob Ihr Prozess die E/A automatisch in jedem Zyklus oder bei der Auslösung von Ereignissen aktualisiert, stellt die S7-1200 fünf Teilprozessabbilder bereit. Das erste Teilprozessabbild, TPA0, ist für E/A bestimmt, die automatisch in jedem Zyklus aktualisiert werden. Hierbei handelt es sich um die Standardzuweisung. Den übrigen vier Teilprozessabbildern, TPA1, TPA2, TPA3 und TPA4, können Sie die Aktualisierung des Prozessabbilds bei verschiedenen Alarmereignissen zuweisen. Die E/A weisen Sie den Teilprozessabbildern in der Gerätekonfiguration zu, und die Teilprozessabbilder wiederum weisen Sie den Alarmereignissen zu, wenn Sie Alarm-OBs erstellen (Seite 178) oder OB-Eigenschaften bearbeiten (Seite 178).

Standardmäßig stellt STEP 7 beim Einfügen eines Moduls in der Gerätesicht für die Aktualisierung des Prozessabbilds die Option "Automatische Aktualisierung" ein. Bei E/A, für die die "Automatische Aktualisierung" vorgegeben ist, bearbeitet die CPU den Datenaustausch zwischen dem Modul und dem Prozessabbild automatisch in jedem Zyklus.

Um digitale oder analoge E/A einem Teilprozessabbild zuzuweisen oder um E/A aus der Aktualisierung des Prozessabbilds auszuschließen, gehen Sie wie folgt vor:

1. Öffnen Sie für das entsprechende Gerät das Register "Eigenschaften" in der Gerätekonfiguration.
2. Erweitern Sie ggf. die Auswahl unter "Allgemein", um die gewünschten E/A anzuzeigen.
3. Wählen Sie "E/A-Adressen".

4. Wählen Sie optional einen spezifischen OB aus der Klappliste "Organisationsbaustein" aus.
5. Ändern Sie über die Klappliste "Prozessabbild" die Einstellung "Automatische Aktualisierung" in "TPA1", "TPA2", "TPA3", "TPA4" oder "Keine". Die Auswahl "Keine" bedeutet, dass Sie diese E/A nur über direkte Anweisungen lesen und schreiben können. Um die Eingänge und Ausgänge wieder in die automatische Aktualisierung des Prozessabbilds aufzunehmen, stellen Sie für diese Option wieder "Automatische Aktualisierung" ein.



Sie können bei Ausführung einer Anweisung die Werte physischer Eingänge direkt lesen und auch direkt Werte in physische Ausgänge schreiben. Beim direkten Lesen wird auf den aktuellen Zustand des physischen Eingangs zugegriffen. Das Prozessabbild der Eingänge wird, unabhängig davon, ob der Eingang für die Speicherung im Prozessabbild konfiguriert ist, nicht aktualisiert. Beim direkten Schreiben in den physischen Ausgang werden sowohl das Prozessabbild der Ausgänge (sofern der Ausgang für die Speicherung im Prozessabbild konfiguriert ist) als auch der physische Ausgang aktualisiert. Hängen Sie die Endung ":P" an die E/A-Adresse an, wenn Sie möchten, dass das Programm direkt über den physischen Eingang bzw. Ausgang auf die E/A-Daten zugreift und nicht über das Prozessabbild.

Hinweis

Verwenden von Teilprozessabbildern

Wenn Sie einem der Teilprozessabbilder TPA1 bis TPA4 einige E/A zuweisen, dem Teilprozessabbild jedoch keinen OB zuweisen, aktualisiert die CPU diese E/A niemals im Prozessabbild. Die Zuweisung von E/A zu einem TPA ohne entsprechende OB-Zuweisung kommt der Einstellung des Prozessabbilds für "Keine" Aktualisierung gleich. Sie können die E/A mit einer direkten Leseanweisung direkt aus den physischen E/A lesen oder mit einer direkten Schreibweisung direkt in die physische E/A schreiben. Die CPU aktualisiert das Prozessabbild nicht.

Die CPU unterstützt dezentrale Peripherie bei PROFINET, PROFIBUS und AS-i-Netzwerken (Seite 589).

5.1.1 Betriebszustände der CPU

Die CPU hat drei Betriebszustände: Betriebszustand STOP, Betriebszustand STARTUP und Betriebszustand RUN. Die Status-LEDs auf der Vorderseite der CPU geben den aktuellen Betriebszustand an.

- Im Betriebszustand STOP führt die CPU das Programm nicht aus. Sie können ein Projekt in die CPU laden.
- Im Betriebszustand STARTUP werden die Anlauf-OBs (sofern vorhanden) einmal abgearbeitet. In diesem Betriebszustand verarbeitet die CPU keine Prozessalarmereignisse.
- Im Betriebszustand RUN werden die Programmzyklus-OBs wiederholt ausgeführt. Es können jederzeit im Betriebszustand RUN Alarmereignisse auftreten, die die Ausführung der entsprechenden Alarmereignis-OBs bewirken. Sie können Teile eines Projekts im Betriebszustand RUN laden (Seite 1232).

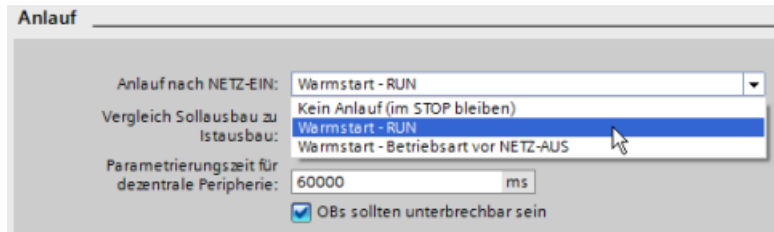
Die CPU unterstützt den Warmstart, um in den Betriebszustand RUN zu gehen. Während eines Warmstarts wird kein Urlöschen ausgeführt. Die CPU initialisiert alle nicht remanenten System- und Anwenderdaten bei einem Warmstart und puffert die Werte aller remanenten Anwenderdaten.

Beim Urlöschen werden der Arbeitsspeicher sowie alle remanenten und nicht remanenten Speicherbereiche gelöscht, der Ladespeicher in den Arbeitsspeicher kopiert und die Ausgänge in die konfigurierte "Reaktion auf CPU-STOP" versetzt. Der Diagnosepuffer und die dauerhaft gespeicherten Werte der IP-Adresse werden beim Urlöschen nicht gelöscht.

Sie können die Einstellung "Anlauf nach NETZ-EIN" der CPU konfigurieren. Diese Einstellungen finden Sie in der Gerätekonfiguration der CPU unter "Anlauf". Beim Einschalten führt die CPU eine Reihe von Diagnoseprüfungen und anschließend die Systeminitialisierung durch. Während der Initialisierung des Systems löscht die CPU den gesamten nicht remanenten Speicherbereich der Merker und setzt alle nicht remanenten DB-Inhalte auf die Anfangswerte aus dem Ladespeicher zurück. Die CPU puffert den remanenten Speicherbereich der Merker und remanente DB-Inhalte und nimmt dann den entsprechenden Betriebszustand ein. Bestimmte

Fehler verhindern, dass die CPU in den Betriebszustand RUN geht. Die CPU unterstützt die folgenden Konfigurationsoptionen:

- Kein Neustart (in STOP bleiben)
- Warmstart - RUN
- Warmstart - Betriebsart vor NETZ-AUS



ACHTUNG

Behebbarer Fehler können die CPU veranlassen, in den Betriebszustand STOP zu gehen.

Die CPU kann aufgrund von behebbaren Fehlern in den Betriebszustand STOP gehen, z. B. bei:

- Ausfall eines ersetzbaren Signalmoduls
- Temporären Fehlern wie Störungen der Netzleitung oder einem unvorhersehbaren Anlaufereignis.

Solche Situationen können zu Sachschäden führen.

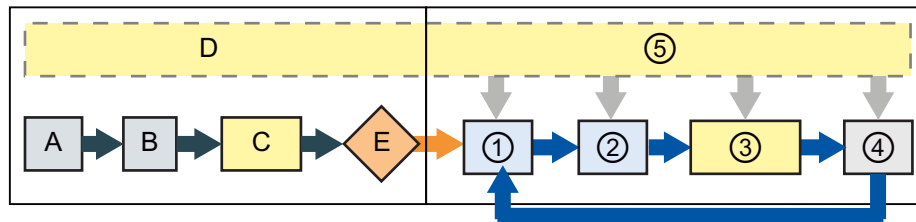
Wenn für die CPU die Einstellung "Warmstart - Betriebsart vor NETZ-AUS" konfiguriert ist, kehrt sie in den Betriebszustand zurück, in dem sie vor dem Netzausfall oder Fehler war. Befand sich die CPU zum Zeitpunkt des Netzausfalls oder Fehlers im Betriebszustand STOP, kehrt sie beim Anlaufen in STOP zurück. Die CPU bleibt im Betriebszustand STOP, bis sie einen Befehl für den Wechsel nach RUN erhält. Befand sich die CPU zum Zeitpunkt des Netzausfalls oder Fehlers im Betriebszustand RUN, kehrt sie beim Anlaufen in RUN zurück. Die CPU kehrt in RUN zurück, sofern sie keine Fehler erkennt, die den Wechsel in RUN verhindern würden.

CPUs, die unabhängig von einer STEP 7-Verbindung laufen sollen, sollten für "Warmstart - RUN" konfiguriert werden. Damit kann die CPU beim nächsten Anlauf in den Betriebszustand RUN zurückkehren.

Sie können den aktuellen Betriebszustand mit den Befehlen "STOP" und "RUN" (Seite 1218) in den Online-Tools der Programmiersoftware ändern. Sie können ferner eine STP-Anweisung (Seite 305) in Ihr Programm einfügen, um die CPU in den Betriebszustand STOP zu versetzen. Auf

diese Weise können Sie die Bearbeitung Ihres Programms abhängig von der Programmlogik unterbrechen.

- Im Betriebszustand STOP bearbeitet die CPU Kommunikationsanforderungen (je nach Bedarf) und führt Selbstdiagnosen durch. Die CPU führt das Anwenderprogramm nicht aus. Es finden keine automatischen Aktualisierungen des Prozessabbilds statt.
- In den Betriebszuständen STARTUP und RUN führt die CPU die im folgenden Bild gezeigten Aufgaben aus.



STARTUP

- A Der Zustand der physischen Eingänge wird in den Speicherbereich E kopiert
- B Speicherbereich A (Bild) wird je nach Konfiguration mit null, dem letzten Wert oder dem konfigurierten Ersatzwert initialisiert. PB-, PN- und AS-i-Ausgänge werden auf null gesetzt.
- C Nicht remanente Speicherbereiche der Merker und Datenbausteine werden mit ihrem Anfangswert initialisiert und die konfigurierten Weckalarm- und Uhrzeitereignisse werden aktiviert.
Die Anlauf-OBs werden ausgeführt.
- D Alle Alarmereignisse werden in der Warteschlange für die Verarbeitung im Betriebszustand RUN gespeichert
- E Das Schreiben von Speicherbereich A in die physischen Ausgänge wird aktiviert

RUN

- ① Speicherbereich A wird in die physischen Ausgänge geschrieben
- ② Der Zustand der physischen Eingänge wird in den Speicherbereich E kopiert
- ③ Die Programmzyklus-OBs werden ausgeführt
- ④ Führt Selbstdiagnose durch
- ⑤ Alarme und Kommunikation werden in allen Teilen des Zyklus bearbeitet

Hinweis

Die Kommunikation, einschließlich HMI-Kommunikation, kann nur Programmzyklus-OBs unterbrechen.

STARTUP-Verarbeitung

Immer wenn der Betriebszustand von STOP nach RUN wechselt, löscht die CPU das Prozessabbild der Eingänge, initialisiert das Prozessabbild der Ausgänge und verarbeitet die Anlauf-OBs. Alle Lesezugriffe auf das Prozessabbild der Eingänge von Anweisungen in den Anlauf-OBs ergeben den Wert null und nicht den aktuellen Wert des physischen Eingangs. Deshalb müssen Sie, um den aktuellen Zustand eines physischen Eingangs im Betriebszustand STARTUP zu lesen, den Eingang direkt lesen. Die Anlauf-OBs und zugehörigen FCs und FBs werden anschließend ausgeführt. Sind mehrere Anlauf-OBs vorhanden, so arbeitet die CPU diese fortlaufend nach OB-Nummer, beginnend mit der niedrigsten Nummer ab.

Jeder Anlauf-OB enthält Anlaufinformationen, damit Sie die Gültigkeit der remanenten Daten und der Echtzeituhr ermitteln können. Sie können in den Anlauf-OBs Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen. Die folgenden Anlaufadressen werden von Anlauf-OBs unterstützt:

Tabelle 5-1 Vom Anlauf-OB unterstützte Anlaufadressen

Eingang	Datentyp	Beschreibung
LostRetentive	Bool	Dieses Bit ist wahr, wenn die Speicherbereiche der remanenten Daten verloren gegangen sind.
LostRTC	Bool	Dieses Bit ist wahr, wenn die Echtzeituhr verloren gegangen ist.

Die CPU führt während der Anlaufverarbeitung auch die folgenden Aufgaben aus.

- Alarme werden während der Anlaufphase in eine Warteschlange gestellt und nicht bearbeitet
- In der Anlaufphase findet keine Zykluszeitüberwachung statt
- Beim Anlaufen kann die Konfiguration der schnellen Zähler (HSC), der Impulsdauermodulation (PWM) und der Punkt-zu-Punkt-Kommunikationsmodule geändert werden
- Die eigentlichen Funktionsabläufe von HSC, PWM und PTP-Modulen finden nur im Betriebszustand RUN statt

Nach der Ausführung der Anlauf-OBs geht die CPU in den Betriebszustand RUN und bearbeitet die Steuerungsaufgaben in einem fortlaufenden Zyklus.

5.1.2 Verarbeitung des Zyklus im Betriebszustand RUN

In jedem Zyklus schreibt die CPU in die Ausgänge, sie liest die Eingänge, führt das Anwenderprogramm aus, aktualisiert die Kommunikationsmodule und antwortet auf Anwenderalarmereignisse und Kommunikationsanfragen. Kommunikationsanfragen werden während des Zyklus regelmäßig bearbeitet.

Diese Aktionen (außer den Anwenderalarmereignissen) werden zyklisch fortlaufend bearbeitet. Anwenderalarmereignisse, die aktiviert sind, werden nach der Priorität in der Reihenfolge ihres Auftretens abgearbeitet. Bei Alarmereignissen liest die CPU die Eingänge, führt den OB aus und schreibt in die Ausgänge, ggf. mit dem zugehörigen Teilprozessabbild (PIP).

Das System gewährleistet, dass der Zyklus innerhalb der maximalen Zykluszeit abgearbeitet wird, sonst wird ein Zeitfehler erzeugt.

- Jeder Zyklus beginnt mit der Abfrage der aktuellen Werte der digitalen und analogen Ausgänge im Prozessabbild und dem Schreiben dieser Werte in die physischen Ausgänge von CPU, SB und SMs, die für die automatische E/A-Aktualisierung konfiguriert sind (Standardkonfiguration). Greift eine Anweisung auf einen physischen Ausgang zu, so werden der Ausgang im Prozessabbild und der physische Ausgang aktualisiert.
- Im weiteren Verlauf des Zyklus werden die aktuellen Werte der digitalen und analogen Eingänge aus der CPU, der SB und den SMs, die für die automatische E/A-Aktualisierung konfiguriert sind (Standardkonfiguration), ausgelesen und diese Werte in das Prozessabbild geschrieben. Greift eine Anweisung auf einen physischen Eingang zu, so wird der Wert des physischen Eingangs geändert, der Eingang im Prozessabbild jedoch nicht aktualisiert.
- Nach dem Lesen der Eingänge wird das Anwenderprogramm von der ersten Anweisung bis zur letzten Anweisung ausgeführt. Darin enthalten sind alle Programmzyklus-OBs sowie alle zugehörigen FCs und FBs. Die Programmzyklus-OBs werden fortlaufend nach der OB-Nummer, beginnend mit der niedrigsten OB-Nummer, abgearbeitet.

Die Kommunikationsbearbeitung tritt während des Zyklus regelmäßig auf und unterbricht möglicherweise die Ausführung des Anwenderprogramms.

Zu den Selbstdiagnosen gehören regelmäßige Prüfungen des System und die Abfrage des Zustands der E/A-Module.

Alarmer können in jedem Teil des Zyklus auftreten, sie sind ereignisgesteuert. Tritt ein Ereignis auf, so unterbricht die CPU den Zyklus und ruft den OB für die Verarbeitung des Ereignisses auf. Wenn der OB das Ereignis abgearbeitet hat, setzt die CPU die Ausführung des Anwenderprogramms an der Stelle fort, an der es zuvor unterbrochen wurde.

5.1.3 Organisationsbausteine (OBs)

OBs steuern die Ausführung des Anwenderprogramms. Die Ausführung eines Organisationsbausteins wird durch bestimmte Ereignisse in der CPU angestoßen. OBs können sich nicht gegenseitig aufrufen. Sie können auch nicht aus einer FC oder einem FB aufgerufen werden. Nur ein Ereignis wie ein Diagnosealarm oder ein Zeitintervall veranlasst die CPU zur Ausführung eines OB. Die CPU behandelt OBs entsprechend ihrer jeweiligen Prioritätsklasse, wobei OBs mit höherer Priorität vor OBs mit geringerer Priorität ausgeführt werden. Die niedrigste Prioritätsklasse ist 1 (für den Hauptprogrammzyklus), die höchste ist 26.

5.1.3.1 Programmzyklus-OB

Programmzyklus-OBs werden zyklisch ausgeführt, wenn die CPU im Betriebszustand RUN ist. Der Hauptbaustein des Programms ist ein Programmzyklus-OB. Er enthält die Anweisungen für die Steuerung Ihres Programms, und aus ihm heraus werden weitere Anwenderbausteine aufgerufen. Sie können mehrere Programmzyklus-OBs anlegen, die die CPU in numerischer Reihenfolge ausführt. Main (OB 1) ist der Standardbaustein.

Programmzyklusereignisse

Das Programmzyklusereignis tritt einmal in jedem Programmzyklus auf. Während des Programmzyklus schreibt die CPU in die Ausgänge, liest die Eingänge und führt Programmzyklus-OBs aus. Das Programmzyklusereignis ist erforderlich und immer aktiviert. Möglicherweise haben Sie für das Programmzyklusereignis keinen Programmzyklus-OB oder Sie haben mehrere OBs. Nachdem das Programmzyklusereignis eingetreten ist, führt die CPU den Programmzyklus-OB mit der niedrigsten Nummer (in der Regel OB1) aus. Die anderen Programmzyklus-OBs werden von der CPU sequentiell (in numerischer Reihenfolge) innerhalb des Programmzyklus ausgeführt. Das Programm wird zyklisch ausgeführt, so dass das Programmzyklusereignis zu den folgenden Zeitpunkten eintritt:

- Wenn der letzte Anlauf-OB abgearbeitet ist
- Wenn der letzte Programmzyklus-OB abgearbeitet ist

Tabelle 5-2 Anlaufinformationen für einen Programmzyklus-OB

Eingang	Datentyp	Beschreibung
Initial_Call	Bool	Wahr beim ersten Aufruf des OB
Remanence	Bool	Wahr, wenn remanente Daten vorhanden sind

5.1.3.2 Anlauf-OB

Anlauf-OBs werden einmal ausgeführt, wenn der Betriebszustand der CPU von STOP nach RUN wechselt, beim Hochfahren in den Betriebszustand RUN und bei einem vorgegebenen Wechsel von STOP nach RUN. Anschließend beginnt die Ausführung des Programmzyklus.

Anlaufereignisse

Das Anlaufereignis tritt einmal bei einem Wechsel von STOP in RUN auf und verursacht die Ausführung des Anlauf-OBs durch die CPU. Für das Anlaufereignis können Sie mehrere OBs konfigurieren. Die Anlauf-OBs werden in numerischer Reihenfolge ausgeführt.

Tabelle 5-3 Startinformationen für einen Anlauf-OB

Eingang	Datentyp	Beschreibung
LostRetentive	Bool	Wahr, wenn remanente Daten verloren gegangen sind
LostRTC	Bool	Wahr, wenn Datum und Uhrzeit verloren gegangen sind

5.1.3.3 Verzögerungsalarm-OB

Verzögerungsalarm-OBs werden nach einer von Ihnen konfigurierten Zeitverzögerung ausgeführt.

Verzögerungsalarmereignisse

Verzögerungsalarmereignisse konfigurieren Sie für das Auftreten nach einer angegebenen Verzögerungszeit. Die Verzögerungszeit weisen Sie mit der Anweisung SRT_DINT zu. Die Verzögerungsereignisse unterbrechen den Programmzyklus, um den entsprechenden Verzögerungsalarm-OB auszuführen. Einem Verzögerungsereignis können Sie nur einen Verzögerungsalarm-OB zuordnen. Die CPU unterstützt vier Verzögerungsereignisse.

Tabelle 5-4 Startinformationen für einen Verzögerungsalarm-OB

Eingang	Datentyp	Beschreibung
Sign	Word	An auslösenden Aufruf von SRT_DINT übergebene Kennung

5.1.3.4 Weckalarm-OB

Weckalarm-OBs werden in bestimmten Abständen ausgeführt. Sie können bis zu vier Weckalarmereignisse konfigurieren, wobei jedem Ereignis jeweils ein OB entspricht.

Weckalarmereignisse

Die Weckalarmereignisse ermöglichen Ihnen, die Ausführung eines Alarm-OBs zu einer konfigurierten Zykluszeit einzurichten. Die Anfangszykluszeit wird beim Anlegen des Weckalarm-OBs konfiguriert. Ein Weckalarmereignis unterbricht den Programmzyklus und führt den entsprechenden Weckalarm-OB aus. Zu beachten ist, dass ein Weckalarmereignis eine höhere Prioritätsklasse als das Programmzyklusereignis hat.

Einem Weckalarmereignis kann nur genau ein Weckalarm-OB zugewiesen werden.

Sie können jedem Weckalarm eine Phasenverschiebung zuweisen, so dass die Ausführung von Weckalarmen um den Wert der Phasenverschiebung gegeneinander verschoben werden kann. Ist beispielsweise ein Weckalarmereignis für 5 ms und ein weiteres für 10 ms projiziert, treten alle 10 ms beide Ereignisse gleichzeitig auf. Wird das 5-ms-Ereignis um 1 bis 4 ms und das 10-ms-Ereignis um 0 ms phasenverschoben, treten die beiden Ereignisse nicht gleichzeitig auf.

Die Phasenverschiebung ist standardmäßig auf 0 eingestellt. Um die anfängliche Phasenverschiebung oder die Zykluszeit eines zyklischen Ereignisses zu ändern, gehen Sie wie folgt vor:

1. Klicken Sie mit der rechten Maustaste auf den Weckalarm-OB in der Projektnavigation.
2. Wählen Sie im Kontextmenü den Befehl "Eigenschaften".
3. Klicken Sie im Dialog "Weckalarm [OB 30]" auf "Weckalarm" und geben Sie die neuen Anfangswerte ein.

Die maximale Phasenverschiebung beträgt 6000 ms (6 Sekunden) oder die maximale Zykluszeit, je nachdem, was kürzer ist.

Sie können die Zykluszeit und Phasenverschiebung auch aus Ihrem Programm abfragen und ändern. Verwenden Sie dazu die Anweisungen Weckalarm-Parameter abfragen (QRY_CINT) und Weckalarm-Parameter setzen (SET_CINT). Die über die Anweisung SET_CINT eingegebenen Werte für Zykluszeit und Phasenverschiebung werden bei Ausschalten oder Wechsel in STOP nicht gespeichert; beim Wiedereinschalten oder bei Rückkehr in RUN werden wieder die ursprünglichen Werte verwendet. Die CPU unterstützt insgesamt vier Weckalarmereignisse.

5.1.3.5 Prozessalarm-OB

Prozessalarm-OBs werden ausgeführt, sobald das entsprechende Prozessereignis auftritt. Ein Prozessalarm-OB unterbricht den normalen Programmablauf als Reaktion auf ein Signal eines Prozessereignisses.

Prozessalarmereignisse

Prozessalarmereignisse werden durch eine Veränderung in der Hardware ausgelöst, z. B. eine steigende oder fallende Flanke an einem Eingang oder ein Ereignis an einem HSC (schnellen Zähler). Die S7-1200 unterstützt genau einen Alarm-OB für jedes Prozessalarmereignis. Sie können die Prozessereignisse in der Gerätekonfiguration aktivieren und einem Ereignis in der Gerätekonfiguration oder mit einer ATTACH-Anweisung im Anwenderprogramm einen OB zuweisen. Die CPU unterstützt mehrere Prozessalarmereignisse. Welche Ereignisse verfügbar sind, hängt vom CPU-Modell und der Anzahl der Eingänge ab.

Für Prozessalarmereignisse gelten die folgenden Grenzwerte:

Flanken:

- Ereignisse steigende Flanke: maximal 16
- Ereignisse fallende Flanke: maximal 16

HSC-Ereignisse:

- CV=PV: maximal 6
- Richtungswechsel: maximal 6
- Externes Rücksetzen: maximal 6

Tabelle 5-5 Anlaufinformationen für einen Weckalarm-OB

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardwarekennung des Moduls, das den Weckalarm ausgelöst hat.
USI	WORD	Anwenderstrukturkennung (16#0001 bis 16#FFFF), reserviert für künftige Verwendung
IChannel	USINT	Nummer des Kanals, der den Alarm ausgelöst hat
EventType	BYTE	Kennung des modulspezifischen Ereignistyps des Ereignisses, das den Weckalarm auslöst, z. B. eine fallende oder steigende Flanke

Die Bits in EventType sind von dem auslösenden Modul abhängig. Dies wird nachstehend gezeigt:

Modul/ Submodul	Wert	Prozessereignis
Integrierte E/A von CPU oder SB	16#0	Steigende Flanke
	16#1	Fallende Flanke
HSC	16#0	HSC CV=RV1
	16#1	HSC Richtungswechsel
	16#2	HSC Rücksetzen
	16#3	HSC CV=RV2

5.1.3.6 Zeitfehler-OB

Sofern konfiguriert, wird der Zeitfehler-OB (OB 80) ausgeführt, wenn die maximale Zykluszeit überschritten wird oder ein Zeitfehlerereignis auftritt. Wird dieser OB ausgelöst, beginnt die Ausführung, die den normalen Programmablauf oder auch einen anderen Ereignis-OB unterbricht.

Das Auftreten eines dieser Ereignisse erzeugt einen Eintrag im Diagnosepuffer, der das Ereignis beschreibt. Der Eintrag im Diagnosepuffer wird unabhängig davon erzeugt, ob ein Zeitfehler-OB vorhanden ist oder nicht.

Zeitfehleralarmereignisse

Das Auftreten eines beliebigen Zeitfehlers führt zu einem Zeitfehlerereignis:

- **Überschreiten der maximalen Zykluszeit**
Der Fehler "Maximale Zykluszeit überschritten" tritt auf, wenn der Programmzyklus nicht innerhalb der angegebenen maximalen Zykluszeit beendet wird. Kapitel "Überwachen und Konfigurieren der Zykluszeit" (Seite 90) enthält ausführlichere Informationen über die maximale Zykluszeit, die Konfiguration der maximalen Zykluszeit in den Eigenschaften der CPU und das Zurücksetzen des Zykluszählers.
- **CPU kann den angeforderten OB nicht starten, weil ein zweiter Alarm (Weck- oder Verzögerungsalarm) gestartet wurde, bevor die CPU die Ausführung des ersten Alarm-OBs beendet hat**
- **Warteschlangenüberlauf**
Der Fehler "Warteschlangenüberlauf" tritt ein, wenn die Alarme schneller auftreten, als sie von der CPU verarbeitet werden können. Die Zahl anstehender Ereignisse wird von der CPU begrenzt, indem jedem Ereignistyp eine eigene Warteschlange zugewiesen wird. Tritt ein Ereignis auf, wenn die entsprechende Warteschlange voll ist, wird von der CPU ein Zeitfehlerereignis erzeugt.

Alle Zeitfehlerereignisse lösen die Ausführung des Zeitfehler-OBs (sofern vorhanden) aus. Ist kein Zeitfehler-OB vorhanden, geht die CPU in den Betriebszustand STOP.

Durch Ausführen der Anweisung RE_TRIGR (Seite 304) zum Neustarten der Zykluszeitüberwachung kann das Anwenderprogramm die Ausführung des Programmzyklus bis zum Zehnfachen der konfigurierten maximalen Zykluszeit verlängern. Wenn jedoch der Fehler "Maximale Zykluszeit überschritten" zweimal in demselben Programmzyklus auftritt, ohne dass die Zykluszeit zurückgesetzt wird, geht die CPU in STOP, unabhängig davon, ob der Zeitfehler-OB vorhanden ist. Weitere Informationen finden Sie im Abschnitt "Überwachen der Zykluszeit" im S7-1200 Systemhandbuch (Seite 90).

Der Zeitfehler-OB enthält Anlaufinformationen, anhand derer Sie ermitteln können, welches Ereignis und welcher OB den Zeitfehler erzeugt hat. Sie können im OB Anweisungen

programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Tabelle 5-6 Anlaufinformationen für den Zeitfehler-OB (OB 80)

Eingang	Datentyp	Beschreibung
fault_id	BYTE	16#01 - Maximale Zykluszeit überschritten 16#02 - Gewünschter OB kann nicht gestartet werden 16#07 und 16#09 - Warteschlangenüberlauf
csg_OBnr	OB_ANY	Nummer des OBs, der bei Auftreten des Fehlers ausgeführt wurde
csg_prio	UINT	Priorität des fehlerverursachenden OBs

Um einen Zeitfehler-OB in Ihr Projekt einzufügen, müssen Sie einen Zeitfehleralarm hinzufügen, indem Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen" doppelklicken und dann "Organisationsbaustein" und "Zeitfehler" auswählen.

Die Priorität einer neuen CPU V4.0 ist 22. Wenn Sie eine CPU V3.0 durch eine CPU V4.0 (Seite 1456) ersetzen, ist die Priorität 26, also die Priorität, die für V3.0 gültig war. In beiden Fällen kann das Prioritätsfeld geändert werden und Sie können für die Priorität einen beliebigen Wert im Bereich von 22 bis 26 festlegen.

5.1.3.7 Diagnosefehler-OB

Der Diagnosefehler-OB wird ausgeführt, wenn die CPU einen Diagnosefehler erkennt oder wenn ein diagnosefähiges Modul einen Fehler erkennt und der Diagnosefehleralarm für das Modul aktiviert ist. Ein Diagnosefehler-OB unterbricht den normalen Programmablauf. Wenn die CPU in den Betriebszustand STOP gehen soll, sobald dieser Fehlertyp erkannt wird, können Sie in den Diagnosefehler-OB eine STP-Anweisung einfügen.

Wenn Sie keinen Diagnosefehler-OB in Ihr Programm einfügen, ignoriert die CPU den Fehler und bleibt in RUN.

Diagnosefehlerereignisse

Analoge (lokale), PROFINET-, PROFIBUS- und einige digitale (lokale) Geräte können Diagnosefehler erkennen und melden. Das Auftreten bzw. Verschwinden eines von verschiedenen Diagnosefehlern führt zu einem Diagnosefehlerereignis. Die folgenden Diagnosefehler werden unterstützt:

- Keine Anwenderspannung
- Oberer Grenzwert überschritten
- Unterer Grenzwert überschritten
- Drahtbruch
- Kurzschluss

Diagnosefehlerereignisse lösen die Ausführung des Diagnosefehler-OBs (OB 82) aus, sofern dieser vorhanden ist. Ist er nicht vorhanden, ignoriert die CPU den Fehler.

Um einen Diagnosefehler-OB in Ihr Projekt einzufügen, müssen Sie einen Diagnosefehleralarm hinzufügen. Hier doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf

"Neuen Baustein hinzufügen" und wählen dann "Organisationsbaustein" und "Diagnosefehler" aus.

Hinweis

Diagnosefehler bei mehrkanaligen lokalen Analoggeräten (E/A, RTD und Thermoelement)

Der Diagnosefehler-OB kann nur die Diagnosefehler jeweils eines Kanals verarbeiten.

Wenn in zwei Kanälen eines mehrkanaligen Geräts Fehler auftreten, löst der zweite Fehler den Diagnosefehler-OB nur unter folgenden Bedingungen aus: Der Fehler des ersten Kanals wird behoben, die vom ersten Fehler ausgelöste Ausführung des Diagnosefehler-OBs ist beendet und der zweite Fehler liegt weiterhin vor.

Der Diagnosefehler-OB enthält Anlaufinformationen, anhand derer Sie ermitteln können, ob das Ereignis wegen des Auftretens oder Verschwindens eines Fehlers ausgelöst wurde und welches Gerät und welcher Kanal den Fehler gemeldet haben. Sie können im Diagnosefehler-OB Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Hinweis

Die Anlaufinformationen des Diagnosefehler-OBs beziehen sich auf das Submodul als Ganzes, wenn kein Diagnoseereignis ansteht

In V3.0 gaben die Anlaufinformationen eines gehenden Diagnosefehlerereignisses stets die Quelle des Ereignisses an. In V4.0 beziehen sich die Anlaufinformationen, sofern durch das gehende Ereignis beim Submodul keine Diagnose mehr ansteht, auch dann auf das Submodul als Ganzes (16#8000), wenn es sich bei der Quelle des Ereignisses um einen bestimmten Kanal handelte.

Beispiel: Wenn ein Drahtbruch einen Diagnosefehler an Kanal 2 auslöst, der Fehler dann behoben und das Diagnosefehlerereignis gelöscht wird, beziehen sich die Anlaufinformationen nicht auf Kanal 2, sondern auf das Submodul (16#8000).

Tabelle 5-7 Anlaufinformationen für den Diagnosefehler-OB

Eingang	Datentyp	Beschreibung
I0state	WORD	E/A-Zustand des Geräts: <ul style="list-style-type: none"> • Bit 0 = 1, wenn die Konfiguration korrekt ist, und Bit 0 = 0, wenn die Konfiguration nicht mehr korrekt ist. • Bit 4 = 1, wenn ein Fehler vorliegt (Beispiel: Drahtbruch). (Bit 4 = 0, wenn kein Fehler vorliegt.) • Bit 5 = 1, wenn die Konfiguration nicht korrekt ist, und Bit 5 = 0, wenn die Konfiguration wieder korrekt ist. • Bit 7 = 1, wenn ein E/A-Zugriffsfehler aufgetreten ist. Die Hardwareerkennung der E/A mit dem Zugriffsfehler finden Sie in LADDR. (Bit 6 = 0, wenn kein Fehler vorliegt.)
LADDR	HW_ANY	Hardwareerkennung des Geräts oder der Funktionseinheit, das bzw. die den Fehler gemeldet hat ¹

Eingang	Datentyp	Beschreibung
Channel	UINT	Kanalnummer
MultiError	BOOL	WAHR, wenn mehrere Fehler vorliegen

¹ Der Eingang LADDR enthält die Hardwarekennung des Geräts bzw. der Funktionseinheit, das bzw. die den Fehler ausgegeben hat. Die Hardwarekennung wird automatisch zugewiesen, wenn Komponenten in die Geräte- oder Netzsicht eingefügt werden. Sie wird im Register "Konstanten" von PLC-Variablen angezeigt. Der Hardwarekennung wird zudem automatisch ein Name zugewiesen. Diese Einträge im Register "Konstanten" der PLC-Variablen können nicht geändert werden.

5.1.3.8 OB "Ziehen oder Stecken von Modulen"

Der OB "Ziehen oder Stecken von Modulen" wird ausgeführt, wenn ein konfiguriertes und nicht deaktiviertes dezentrales E/A-Modul oder -Submodul (PROFIBUS, PROFINET, AS-i) ein Ereignis in Bezug auf das Ziehen oder Stecken eines Moduls generiert.

Ereignis "Ziehen oder Stecken von Modulen"

Bei den folgenden Bedingungen wird ein Ereignis "Ziehen oder Stecken von Modulen" generiert:

- Ein Anwender zieht oder steckt ein konfiguriertes Modul
- Ein konfiguriertes Modul ist physisch nicht im Erweiterungsbaugruppenträger vorhanden
- Ein inkompatibles Modul, das nicht dem konfigurierten Modul entspricht, befindet sich in einem Erweiterungsbaugruppenträger
- Ein mit einem konfigurierten Modul kompatibles Modul befindet sich in einem Erweiterungsbaugruppenträger, doch die Konfiguration gestattet keine Ersatzgeräte
- Ein Modul oder Submodul weist Parametrierungsfehler auf

Wurde dieser OB nicht programmiert, bleibt die CPU in RUN, wenn eine dieser Bedingungen für ein konfiguriertes und nicht deaktiviertes dezentrales E/A-Modul eintritt.

Unabhängig davon, ob dieser OB programmiert wurde, wechselt die CPU in STOP, wenn eine dieser Bedingungen für ein Modul im zentralen Baugruppenträger eintritt.

Tabelle 5-8 Anlaufinformationen für den OB "Ziehen oder Stecken von Modulen"

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardwarekennung
Event_Class	Byte	16#38: Modul gesteckt 16#29: Modul gezogen
Fault_ID	Byte	Fehler-ID

5.1.3.9 Baugruppenträger- oder Stationsfehler-OB ("Rack or station failure OB")

Der Baugruppenträger- oder Stationsfehler-OB ("Rack or station failure OB") wird ausgeführt, wenn die CPU einen Fehler oder Kommunikationsausfall an einem dezentralen Baugruppenträger/einer dezentralen Station erkennt.

Baugruppenträger- oder Stationsfehlerereignis

Die CPU erzeugt ein Baugruppenträger- oder Stationsfehlerereignis, wenn eine der folgenden Situationen erkannt wird:

- Ausfall eines DP-Mastersystems oder eines PROFINET IO-Systems (bei einem kommenden oder gehenden Ereignis)
- Ausfall eines DP-Slaves oder eines IO-Geräts (bei einem kommenden oder gehenden Ereignis)
- Ausfall eines Submoduls eines PROFINET I-Device

Ist dieser OB nicht programmiert, bleibt die CPU in RUN, wenn eine dieser Bedingungen eintritt.

Tabelle 5-9 Anlaufinformationen für den Baugruppenträger- oder Stationsfehler-OB

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardware-Kennung
Event_Class	Byte	16#38: gehendes Ereignis 16#39: kommendes Ereignis
Fault_ID	Byte	Fehlerkennung

5.1.3.10 Uhrzeit-OB

Uhrzeit-OBs werden anhand konfigurierter Uhrzeitbedingungen ausgeführt. Die CPU unterstützt zwei Uhrzeit-OBs.

Uhrzeitereignisse

Sie können ein Uhrzeitalarmereignis so konfigurieren, dass es zu einer bestimmten Uhrzeit oder zyklisch auftritt, wobei Sie einen der folgenden Zyklen einstellen können:

- Minütlich: Der Alarm tritt einmal pro Minute auf.
- Stündlich: Der Alarm tritt einmal pro Stunde auf.
- Täglich: Der Alarm tritt täglich zu einer bestimmten Zeit (Stunde und Minute) auf.
- Wöchentlich: Der Alarm tritt wöchentlich zu einer bestimmten Zeit an einem bestimmten Wochentag auf (zum Beispiel jeden Dienstag um 16:30 h).
- Monatlich: Der Alarm tritt monatlich zu einer bestimmten Zeit an einem bestimmten Tag des Monats auf. Für den Tag muss eine Zahl im Bereich von 1 bis 28 festgelegt werden.
- Am Monatsende: Der Alarm tritt am letzten Tag jedes Monats zu einer bestimmten Zeit auf.
- Jährlich: Der Alarm tritt einmal jährlich an einem bestimmten Datum (Monat und Tag) auf. Der 29. Februar kann nicht als Datum eingegeben werden.

Tabelle 5-10 Anlaufinformationen für den Uhrzeit-OB

Eingang	Datentyp	Beschreibung
CaughtUp	Bool	Aufruf des OB wurde abgefangen, weil die Zeit vorgestellt wurde
SecondTimes	Bool	Aufruf des OB startet erneut, weil die Zeit zurückgestellt wurde

5.1.3.11 Zustands-OB

Zustands-OBs werden ausgeführt, wenn ein DPV1- oder PNIO-Slave einen Zustandsalarm auslöst. Dies kann vorkommen, wenn eine Komponente (Modul oder BG-Träger) eines DPV1- oder PNIO-Slaves ihren Betriebszustand, zum Beispiel von RUN nach STOP, ändert.

Zustandsergebnisse

Ausführlichere Informationen zu Ereignissen, die einen Zustandsalarm auslösen, sind in der Herstellerdokumentation des DPV1- oder PNIO-Slaves zu finden.

Tabelle 5-11 Anlaufinformationen für einen Zustands-OB

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardware-Kennung
Slot	UInt	Steckplatznummer
Specifier	Word	Alarm Specifier

5.1.3.12 Aktualisierungs-OB

Aktualisierungs-OBs werden ausgeführt, wenn ein DPV1- oder PNIO-Slave einen Aktualisierungsalarm auslöst.

Aktualisierungsergebnisse

Ausführlichere Informationen zu Ereignissen, die einen Aktualisierungsalarm auslösen, sind in der Herstellerdokumentation des DPV1- oder PNIO-Slaves zu finden.

Tabelle 5-12 Anlaufinformationen für einen Aktualisierungs-OB

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardware-Kennung
Slot	UInt	Steckplatznummer
Specifier	Word	Alarm Specifier

5.1.3.13 Profil-OB

Profil-OBs werden ausgeführt, wenn ein DPV1- oder PNIO-Slave einen profilspezifischen Alarm auslöst.

Profilereignisse

Ausführlichere Informationen zu Ereignissen, die einen Profilalarm auslösen, sind in der Herstellerdokumentation des DPV1- oder PNIO-Slaves zu finden.

Tabelle 5-13 Anlaufinformationen für einen Profil-OB

Eingang	Datentyp	Beschreibung
LADDR	HW_IO	Hardware-Kennung
Slot	UInt	Steckplatznummer
Specifier	Word	Alarm Specifier

5.1.3.14 OB MC-Servo- und MC-Interpolator

STEP 7 erstellt die schreibgeschützten OBs MC-Servo- und MC-Interpolator automatisch, wenn Sie ein Technologieobjekt für die Bewegungssteuerung erstellen und für die Antriebsschnittstelle "Analoge Antriebsanbindung" oder "PROFIdrive" einstellen. Sie brauchen keine OB-Eigenschaften zu ändern und diesen OB auch nicht direkt zu erstellen. Die CPU nutzt diese OBs für die Regelung. Weitere Informationen finden Sie im STEP 7 Informationssystem.

5.1.3.15 MC-PreServo

Sie können den OB MC-PreServo so programmieren, dass er Programmlogik für das STEP 7-Programm enthält. Diese Programmlogik wird dann direkt vor der Bearbeitung des OB MC-Servo ausgeführt.

Ereignisse des MC-PreServo

Mit dem OB MC-PreServo können Sie die konfigurierten Informationen zum Anwendungszyklus innerhalb von Mikrosekunden auslesen.

Tabelle 5-14 Startinformationen für den OB MC-PreServo

Eingang	Datentyp	Beschreibung
Initial_Call	BOOL	WAHR gibt an, dass der erste Aufruf dieses OB beim Wechsel von STOP in RUN erfolgt
PIP_Input	BOOL	WAHR weist darauf hin, dass das zugehörige Prozessabbild der Eingänge auf dem neuesten Stand ist.
PIP_Output	BOOL	WAHR zeigt an, dass die CPU das zugehörige Prozessabbild der Ausgänge nach dem letzten Zyklus in angemessener Zeit an die Ausgänge übertragen hat.
IO_System	USINT	Nummer des dezentralen Peripheriesystems, das den Alarm auslöst
Event_Count	INT	n: Anzahl verlorener Zyklen -1: unbekannte Anzahl verlorener Zyklen (z. B. weil sich der Zyklus geändert hat)
Synchronous	BOOL	Reserviert
CycleTime	UDINT	Anzeige des für den OB MC-Servo konfigurierten Anwendungszyklus in Mikrosekunden

5.1.3.16 MC-PostServo

Sie können den OB MC-PreServo so programmieren, dass er Programmlogik für das STEP 7-Programm enthält. Diese Programmlogik wird dann direkt nach der Bearbeitung des OB MC-Servo ausgeführt.

Ereignisse von MC-PostServo

Mit dem OB MC-PreServo können Sie die konfigurierten Informationen zum Anwendungszyklus innerhalb von Mikrosekunden auslesen.

Tabelle 5-15 Startinformationen für den OB MC-PostServo

Eingang	Datentyp	Beschreibung
Initial_Call	BOOL	WAHR gibt an, dass der erste Aufruf dieses OB beim Wechsel von STOP in RUN erfolgt
PIP_Input	BOOL	WAHR weist darauf hin, dass das zugehörige Prozessabbild der Eingänge auf dem neuesten Stand ist.
PIP_Output	BOOL	WAHR zeigt an, dass die CPU das zugehörige Prozessabbild der Ausgänge nach dem letzten Zyklus in angemessener Zeit an die Ausgänge übertragen hat.
IO_System	USINT	Nummer des dezentralen Peripheriesystems, das den Alarm auslöst
Event_Count	INT	n: Anzahl verlorener Zyklen -1: unbekannte Anzahl verlorener Zyklen (z. B. weil sich der Zyklus geändert hat)
Synchronous	BOOL	Reserviert
CycleTime	UDINT	Anzeige des für den OB MC-Servo konfigurierten Anwendungszyklus in Mikrosekunden

5.1.3.17 Prioritäten und Warteschlange für die Ausführung von Ereignissen

Die CPU-Bearbeitung wird durch Ereignisse gesteuert. Ein Ereignis löst die Ausführung eines Alarm-OBs aus. Sie können während der Erstellung des Bausteins, während der Gerätekonfiguration oder über eine Anweisung ATTACH oder DETACH den Alarm-OB für ein Ereignis angeben. Einige Ereignisse wie das Programmzyklusereignis oder zyklische Ereignisse treten regelmäßig auf. Andere Ereignisse wie das Anlaufereignis oder Zeitverzögerungsereignisse treten einmalig auf. Einige Ereignisse treten auf, wenn die Hardware ein Ereignis auslöst, z. B. ein Flankenereignis an einem Eingang oder ein Ereignis eines schnellen Zählers. Ereignisse wie das Diagnosefehler- und das Zeitfehlerereignis treten nur im Fehlerfall auf. Die Ereignisprioritäten und Warteschlangen dienen zum Festlegen der Verarbeitungsreihenfolge der Alarm-OBs.

Die CPU verarbeitet die Ereignisse in der Reihenfolge ihrer Priorität, wobei 1 die niedrigste Priorität und 26 die höchste Priorität ist. Vor der Version 4.0 der S7-1200 CPUs gehörte jeder OB-Typ zu einer festen Prioritätsklasse (1 bis 26). Ab V4.0 können Sie jedem von Ihnen konfigurierten OB eine Prioritätsklasse zuweisen. Sie konfigurieren die Prioritätsklasse in den OB-Eigenschaften.

Unterbrechbare und nicht unterbrechbare Ausführungsarten

OBs (Seite 75) werden in der Reihenfolge der Priorität der Ereignisse ausgeführt, die die OBs auslösen. In den Anlaufeigenschaften der Gerätekonfiguration der CPU (Seite 153) können Sie konfigurieren, ob die OB-Ausführung unterbrechbar oder nicht unterbrechbar sein soll. Beachten Sie, dass Programmzyklus-OBs immer unterbrechbar sind. Alle anderen OBs können Sie jedoch als unterbrechbar oder als nicht unterbrechbar konfigurieren.

Wenn Sie die unterbrechbare Ausführungsart festlegen, wird die Ausführung eines OBs unterbrochen, falls ein Ereignis mit höherer Priorität vor dem Ausführungsende des OBs auftritt, um die Ausführung des OBs höherer Priorität zu ermöglichen. Daraufhin wird das Ereignis mit höherer Priorität ausgeführt und anschließend der unterbrochene OB fortgesetzt. Treten während der Ausführung eines unterbrechbaren OBs mehrere Ereignisse auf, verarbeitet die CPU diese Ereignisse in der Reihenfolge der Priorität.

Wenn Sie nicht die unterbrechbare Ausführungsart festlegen, wird die Ausführung eines OBs unabhängig von anderen während dieses Zeitraums ausgelösten Ereignissen beendet.

Betrachten Sie die beiden folgenden Fälle, in denen Alarmereignisse einen Zyklus-OB und einen Zeitverzögerungs-OB auslösen. In beiden Fällen ist der Zeitverzögerungs-OB (OB 201) keinem Teilprozessabbild zugewiesen (Seite 67) und er wird mit Priorität 4 ausgeführt. Der Zyklus-OB (OB 200) ist dem Teilprozessabbild TPA1 zugewiesen und wird mit Priorität 2 ausgeführt. Die folgenden Abbildungen zeigen die Unterschiede in der Ausführung zwischen der unterbrechbaren und der nicht unterbrechbaren Ausführungsart:

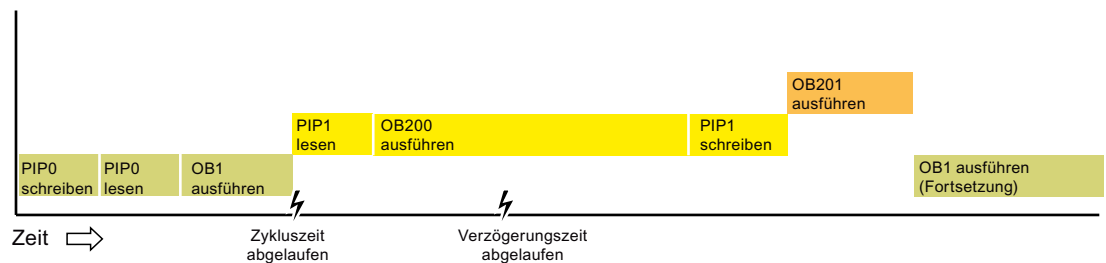


Bild 5-1 Fall 1: Nicht unterbrechbare OB-Ausführung

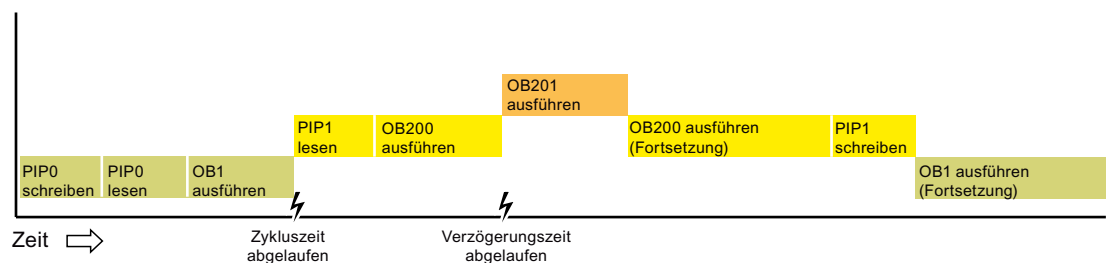


Bild 5-2 Fall 2: Unterbrechbare OB-Ausführung

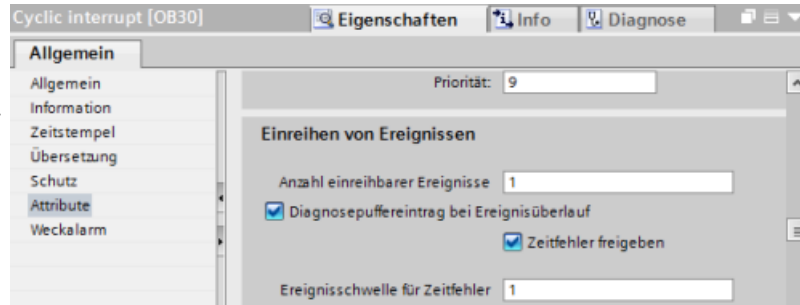
Hinweis

Wenn Sie die OB-Ausführungsart als nicht unterbrechbar festlegen, kann ein Zeitfehler-OB lediglich Programmzyklus-OBs unterbrechen. Vor V4.0 der S7-1200 CPUs konnte ein Zeitfehler-OB jeden ausgeführten OB unterbrechen. Ab V4.0 müssen Sie die OB-Ausführung als unterbrechbar konfigurieren, wenn ein Zeitfehler-OB (oder ein anderer OB höherer Priorität) auch andere ausgeführte OBs und nicht nur Programmzyklus-OBs unterbrechen können soll.

Wissenswertes zu Prioritäten und Warteschlange für die Ausführung von Ereignissen

Die CPU begrenzt die Anzahl anstehender Ereignisse aus einer einzigen Quelle, indem jedem Ereignistyp eine eigene Warteschlange zugewiesen wird. Sobald die maximale Zahl anstehender Ereignisse eines bestimmten Typs erreicht ist, wird das nächste Ereignis nicht mehr bearbeitet und geht verloren. Mit einem Zeitfehleralarm-OB (Seite 79) können Sie auf einen Überlauf der Warteschlange reagieren.

Beachten Sie, dass Sie in STEP 7 für den Weckalarm-OB und den Uhrzeit-OB einige spezifische Parameter für die Ereigniswarteschlange konfigurieren können.



Weitere Informationen zum Überlastverhalten der CPU und zu Ereigniswarteschlangen finden Sie im STEP 7 Informationssystem.

Jedes Ereignis einer CPU hat eine Priorität. Die Ereignisse werden von der CPU im Allgemeinen in der Reihenfolge der Priorität (höchste zuerst) bearbeitet. Ereignisse mit gleicher Priorität werden von der CPU nach dem First-In-First-Out-Prinzip bearbeitet.

Tabelle 5-16 OB-Ereignisse

Ereignis	Zulässige Anzahl	OB-Standardpriorität
Programmzyklus	1 Programmzyklusereignis Mehrere OBs zulässig	1 ¹
Anlauf	1 Anlaufereignis ¹ Mehrere OBs zulässig	1 ¹
Verzögerung	Bis zu 4 Zeitereignisse 1 OB je Ereignis	OB 20: 3 OB 21: 4 OB 22: 5 OB 23: 6 OB 123 bis OB 32767: 3
Weckalarm	Bis zu 4 Ereignisse 1 OB je Ereignis	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 bis OB 32767: 7
Prozessalarm	Bis zu 50 Prozessalarmereignisse ² 1 OB je Ereignis, Sie können jedoch den gleichen OB für mehrere Ereignisse verwenden	18
		18
Zeitfehler	1 Ereignis (nur wenn konfiguriert) ³	22 oder 26 ⁴
Diagnosefehler	1 Ereignis (nur wenn konfiguriert)	5

Ereignis	Zulässige Anzahl	OB-Standardpriorität
Ziehen oder Stecken von Modulen	1 Ereignis	6
Fehler bei Baugruppen-träger oder Station	1 Ereignis	6
Uhrzeitalarm	Bis zu 2 Ereignisse	2
Zustand	1 Ereignis	4
Update	1 Ereignis	4
Profil	1 Ereignis	4
MC-Servo	1 Ereignis	25
MC-Interpolator	1 Ereignis	24

- ¹ Das Anlauf- und das Programmzyklusereignis treten nie gleichzeitig auf, weil der Anlauf zuerst beendet sein muss, bevor der Programmzyklus gestartet wird.
- ² Bei Verwendung der Anweisungen DETACH und ATTACH sind mehr als 50 Prozessalarm-OBs möglich.
- ³ Sie können die CPU so konfigurieren, dass sie in RUN bleibt, wenn die maximale Zykluszeit überschritten wird, oder Sie können mit der Anweisung RE_TRIGR die Zykluszeit zurücksetzen. Wenn die maximale Zykluszeit jedoch in einem Zyklus zum zweiten Mal überschritten wird, geht die CPU in den Betriebszustand STOP.
- ⁴ Die Priorität bei einer neuen CPU V4.0 oder V4.1 ist 22. Wenn Sie eine CPU V3.0 durch eine CPU V4.0 oder V4.1 ersetzen, ist die Priorität 26: also die Priorität, die für V3.0 gültig war. In beiden Fällen kann das Prioritätsfeld geändert werden und Sie können für die Priorität einen beliebigen Wert im Bereich von 22 bis 26 festlegen.

Weitere Informationen dazu finden Sie unter "Ersatz einer CPU V3.0 durch eine CPU V4.x (Seite 1456)".

Außerdem erkennt die CPU andere Ereignisse, die nicht über zugehörige OBs verfügen. Die folgende Tabelle zeigt diese Ereignisse und die entsprechenden Aktionen der CPU:

Tabelle 5-17 Zusätzliche Ereignisse

Ereignis	Beschreibung	CPU-Aktion
Peripheriezugriffsfehler	Fehler beim direkten Lesen/Schreiben der E/A	Die CPU schreibt das erste Auftreten in den Diagnosepuffer und bleibt im Betriebszustand RUN. Die Fehlerursache können Sie mit der Anweisung GET_ERROR_ID (Seite 305) abfragen.
Fehler bezüglich der maximalen Zykluszeit	Die CPU überschreitet die konfigurierte Zykluszeit zum zweiten Mal	Die CPU schreibt den Fehler in den Diagnosepuffer und geht in den Betriebszustand STOP.

Ereignis	Beschreibung	CPU-Aktion
Peripheriezugriffsfehler	E/A-Fehler beim Aktualisieren des Prozessabbilds	Die CPU schreibt das erste Auftreten in den Diagnosepuffer und bleibt im Betriebszustand RUN.
Programmierfehler	Fehler beim Ausführen des Programms	<ul style="list-style-type: none"> • Wenn die bausteinlokale Fehlerbearbeitung aktiviert ist, gibt das System eine Fehlerursache in die Fehlerstruktur ein. Die Fehlerursache können Sie mit der Anweisung GET_ERROR_ID (Seite 305) abfragen. • Wenn die globale Fehlerbearbeitung aktiviert ist, gibt das System ein Zugriffsfehler-Startereignis in den Diagnosepuffer ein und bleibt im Betriebszustand RUN.

Latenzzeit

Die Ereignis-Latenzzeit (d.h. die Zeit zwischen der Mitteilung der CPU über das Auftreten eines Ereignisses und dem Start der Ausführung der ersten Anweisung im OB für die Ereignisbearbeitung) beträgt ca. 175 µs, wenn zum Zeitpunkt des Alarmereignisses nur ein Programmzyklus-OB als Bearbeitungsroutine aktiv ist.

5.1.4 Überwachen und Konfigurieren der Zykluszeit

Die Zykluszeit ist die Zeit, die das Betriebssystem der CPU benötigt, um die zyklische Phase des Betriebszustands RUN auszuführen. Die CPU bietet zwei Verfahren zum Überwachen der Zykluszeit:

- Maximale Zykluszeit
- Mindestzykluszeit

Die Zyklusüberwachung beginnt nach abgeschlossenem Anlaufereignis. Projiziert werden kann diese Funktion in der CPU unter "Gerätekonfiguration > Zykluszeit".

Die CPU überwacht den Zyklus und reagiert, wenn die Zykluszeit die konfigurierte maximale Zykluszeit überschreitet. Die CPU generiert einen Fehler und reagiert wie folgt, wenn die Zykluszeit die konfigurierte maximale Zykluszeit überschreitet:

- Wenn das Anwenderprogramm einen Zeitfehleralarm-OB (Seite 79) enthält, führt die CPU ihn aus.
- Wenn das Anwenderprogramm keinen Zeitfehleralarm-OB enthält, generiert das Zeitfehlerereignis einen Diagnosepuffereintrag. Die CPU geht dann in den Betriebszustand STOP.

Die Anweisung RE_TRIGR (Seite 304) (Zykluszeitüberwachung neu starten) ermöglicht das Zurücksetzen des Zeitgebers für die Messung der Zykluszeit. Wenn die für die aktuelle Programmzyklusausführung abgelaufene Zeit weniger als das Zehnfache der konfigurierten maximalen Zykluszeit beträgt, stößt die Anweisung RE_TRIGR die Zykluszeitüberwachung erneut an und gibt ENO = WAHR zurück. Ist dies nicht der Fall, stößt die Anweisung RE_TRIGR die Zykluszeitüberwachung nicht erneut an. Sie gibt ENO = FALSCH zurück.

Typischerweise wird der Zyklus so schnell wie möglich ausgeführt und der nächste Zyklus beginnt, sobald der vorherige beendet ist. Je nach Anwenderprogramm und Kommunikationsaufgaben kann die Zykluszeit jedoch variieren. Um solche Schwankungen zu vermeiden, unterstützt die CPU eine optionale Mindestzykluszeit. Wenn Sie diese optionale Funktion aktivieren und eine Mindestzykluszeit in ms vorgeben, verzögert die CPU nach der Ausführung der Programmzyklus-OBs die Wiederholung des Programmzyklus, bis die Mindestzykluszeit abgelaufen ist.

Führt die CPU einen Zyklus schneller aus als mit der Mindestzykluszeit festgelegt ist, so nutzt die CPU die verbleibende Zeit für Laufzeitdiagnosen und/oder für die Bearbeitung von Kommunikationsanforderungen.

Wird der Zyklus nicht innerhalb der vorgegebenen Mindestzykluszeit beendet, so wird er normal bis zum Ende ausgeführt (einschließlich der Kommunikationsbearbeitung) und die Überschreitung der Mindestzeit erzeugt keine Systemreaktion. In der folgenden Tabelle sind die Bereiche und Voreinstellungen für die Zykluszeitüberwachung angegeben.

Tabelle 5-18 Bereich für die Zykluszeit

Zykluszeit	Bereich (ms)	Voreinstellung
Maximale Zykluszeit ¹	1 bis 6000	150 ms
Mindestzykluszeit ²	1 bis maximale Zykluszeit	Deaktiviert

¹ Die maximale Zykluszeit ist immer aktiviert. Richten Sie eine Zykluszeit zwischen 1 ms und 6000 ms ein. Die Voreinstellung ist 150 ms.

² Die Mindestzykluszeit ist optional und standardmäßig deaktiviert. Richten Sie ggf. eine Zykluszeit zwischen 1 ms und der maximalen Zykluszeit ein.

Konfigurieren von Zykluszeit und Kommunikationslast

In den CPU-Eigenschaften der Gerätekonfiguration können Sie die folgenden Parameter einstellen:

- **Zyklus:** Hier können Sie eine maximale Zyklusüberwachungszeit eingeben. Sie können auch eine Mindestzykluszeit eingeben bzw. aktivieren.

The screenshot shows a configuration window titled 'Zyklus'. It contains three input fields: 'Zyklusüberwachungszeit' with the value '150' and unit 'ms', a checkbox labeled 'Mindestzykluszeit für zyklische OBs aktivieren' which is currently unchecked, and 'Mindestzykluszeit' with the value '1' and unit 'ms'.

- **Kommunikationslast:** Sie können einen prozentualen Anteil der Zeit für Kommunikationsaufgaben festlegen.

The screenshot shows a configuration window titled 'Kommunikationslast'. It contains one input field: 'Zyklusbelastung durch Kommunikation' with the value '20' and unit '%'.

Hinweis

Kommunikationspriorität

Kommunikationsaufgaben haben die Priorität 1. Da 1 die niedrigste Priorität ist, können andere CPU-Ereignisse die Verarbeitung der Kommunikation unterbrechen. Unterbrechungen von anderen Ereignissen können sich negativ auf die Kommunikationsverarbeitung während des Zyklus auswirken. Sie können den Prozentwert für die "Zyklusbelastung durch Kommunikation" anpassen, um den Anteil des Zyklus, der für die Kommunikationsverarbeitung bereitgestellt wird, zu erhöhen.

Ausführliche Informationen über den Zyklus finden Sie unter "Überwachung des Zyklus" (Seite 90).

5.1.5 CPU-Speicher

Speicherverwaltung

Die CPU stellt die folgenden Speicherbereiche für Anwenderprogramm, Daten und Konfiguration bereit:

- Der Ladespeicher ist ein nicht-flüchtiger Speicher für Anwenderprogramm, Daten und Konfiguration. Beim Laden eines Projekts in die CPU speichert die CPU das Programm zunächst im Ladespeicher. Dieser Speicher befindet sich entweder auf einer Memory Card (sofern vorhanden) oder in der CPU. Die CPU speichert diesen nicht-flüchtigen Speicherbereich auch bei einem Spannungsausfall. Die Memory Card unterstützt einen größeren Speicherbereich als den in der CPU integrierten.
- Der Arbeitsspeicher ist ein flüchtiger Speicher für einige Elemente des Anwenderprojekts während der Bearbeitung des Anwenderprogramms. Die CPU kopiert einige Elemente des Projekts vom Ladespeicher in den Arbeitsspeicher. Dieser flüchtige Speicherbereich geht bei Spannungsausfall verloren und wird bei Spannungsrückkehr von der CPU wiederhergestellt.
- Der remanente Speicher ist ein nicht-flüchtiger Speicher für eine begrenzte Menge an Arbeitsspeicherwerten. Die CPU nutzt den remanenten Speicherbereich zum Speichern der Werte ausgewählter Adressen des Anwenderspeichers bei Spannungsausfall. Wenn eine Abschaltung oder ein Spannungsausfall auftritt, stellt die CPU diese remanenten Werte beim Anlauf wieder her.

Zur Anzeige der Speicherauslastung durch einen übersetzten Programmbaustein doppelklicken Sie auf den Baustein im Ordner "Programmbausteine" der STEP 7-Projektnavigation und wählen im Kontextmenü die Option "Eigenschaften". In den Übersetzungseigenschaften werden der Ladespeicher und der Arbeitsspeicher für den übersetzten Baustein angezeigt.

Um die Speicherauslastung der Online-CPU anzuzeigen, doppelklicken Sie in STEP 7 auf "Online & Diagnose", erweitern "Diagnose" und wählen "Speicher".

Remanenter Speicher

Sie können Datenverlust nach Spannungsausfall dadurch vermeiden, dass Sie bestimmte Daten als remanent definieren. Die folgenden Daten können Sie in der CPU als remanent definieren:

- **Merker (M):** Sie können die Größe des remanenten Speichers für Merker in der PLC-Variablen-tabelle oder in der Zuweisungsliste definieren. Der remanente Merkerspeicher beginnt immer an MBO und läuft über die angegebene Anzahl von Bytes ununterbrochen weiter. Geben Sie diesen Wert in der PLC-Variablen-tabelle oder in der Zuweisungsliste an, indem Sie in der Funktionsleiste auf das Symbol "Remanent" klicken. Geben Sie die Anzahl der Merkerbytes an, die ab MBO remanent sein sollen.
Hinweis: Für jeden Baustein kann die Zuweisungsliste durch Auswahl eines Bausteins im Ordner Programm-bausteine und den Menübefehl **Werkzeuge > Belegungsplan** angezeigt werden.
- **Variablen eines Funktionsbausteins (FB):** Wenn ein FB vom Typ "Optimierter Baustein-zugriff" ist, enthält der Schnittstelleneditor für diesen FB eine Spalte "Remanent". In dieser Spalte können Sie für jede Variable einzeln entweder "Remanent", "Nicht remanent" oder "Im IDB setzen" auswählen. Wenn Sie einen solchen FB im Programm anordnen, enthält der dem FB entsprechende Instanz-DB ebenfalls die Spalte "Remanent". Sie können den remanenten Zustand einer Variable im Schnittstelleneditor des Instanz-DBs nur ändern, wenn Sie im optimierten FB im Bereich "Remanent" für die Variable die Option "Im IDB setzen" (Im Instanzdatenbaustein setzen) ausgewählt haben.
Wenn ein FB **nicht** vom Typ "Optimierter Baustein-zugriff" ist, enthält der Schnittstelleneditor für diesen FB keine Spalte "Remanent". Wenn Sie einen solchen FB im Programm anordnen, enthält der dem FB entsprechende Instanz-DB jedoch eine Spalte "Remanent", die bearbeitet werden kann. In diesem Fall wirkt sich die Auswahl der Option "Remanent" für eine Variable so aus, dass **alle** Variablen ausgewählt werden. Ebenso wirkt sich die Deaktivierung der Option "Remanent" für eine Variable so aus, dass **alle** Variablen deaktiviert werden.
Um anzuzeigen, ob ein FB über optimierten Baustein-zugriff verfügt, öffnen Sie die Eigenschaften des FB und wählen die Attribute aus.
- **Variablen eines globalen Datenbausteins:** Wenn Sie "Optimierter Baustein-zugriff" für die Attribute in den Eigenschaften des Datenbausteins aktivieren, können Sie für jede Variable einstellen, ob sie remanent sein soll oder nicht. Wenn Sie "Optimierter Baustein-zugriff" nicht aktivieren, haben alle Datenbausteinvariablen den gleichen Zustand. Die Variablen sind entweder alle remanent oder alle nicht remanent.

Die CPU unterstützt insgesamt 14336 Byte remanenter Daten für ein V4.5-Projekt, das in eine PLC mit Firmware V4.5 geladen ist. Ist ein V4.4-Projekt geladen, betragen die remanenten Daten 10240 Byte. Um zu ermitteln, wieviele Bytes verfügbar sind, klicken Sie in der PLC-Variablen-tabelle oder in der Zuweisungsliste in der Funktionsleiste auf das Symbol "Remanent". Hier geben Sie zwar den remanenten Bereich für den Merkerspeicher an, doch die zweite Zeile gibt den verbleibenden Gesamtspeicher für M und DB zusammen an. Beachten Sie, dass Sie alle Datenbausteine mit remanenten Variablen übersetzen müssen, damit dieser Wert korrekt ist.

Hinweis


Durch das Laden eines Programms werden die Werte im remanenten Speicher weder gelöscht noch verändert. Soll der remanente Speicher vor dem Laden gelöscht werden, ist vor dem Laden des Programms die CPU auf die Werkseinstellungen zurückzusetzen.

5.1.5.1 System- und Taktmerker

In den CPU-Eigenschaften können Sie Bytes für "Systemspeicher" und "Taktmerker" aktivieren. Die einzelnen Bits dieser Funktionen können in Ihrer Programmlogik über die Variablennamen referenziert werden.

- Sie können im Merkerbereich ein Byte als Systemspeicher zuweisen. Das Byte des Systemmerkers bietet die folgenden vier Bits, die von Ihrem Anwenderprogramm über die folgenden Variablennamen referenziert werden können:
 - Erster Zyklus: Das Bit (Variablenname "FirstScan") wird nach dem Ende des Anlauf-OBs für den ersten Zyklus auf 1 gesetzt. (Nach dem Ende des ersten Zyklus wird das Bit "Erster Zyklus" auf 0 gesetzt.)
 - Diagnosezustand geändert: (Variablenname: "DiagStatusUpdate") wird einen Zyklus lang auf 1 gesetzt, nachdem die CPU ein Diagnoseereignis erfasst hat. Weil die CPU das Bit "DiagStatusUpdate" erst am Ende der ersten Ausführung des Programmzyklus-OBs setzt, kann Ihr Anwenderprogramm nicht erkennen, ob während der Ausführung des Anlauf-OBs oder während der ersten Ausführung des Programmzyklus-OBs eine Diagnoseänderung auftrat.
 - Immer 1 (high): Das Bit (Variablenname "AlwaysTRUE") ist immer auf 1.
 - Immer 0 (low): Das Bit (Variablenname "AlwaysFALSE") ist immer auf 0.
- Sie können im Merkerbereich ein Byte als Taktmerker zuweisen. Jedes Bit dieses als Taktmerker konfigurierten Bytes erzeugt einen Rechteckimpuls. Das Byte des Taktmerkers bietet 8 verschiedene Frequenzen, von 0,5 Hz (langsam) bis 10 Hz (schnell). Sie können diese Bits, besonders in Verbindung mit Flankenweisungen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwenderprogramm verwenden.

Die CPU initialisiert diese Bytes beim Wechsel von STOP in STARTUP. Die Bits des Taktmerkers wechseln während der Betriebsarten STARTUP und RUN synchron zum CPU-Takt.

 **VORSICHT**

Gefahren beim Überschreiben der Bits von Systemspeicher oder Taktmerker

Ein Überschreiben der Bits von Systemspeicher oder Taktmerker kann die Daten in diesen Funktionen beschädigen und bewirken, dass Ihr Anwenderprogramm fehlerhaft arbeitet, was zu Sachschaden und Verletzungen des Personals führen kann.

Weil Taktmerker und Systemspeicher nicht reservierter Speicher im Bereich der Merker sind, können Anweisungen und Kommunikation in diese Adressen schreiben und Daten beschädigen.

Vermeiden Sie das Schreiben von Daten in diese Adressen, um sicherzustellen, dass diese Funktionen einwandfrei arbeiten, und implementieren Sie immer eine Not-Aus-Schaltung für Ihren Prozess oder Ihre Maschine.

Der Systemspeicher konfiguriert ein Byte mit Bits, die bei einem spezifischen Ereignis eingeschaltet werden (Wert = 1).

Systemmerkerbits

Verwendung des Systemmerkerbytes aktivieren

Adresse des Systemmerkerbytes (MBx):

Erster Zyklus:

Diagnosestatus geändert:

Immer 1 (high):

Immer 0 (low):

Tabelle 5-19 Systemspeicher

7	6	5	4	3	2	1	0
Reserviert Wert 0				Immer ausgeschaltet Wert 0	Immer eingeschaltet Wert 1	Diagnosestatusanzeige <ul style="list-style-type: none"> • 1: Änderung • 0: Keine Änderung 	Anzeige erster Zyklus <ul style="list-style-type: none"> • 1: Erster Zyklus nach Anlauf • 0: Nicht der erste Zyklus

Der Taktmerker konfiguriert ein Byte, das die einzelnen Bits in bestimmten Abständen ein- und ausschaltet. Jeder Taktmerker erzeugt einen Rechteckimpuls im entsprechenden Merkerspeicher M. Diese Bits können, vor allem in Verbindung mit Flankenweisungen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwendercode verwendet werden.

Taktmerkerbits

Verwendung des Taktmerkerbytes aktivieren

Adresse des Taktmerkerbytes (MBx):

Takt 10 Hz:

Takt 5 Hz:

Takt 2,5 Hz:

Takt 2 Hz:

Takt 1,25 Hz:

Takt 1 Hz:

Takt 0,625 Hz:

Takt 0,5 Hz:

Tabelle 5-20 Taktmerker

Bitnummer	7	6	5	4	3	2	1	0
Variablenname								
Period(en)	2.0	1.6	1.0	0.8	0.5	0.4	0.2	0.1
Frequenz (Hz)	0.5	0.625	1	1.25	2	2.5	5	10

Weil der Taktmerker asynchron zum CPU-Zyklus läuft, kann sich der Zustand des Taktmerkers während eines langen Zyklus mehrere Male ändern.

5.1.6 Diagnosepuffer

Die CPU unterstützt einen Diagnosepuffer, der für jedes Diagnoseereignis einen Eintrag enthält. Jeder Eintrag umfasst das Datum und die Uhrzeit, zu denen das Ereignis aufgetreten ist, eine Ereigniskategorie und eine Ereignisbeschreibung. Die Einträge werden in chronologischer Reihenfolge angezeigt, wobei das jüngste Ereignis an oberster Stelle steht. In diesem Puffer werden bis zu 50 aktuelle Ereignisse gespeichert. Ist der Puffer voll, wird immer das älteste Ereignis durch ein neues Ereignis überschrieben. Bei einer Unterbrechung der Stromversorgung werden die Ereignisse gespeichert.

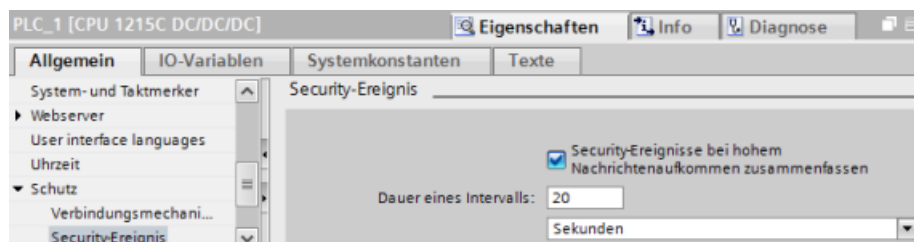
Im Diagnosepuffer werden die folgenden Ereignistypen erfasst:

- Jedes Diagnoseereignis des Systems, CPU-Fehler und Modulfehler
- Jeder Zustandswechsel der CPU (Anlauf, Wechsel in Betriebszustand STOP oder RUN)

Für den Zugriff auf den Diagnosepuffer (Seite 1219) müssen Sie online sein. Den Puffer finden Sie unter "Online & Diagnose > Diagnose > Diagnosepuffer".

Reduzieren der Anzahl der Sicherheitsdiagnoseereignisse

Einige Sicherheitsereignisse erzeugen wiederholte Einträge im Diagnosepuffer. Diese Meldungen können den Diagnosepuffer füllen und möglicherweise andere Meldungen verdecken. Sie können den PLC so konfigurieren, dass die Anzahl der durch Sicherheitsereignisse ausgelösten Diagnosemeldungen begrenzt wird. Sie treffen Ihre Auswahlen in der Gerätekonfiguration der CPU basierend auf dem Zeitintervall, während dessen Dauer Sie wiederholt auftretende Meldungen unterdrücken möchten:



Wenn Sie auswählen, dass Sicherheitsereignisse innerhalb eines Zeitintervalls zusammengefasst werden sollen, können Sie ein Zeitintervall in Sekunden, Minuten oder Stunden und einen numerischen Wert im Bereich von 1 bis 255 eingeben.

Wenn Sie auswählen, dass Sicherheitsereignisse eingeschränkt werden sollen, werden die folgenden Ereignistypen eingeschränkt:

- Online gehen mit dem richtigen oder falschen Passwort
- Geänderte Kommunikationsdaten erkannt
- Geänderte Daten auf Memory Card erkannt
- Geänderte Firmware-Aktualisierungsdatei erkannt
- In CPU geladene Schutzstufe (Zugriffsschutz) geändert
- Passwortberechtigung eingeschränkt oder aktiviert (mittels Anweisung oder über CPU-Anzeige)
- Online-Zugriff verweigert aufgrund einer Überschreitung der möglichen Anzahl gleichzeitiger Zugriffsversuche

- Zeitablauf, wenn eine vorhandene Online-Verbindung inaktiv ist
- Anmelden beim Webserver mit dem richtigen oder falschen Passwort
- Erstellen einer Sicherungskopie der CPU
- Wiederherstellen der CPU-Konfiguration

5.1.7 Echtzeituhr

Die CPU verfügt über eine Echtzeituhr. Wenn die CPU ausgeschaltet ist, wird die Uhr über einen Hochleistungskondensator mit Spannung versorgt. Der Hochleistungskondensator wird aufgeladen, wenn die CPU eingeschaltet ist. War die CPU mindestens 24 Stunden eingeschaltet, so reicht die Ladung des Hochleistungskondensators für den Betrieb der Uhr typischerweise für 20 Tage.

STEP 7 stellt für die Echtzeituhr die Systemzeit ein, die einen werkseitig voreingestellten Standardwert hat bzw. den Standardwert durch Rücksetzen auf die Werkseinstellungen erhält. Um die Echtzeituhr zu nutzen, müssen Sie sie einstellen. Zeitstempel wie die von Diagnosepuffereinträgen, Datenprotokolldateien und Datenprotokolleinträgen beruhen auf der Systemzeit. Sie stellen die Echtzeituhr über die Funktion "Uhrzeit einstellen" (Seite 1212) in der Ansicht "Online & Diagnose" der Online-CPU ein. STEP 7 berechnet dann die Systemzeit anhand der von Ihnen eingestellten Zeit plus oder minus dem Zeitunterschied des Windows-Betriebssystems zur UTC (Coordinated Universal Time). Wird für die Echtzeituhr die Ortszeit eingestellt, erzeugt dies eine Systemzeit, die der UTC entspricht, sofern die Einstellungen für Zeitzone und Sommer-/Winterzeitumschaltung Ihres Windows-Betriebssystems Ihrem Gebietsschema entsprechen.

STEP 7 enthält Anweisungen (Seite 330) zum Lesen und Schreiben der Systemzeit (RD_SYS_T und WR_SYS_T), zum Lesen der Ortszeit (RD_LOC_T) und zum Einstellen der Zeitzone (SET_TIMEZONE). Die Anweisung RD_LOC_T berechnet die Ortszeit anhand der Zeitzone und der Zeitverschiebung für die Sommer-/Winterzeit, die Sie in der Konfiguration der Echtzeituhr in den allgemeinen Eigenschaften der CPU (Seite 153) eingegeben haben. Über diese Einstellungen können Sie die Zeitzone für Ihre Ortszeit einrichten, Sie können optional die Sommerzeitumschaltung aktivieren und die Anfangs- und Endezeiten für die Sommer-/Winterzeitumschaltung eingeben. Diese Einstellungen können Sie auch mit den Anweisungen SET_TIMEZONE konfigurieren.

5.1.8 Konfigurieren der Ausgänge für den Wechsel von RUN in STOP

Sie können das Verhalten der digitalen und analogen Ausgänge für den Betriebszustand STOP der CPU konfigurieren. Für jeden Ausgang einer CPU, eines SB oder SM können Sie die Werte der Ausgänge einfrieren oder einen Ersatzwert aufschalten:

- Durch einen vorgegebenen Ausgangswert ersetzen (Voreinstellung): Sie geben für jeden Ausgang (Kanal) der CPU, des SB oder SM einen Ersatzwert ein. Der voreingestellte Ersatzwert für digitale Ausgänge ist AUS, für analoge Ausgänge ist er 0.
- Einfrieren der Ausgänge im letzten Zustand: Die Ausgänge behalten ihre aktuellen Werte, die sie zu der Zeit hatten, als der Wechsel von RUN nach STOP auftrat. Nach dem Anlauf werden die Ausgänge auf den Standardersatzwert gesetzt.

Sie konfigurieren das Verhalten der Ausgänge in der Gerätekonfiguration. Wählen Sie die einzelnen Geräte und öffnen Sie das Register "Eigenschaften", um die Ausgänge des jeweiligen Geräts zu konfigurieren.

Hinweis

Einige dezentrale E/A-Module bieten zusätzliche Einstellungen für die Reaktion auf die Betriebsart STOP der CPU. Wählen Sie für diese Module die gewünschte Option aus der Liste in der Gerätekonfiguration aus.

Wenn die CPU von RUN nach STOP wechselt, speichert die CPU das Prozessabbild und schreibt die entsprechenden Werte für digitale und analoge Ausgänge entsprechend der Konfiguration.

5.2 Datenspeicher, Speicherbereiche, E/A und Adressierung

5.2.1 Zugriff auf die Daten der S7-1200

STEP 7 vereinfacht die symbolische Programmierung. Dazu erstellen Sie für die Adressen der Daten symbolische Namen oder "Variablen", die entweder in Form von PLC-Variablen für Speicheradressen und E/A oder in Form von lokalen Variablen innerhalb eines Codebausteins vorkommen. Zum Einfügen dieser Variablen in Ihr Anwenderprogramm geben Sie einfach den Variablennamen für den gewünschten Anweisungsparameter ein.

Zur Verdeutlichung, wie die CPU Speicherbereiche strukturiert und adressiert, wird im Folgenden dargestellt, wie PLC-Variablen auf die "absolute" Adressierung der Daten verweisen. Die CPU bietet mehrere Möglichkeiten für die Datenspeicherung während der Ausführung des Anwenderprogramms:

- **Globaler Speicher:** Die CPU bietet eine Vielzahl von spezialisierten Speicherbereichen, einschließlich Eingänge (E), Ausgänge (A) und Merker (M). Dieser Speicher ist für alle Codebausteine ohne Einschränkung zugänglich.
- **PLC-Variablen-tabelle:** In der STEP 7-PLC-Variablen-tabelle können Sie für bestimmte Adressen im Speicher symbolische Namen eingeben. Diese Variablen gelten global im gesamten STEP 7-Programm und gestatten die Programmierung mit aussagekräftigen Namen.
- **Datenbaustein (DB):** Sie können in Ihr Anwenderprogramm DBs zum Speichern von Daten für die Codebausteine einfügen. Die gespeicherten Daten bleiben nach der Ausführung des zugehörigen Codebausteins erhalten. In einem "globalen" DB werden Daten gespeichert, die von allen Codebausteinen verwendet werden können, in einem Instanz-DB werden jedoch nur Daten für einen bestimmten FB gespeichert, und er ist entsprechend der Parameter des FBs strukturiert.
- **Temporärer Speicher:** Bei jedem Aufruf eines Codebausteins gibt das Betriebssystem der CPU temporären bzw. lokalen Speicherplatz (L) frei, der bei der Ausführung des Bausteins genutzt werden kann. Ist die Ausführung des Codebausteins beendet, weist die CPU den lokalen Speicher für die Ausführung anderer Codebausteine zu.

Jeder Speicherplatz hat eine eindeutige Adresse. Anhand dieser Adresse kann Ihr Anwenderprogramm auf die Informationen an diesem Speicherplatz zugreifen. Durch Verweise auf Speicherbereiche für Eingänge (E) oder Ausgänge (A), z. B. E0.3 oder A1.7, erfolgt ein

Zugriff auf das Prozessbild. Für den direkten Zugriff auf den physischen Eingang oder Ausgang fügen Sie den Verweis ":P" (z. B. E0.3:P, A1.7:P oder "Stop:P") als Anhang hinzu.

Tabelle 5-21 Speicherbereiche

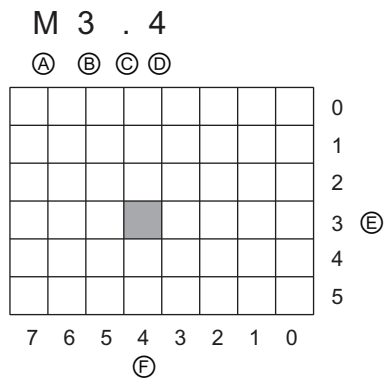
Speicherbereich	Beschreibung	Forcen	Remanent
E Prozessabbild der Eingänge E_:P ¹ (physischer Eingang)	Wird zu Beginn des Zyklus aus den physischen Eingängen kopiert	Nein	Nein
	Direktes Lesen der physischen Eingänge von CPU, SB oder SM	Ja	Nein
A Prozessabbild der Ausgänge A_:P ¹ (physischer Ausgang)	Wird zu Beginn des Zyklus in die physischen Ausgänge kopiert	Nein	Nein
	Direktes Schreiben in die physischen Ausgänge von CPU, SB oder SM	Ja	Nein
M Merker	Steuerung und Datenspeicher	Nein	Ja (optional)
L temporärer Speicher	Temporäre, lokale Daten für einen Baustein	Nein	Nein
DB Datenbaustein	Datenspeicher und auch Parameterspeicher für FBs	Nein	Ja (optional)

¹ Für den direkten Zugriff (Lesen oder Schreiben) auf die physischen Eingänge und Ausgänge hängen Sie den Code ":P" an die Adresse oder Variable an (z. B. E0.3:P, A1.7:P oder "Stop:P").

Jeder Speicherplatz hat eine eindeutige Adresse. Anhand dieser Adresse kann Ihr Anwenderprogramm auf die Informationen an diesem Speicherplatz zugreifen. Die absolute Adresse setzt sich aus den folgenden Elementen zusammen:

- Speicherbereichskennung (wie E, A oder M)
- Größe der Daten, auf die zugegriffen werden soll (wie "B" für Byte, "W" für Word oder "D" für DWord)
- Anfangsadresse der Daten (wie Byte 3 oder Wort 3)

Beim Zugriff auf ein Bit in der Adresse eines Booleschen Werts geben Sie keine Mnemonik für die Größe ein. Sie geben nur den Speicherbereich, die Byte-Adresse und die Bitadresse der Daten ein (wie E0.0, A0.1 oder M3.4).



- A Speicherbereichskennung
- B Adresse des Byte: Byte 3
- C Trennzeichen ("Byte.Bit")
- D Bitadresse im Byte (Bit 4 von 8)
- E Bytes des Speicherbereichs
- F Bits des ausgewählten Byte

In dem Beispiel folgt auf den Speicherbereich und die Adresse des Bytes (M = Bereich der Merker und 3 = Byte 3) ein Punkt ("."), um die Adresse des Bits (Bit 4) abzutrennen.

Zugriff auf Daten in den Speicherbereichen der CPU

STEP 7 vereinfacht die symbolische Programmierung. Typischerweise werden Variablen entweder in der PLC-Variablentabelle, einem Datenbaustein oder in der Schnittstelle eines OB, FC oder FB angelegt. Diese Variablen umfassen einen Namen, Datentyp, Versatz und Kommentar. Außerdem kann in einem Datenbaustein ein Startwert angegeben werden. Diese Variablen können Sie während der Programmierung nutzen, indem Sie den Variablennamen als Parameter für die Anweisung eingeben. Optional können Sie den absoluten Operanden (Speicher, Bereich, Größe und Versatz) als Anweisungsparameter eingeben. Die Beispiele in den folgenden Abschnitten zeigen, wie Sie absolute Operanden eingeben. Das Zeichen % wird vom Programmiereditor automatisch vor dem absoluten Operanden eingefügt. Im Programmiereditor haben Sie die Auswahl zwischen folgenden Ansichten: symbolisch, symbolisch und absolut oder absolut.

E (Prozessabbild der Eingänge): Die CPU fragt die Peripherieeingänge (physikalischen Eingänge) unmittelbar vor der Ausführung eines Zyklus-OB in jedem Zyklus ab und schreibt diese Werte in das Prozessabbild der Eingänge. Auf das Prozessabbild der Eingänge können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Es ist sowohl der Lese- als auch der Schreibzugriff erlaubt, jedoch werden die Eingänge des Prozessabbilds typischerweise nur gelesen.

Tabelle 5-22 Absolute Adressierung für den Speicherbereich E

Bit	E[Byteadresse].[Bitadresse]	E0.1
Byte, Wort oder Doppelwort	E[Größe][Adresse des Anfangsbyte]	EB4, EW5 oder ED12

Wenn Sie den Verweis ":P" an die Adresse anhängen, können Sie die digitalen und analogen Eingänge von CPU, SB, SM oder des dezentralen Moduls direkt lesen. Der Unterschied zwischen dem Zugriff über E_:P statt über E liegt darin, dass die Daten direkt von den angesprochenen Eingängen kommen und nicht aus dem Prozessabbild der Eingänge. Dieser Zugriff über E_:P wird auch als direkter Lesezugriff bezeichnet, weil die Daten direkt aus der Quelle statt aus einer

Kopie, die bei der letzten Aktualisierung des Prozessabbilds der Eingänge erstellt wurde, gelesen werden.

Weil die physikalischen Eingänge ihre Werte direkt aus den mit ihnen verbundenen Feldgeräten erhalten, darf in diese Eingänge nicht geschrieben werden. Zugriffe über E_:P sind somit reine Lesezugriffe, im Gegensatz zu Zugriffen auf Eingänge, die gelesen oder geschrieben werden können.

Zugriffe über E_:P sind ferner durch die Größe der Eingänge begrenzt, die von CPU, SB oder SM unterstützt werden (gerundet auf das nächste Byte). Sind beispielweise die Eingänge eines SB mit 2 DE/2 DA so konfiguriert, dass sie bei E4.0 beginnen, kann mit E4.0:P and E4.1:P oder EB4:P darauf zugegriffen werden. Der Zugriff auf E4.2:P bis E4.7:P wird nicht als Fehler erkannt, ist aber sinnlos, weil diese Adressen nicht belegt sind. Zugriffsversuche auf EW4:P und ED4:P sind verboten, weil sie den Byteversatz dieses SB überschreiten.

Zugriffe über E_:P haben keinen Einfluss auf den im Prozessabbild der Eingänge gespeicherten Wert.

Tabelle 5-23 Absolute Adressierung für den Speicherbereich E (direkt)

Bit	E[Byteadresse].[Bitadresse]:P	E0.1:P
Byte, Wort oder Doppelwort	E[Größe][Anfangsadresse des Byte]:P	EB4:P, EW5:P oder ED12:P

A (Prozessabbild der Ausgänge): Die CPU kopiert die im Prozessabbild der Ausgänge gespeicherten Werte in die physischen Ausgänge. Auf das Prozessabbild der Ausgänge können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Auf die Ausgänge des Prozessabbilds ist sowohl der Lese- als auch der Schreibzugriff erlaubt.

Tabelle 5-24 Absolute Adressierung für den Speicherbereich A

Bit	A[Byteadresse].[Bitadresse]	A1.1
Byte, Wort oder Doppelwort	A[Größe][Adresse des Anfangsbyte]	AB5, AW10, AD40

Wenn Sie den Verweis ":P" an die Adresse anhängen, können Sie direkt in die digitalen und analogen Ausgänge von CPU, SB, SM oder des dezentralen Moduls schreiben. Der Unterschied zwischen dem Zugriff über A_:P statt über A liegt darin, dass die Daten direkt zu den angesprochenen Ausgängen gelangen und zusätzlich zum Prozessabbild der Ausgänge (es wird in beide Bereiche geschrieben). Dieser Zugriff über A_:P wird auch als direkter Schreibzugriff bezeichnet, weil die Daten direkt an die Zieladresse gesendet werden und diese nicht auf die nächste Aktualisierung aus dem Prozessabbild der Ausgänge warten muss.

Weil die physikalischen Ausgänge direkt die mit ihnen verbundenen Feldgeräte steuern, dürfen diese Ausgänge nicht gelesen werden. Zugriffe über A_:P sind somit reine Schreibzugriffe, im Gegensatz zu Zugriffen auf Ausgänge, die gelesen oder geschrieben werden können.

Zugriffe über A_:P sind ferner durch die Größe der Ausgänge begrenzt, die von CPU, SB oder SM unterstützt werden (gerundet auf das nächste Byte). Sind beispielweise die Ausgänge eines SB mit 2 DE/2 DA so konfiguriert, dass sie bei A4.0 beginnen, kann mit A4.0:P and A4.1:P oder AB4:P darauf zugegriffen werden. Der Zugriff auf A4.2:P bis A4.7:P wird nicht als Fehler erkannt, ist aber sinnlos, weil diese Adressen nicht belegt sind. Zugriffsversuche auf AW4:P und AD4:P sind verboten, weil sie den Byteversatz dieses SB überschreiten.

Zugriffe über A_:P beeinflussen den physischen Ausgang sowie den im Prozessabbild der Ausgänge gespeicherten Wert.

Tabelle 5-25 Absolute Adressierung für den Speicherbereich A (direkt)

Bit	A[Byteadresse].[Bitadresse]:P	A1.1:P
Byte, Wort oder Doppelwort	A[Größe][Anfangsadresse des Byte]:P	AB5:P, AW10:P oder AD40:P

M (Speicherbereich der Merker): Den Speicherbereich der Merker (M) können Sie für Steuerungsrelais und Daten verwenden, um Zwischenergebnisse von Anweisungen oder andere Steuerungsinformationen zu speichern. Auf den Speicherbereich der Merker können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Auf den Merkerspeicher ist sowohl der Lese- als auch der Schreibzugriff erlaubt.

Tabelle 5-26 Absolute Adressierung für den Speicherbereich M

Bit	M[Byteadresse].[Bitadresse]	M26.7
Byte, Wort oder Doppelwort	M[Größe][Adresse des Anfangsbyte]	MB20, MW30, MD50

Temp (temporärer Speicher): Die CPU ordnet den temporären Speicher nach Bedarf zu. Die CPU ordnet dem Codebaustein den temporären Speicher zu und initialisiert die Speicheradressen auf 0, wenn der Codebaustein gestartet (bei einem OB) bzw. aufgerufen wird (bei einer FC oder einem FB).

Der temporäre Speicher ist dem Merkerspeicher (M) ähnlich. Es gibt jedoch einen wichtigen Unterschied: Der Merkerspeicher hat "globalen" Geltungsbereich, während der temporäre Speicher "lokalen" Geltungsbereich hat:

- Speicherbereich der Merker: Jeder OB, FB und jede FC kann auf die Daten im Speicherbereich der Merker zugreifen, d.h. die Daten stehen global für alle Elemente des Anwenderprogramms zur Verfügung.
- Temporärer Speicher: Die CPU beschränkt den Zugriff auf die Daten des temporären Speichers auf den OB, FB oder die FC, der bzw. die die Adresse im temporären Speicher erstellt bzw. deklariert hat. Adressen im temporären Speicher bleiben lokal und werden nicht von verschiedenen Codebausteinen gemeinsam genutzt, auch dann nicht, wenn der Codebaustein einen anderen Codebaustein aufruft. Beispiel: Wenn ein OB eine FC aufruft, kann die FC nicht auf den temporären Speicher des OBs, von dem sie aufgerufen wurde, zugreifen.

Die CPU stellt für jede OB-Prioritätsklasse temporären (lokalen) Speicher zur Verfügung:

- 16 KB für Anlauf und Programmzyklus, einschließlich der zugehörigen FBs und FCs
- 6 KB für jeden zusätzlichen Alarmereignis-Thread einschließlich der zugehörigen FBs und FCs

Auf den temporären Speicher können Sie nur über die symbolische Adressierung zugreifen.

Über die Aufrufstruktur in STEP 7 können Sie die Menge des temporären Speichers (Lokaldaten) ermitteln, die die Bausteine in Ihrem Programm belegen. Wählen Sie in der Projektnavigation "Programminformationen" und dann das Register "Aufrufstruktur". Dort werden alle OBs in Ihrem Programm angezeigt, und Sie können sich jeweils die Bausteine anzeigen lassen, die von den OBs aufgerufen werden. Für jeden Baustein werden die zugeordneten Lokaldaten angezeigt. Die Aufrufstruktur können Sie auch in STEP 7 über den Menübefehl **Extras > Aufrufstruktur** aufrufen.

DB (Datenbaustein): Nutzen Sie Datenbausteine zum Speichern verschiedener Arten von Daten, auch Zwischenergebnisse einer Anweisung oder andere Steuerungsparameter für FBs und Datenstrukturen für viele Anweisungen wie Zeiten und Zähler. Auf Datenbausteine können Sie im Bit, Byte, Wort und Doppelwortformat zugreifen. Für nicht schreibgeschützte Datenbausteine ist sowohl der Lese- als auch der Schreibzugriff erlaubt. Für schreibgeschützte Datenbausteine ist nur der Lesezugriff erlaubt.

Tabelle 5-27 Absolute Adressierung für den Speicherbereich DB

Bit	DB[Datenbausteinnummer].DBX[Byteadresse].[Bitadresse]	DB1.DBX2.3
Byte, Wort oder Doppelwort	DB[Datenbausteinnummer].DB [Größe].[Anfangsbyteadresse]	DB1.DBB4, DB10.DBW2, DB20.DBD8

Hinweis

Wenn Sie in KOP oder FUP eine absolute Adresse angeben, stellt STEP 7 dieser Adresse das Zeichen "%" voran, um kenntlich zu machen, dass es sich um eine absolute Adresse handelt. Bei der Programmierung können Sie eine absolute Adresse entweder mit dem oder ohne das Zeichen "%" eingeben (Beispiel: %E0.0 oder E.0). Wenn Sie das Zeichen "%" weglassen, fügt STEP 7 es ein.

In SCL muss das Zeichen "%" vor der Adresse eingegeben werden, um kenntlich zu machen, dass es sich um eine absolute Adresse handelt. Fehlt das "%", erzeugt STEP 7 beim Übersetzen einen undefinierten Variablenfehler.

E/A in der CPU und in E/A-Modulen konfigurieren



Baugruppe	Steckp.	E-Adresse	A-Adresse	Typ	Bestel
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7
PLC_1	1			CPU 1214C DCDC	6ES7
DI14/DO10	1.1	0...1	0...1	DI14/DO10	
AI2	1.2	64...67		AI2	
AO1 x 12Bit	1.3		80...81	AO1 Signalboard	6ES7
HSC_1	1.16	1000.....		Schneller Zähler (f)	
HSC_2	1.17			Schneller Zähler (f)	
HSC_3	1.18			Schneller Zähler (f)	
HSC_4	1.19			Schneller Zähler (f)	
HSC_5	1.20			Schneller Zähler (f)	
HSC_6	1.21			Schneller Zähler (f)	
Pulse_1	1.32			Impulsgenerator (f)	
Pulse_2	1.33			Impulsgenerator (f)	
PROFINET...	X1			PROFINET-Schnitts.	
DIB x DC24V...	2		8	SM 1221 DIB x DC	6ES7

Wenn Sie eine CPU und E/A-Module in Ihre Gerätekonfiguration einfügen, weist STEP 7 E- und A-Adressen automatisch zu. Sie können die voreingestellte Adressierung ändern, indem Sie in der Gerätekonfiguration das Adressfeld auswählen und neue Zahlen eingeben.

- Digitale Eingänge und Ausgänge werden in STEP 7 in Gruppen zu 8 Punkten (1 Byte) zugewiesen, unabhängig davon, ob das Modul alle Ein- bzw. Ausgänge (Punkte) nutzt oder nicht.
- Analoge Eingänge und Ausgänge werden in STEP 7 in Gruppen zu je 2 zugewiesen, wobei jeder analoge Punkt 2 Byte (16 Bit) belegt.

Das Bild zeigt ein Beispiel für eine CPU 1214C mit zwei SMs und einem SB. In diesem Beispiel könnte die Adresse von Modul DI8 auf 2 anstelle von 8 geändert werden. Das Tool unterstützt den Anwender durch Änderung von Adressbereichen, die die falsche Größe haben oder Konflikte mit anderen Adressen aufweisen.

5.3 Verarbeitung von Analogwerten

Analoge Signalmodule liefern Eingangssignale oder erwarten Ausgangswerte, die entweder einen Spannungsbereich oder einen Strombereich darstellen. Diese Bereiche sind $\pm 10\text{ V}$, $\pm 5\text{ V}$, $\pm 2,5\text{ V}$ oder $0\text{ bis }20\text{ mA}$. Die von den Modulen ausgegebenen Werte sind ganzzahlige Werte. Dabei stellt $0\text{ bis }27648$ den Nennbereich für Strom und $-27648\text{ bis }27648$ den für Spannung dar. Werte außerhalb dieser Bereiche stellen entweder einen Überlauf oder einen Unterlauf dar. Einzelheiten zu den Wertetypen, die außerhalb des Bereichs liegen, finden Sie in den Tabellen Darstellung Analogeingang (Seite 1358) und Darstellung Analogausgang (Seite 1359).

In Ihrem Steuerungsprogramm müssen Sie diese Werte wahrscheinlich in physikalischen Einheiten verwenden, um beispielsweise Volumen, Temperatur, Gewicht oder einen anderen quantitativen Wert darzustellen. Hierfür müssen Sie den Analogeingang zunächst in eine Realzahl (Gleitpunktwert) zwischen $0,0$ und $1,0$ normieren. Dann müssen Sie den Wert für den Mindest- und Höchstwert der darzustellenden physikalischen Einheit skalieren. Bei Werten in physikalischen Einheiten, die Sie in einen Analogausgangswert umwandeln müssen, normieren Sie zunächst den Wert in physikalischen Einheiten in einen Wert zwischen $0,0$ und $1,0$. Danach skalieren Sie den Wert zwischen 0 und 27648 oder -27648 und 27648 , je nach Bereiche des Analogmoduls. STEP 7 bietet zu diesem Zweck die Anweisungen `NORM_X` und `SCALE_X` (Seite 291). Sie können auch die Anweisung `CALCULATE` (Seite 235) verwenden, um die Analogwerte zu skalieren.

Beispiel: Analogwertverarbeitung

Betrachten wir beispielsweise einen analogen Eingang mit einem Strombereich von $0 - 20\text{ mA}$. Das analoge Eingangsmodul gibt Messwerte im Bereich von $0\text{ bis }27648$ zurück. Bei diesem Beispiel gehen wir davon aus, dass der analoge Eingangswert für die Messung eines Temperaturbereichs von $50\text{ °C bis }100\text{ °C}$ verwendet wird. Einige Beispielwerte hätten dann die folgenden Bedeutungen:

Analogeingangswert	Technische Einheiten
0	50 °C
6192	62,5 °C
12384	75 °C
18576	87,5 °C
27648	100 °C

Die physikalischen Einheiten in diesem Beispiel würden anhand des analogen Eingangswerts wie folgt berechnet:

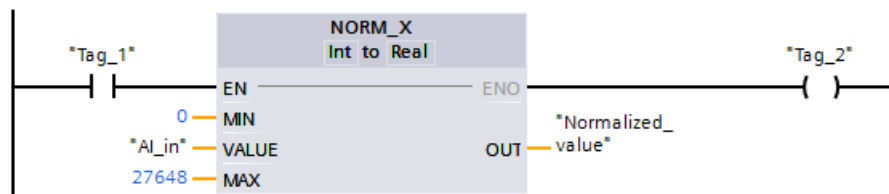
$$\text{Wert in physikalischen Einheiten} = 50 + (\text{Analogeingangswert}) * (100 - 50) / (27648 - 0)$$

Allgemein würde die Gleichung wie folgt aussehen:

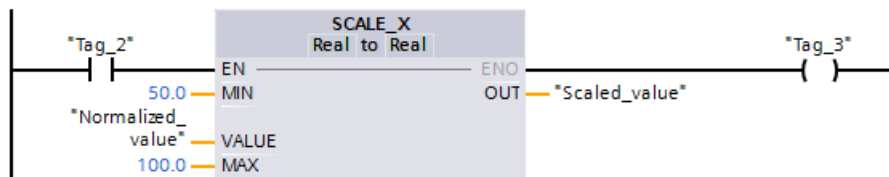
$$\text{Wert in physikalischen Einheiten} = (\text{unterer Bereich der physikalischen Einheiten} + (\text{Analogeingangswert}) * (\text{oberer Bereich der physikalischen Einheiten} - \text{unterer Bereich der physikalischen Einheiten}) / (\text{maximaler Analogeingangsbereich} - \text{minimaler Analogeingangsbereich})$$

In PLC-Anwendungen wird üblicherweise der Analogeingangswert in einen Gleitpunktwert zwischen 0.0 und 1.0 normiert. Das Ergebnis würde dann in einen Gleitpunktwert im Bereich der physikalischen Einheiten umgewandelt. Der Einfachheit halber nutzen die folgenden KOP-Anweisungen konstante Bereichswerte; es können jedoch auch Variablen verwendet werden.

Netzwerk 1



Netzwerk 2



5.4 Datentypen

Datentypen geben die Größe eines Datenelements und die Art der Auswertung der Daten an. Jeder Anweisungsparameter unterstützt mindestens einen Datentyp, einige Parameter unterstützen mehrere Datentypen. Halten Sie den Mauszeiger auf dem Parameterfeld einer Anweisung, damit Ihnen angezeigt wird, welche Datentypen für den jeweiligen Parameter unterstützt werden.

Ein Formalparameter ist die Kennung an einer Anweisung, die die Adresse der von der Anweisung zu verwendenden Daten angibt (Beispiel: Eingang IN1 einer Anweisung ADD). Ein Aktualparameter ist die Adresse (mit vorangestelltem Zeichen "%") oder Konstante, die die von der Anweisung zu verwendenden Daten enthält (Beispiel: %MD400 "Anzahl_Widgets"). Der Datentyp des von Ihnen angegebenen Aktualparameters muss einem der von der Anweisung angegebenen unterstützten Datentypen des Formalparameters entsprechen.

Wenn Sie einen Aktualparameter angeben, müssen Sie entweder eine Variable (Symbol) oder eine absolute (direkte) Adresse angeben. Variablen weisen einem symbolischen Namen (Variablennamen) einen Datentyp, Speicherbereich, Speicherversatz und Kommentar zu und können entweder im PLC-Variableneditor oder im Schnittstelleneditor eines Bausteins (OB, FC, FB und DB) angegeben werden. Wenn Sie eine absolute Adresse eingeben, die keine

zugewiesene Variable hat, müssen Sie eine Größe verwenden, die einem unterstützten Datentyp entspricht, wird bei der Eingabe eine Standardvariable angelegt.

Alle Datentypen, mit Ausnahme der Datentypen String, Struct, Array und DTL sind im PLC-Variableneditor und in den Bausteinschnittstellen verfügbar. String, Struct, Array und DTL sind nur in den Bausteinschnittstellen verfügbar. Für viele der Eingangsparameter können Sie auch einen konstanten Wert eingeben.

- Bit und Bitfolgen (Seite 107): Bool (Boolescher Wert oder Bitwert), Byte (8-Bit-Bytewert), Word (16-Bit-Wortwert), DWord (32-Bit-Doppelwortwert)
- Ganzzahl (Seite 108)
 - USInt (vorzeichenlose 8-Bit-Ganzzahl), SInt (vorzeichenbehaftete 8-Bit-Ganzzahl),
 - UInt (vorzeichenlose 16-Bit-Ganzzahl), Int (vorzeichenbehaftete 16-Bit-Ganzzahl),
 - UDInt (vorzeichenlose 32-Bit-Ganzzahl), DInt (vorzeichenbehaftete 32-Bit-Ganzzahl),
- Gleitpunktzahl/Realzahl (Seite 108): Real (32-Bit-Real- oder Gleitpunktwert), LReal (64-Bit-Real- oder Gleitpunktwert)
- Uhrzeit und Datum (Seite 109): Time (32-Bit-IEC-Zeitwert), Date (16-Bit-Datumswert), TOD (32-Bit-Uhrzeitwert), DTL (12-Byte-Datums- und-Uhrzeitstruktur)
- Zeichen und Zeichenfolge (Seite 110): Char (8-Bit-Einzelzeichen), String (Zeichenfolge variabler Länge mit bis zu 254 Zeichen)
- Array (Seite 112)
- Datenstruktur (Seite 113): Struct
- PLC-Datentyp (Seite 114)
- Datentyp Variant (Seite 114)

Ferner wird das folgende numerische BCD-Format von den Umwandlungsanweisungen unterstützt, obwohl es nicht als Datentyp zur Verfügung steht.

Tabelle 5-28 Größe und Bereich des BCD-Formats

Format	Größe (Bit)	Bereich	Beispiel für konstanten Eintrag
BCD16	16	-999 bis 999	123, -123
BCD32	32	-9999999 bis 9999999	1234567, -1234567

Siehe auch

Unterstützte Datentypen (Seite 824)

5.4.1 Datentypen Bool, Byte, Word und DWord

Tabelle 5-29 Bit- und Bitfolge-Datentypen

Daten- typ	Bit- größe	Zahlen- typ	Zahlen- bereich	Beispiele für Konstanten	Beispiele für Adressen
Bool	1	Boolesch	FALSCH oder WAHR	WAHR	I1.0 Q0.1 M50.7 DB1.DBX2.3 Variablenname
		Binär	2#0 oder 2#1	2#0	
		Vorzeichenlose Ganzzahl	0 oder 1	1	
		Oktal	8#0 oder 8#1	8#1	
		Hexadezimal	16#0 oder 16#1	16#1	
Byte	8	Binär	2#0 bis 2#1111_1111	2#1000_1001	IB2 MB10 DB1.DBB4 Variablenname
		Vorzeichenlose Ganzzahl	0 bis 255	15	
		Vorzeichenbehaf- tete Ganzzahl	-128 bis 127	-63	
		Oktal	8#0 bis 8#377	8#17	
		Hexadezimal	B#16#0 bis B#16#FF, 16#0 bis 16#FF	B#16#F, 16#F	
Word	16	Binär	2#0 bis 2#1111_1111_1111_1111	2#1101_0010_1001_011 0	MW10 DB1.DBW2 Variablenname
		Vorzeichenlose Ganzzahl	0 bis 65535	61680	
		Vorzeichenbehaf- tete Ganzzahl	-32768 bis 32767	72	
		Oktal	8#0 bis 8#177_777	8#170_362	
		Hexadezimal	W#16#0 bis W#16#FFFF, 16#0 bis 16#FFFF	W#16#F1C0, 16#A67B	
DWord	32	Binär	2#0 bis 2#1111_1111_1111_1111_1111 _1111_1111_1111	2#1101_0100_1111_111 0_1000_1100	MD10 DB1.DBD8 Variablenname
		Vorzeichenlose Ganzzahl*	0 bis 4_294_967_295	15_793_935	
		Vorzeichenbehaf- tete Ganzzahl*	-2_147_483_648 bis 2_147_483_647	-400000	
		Oktal	8#0 bis 8#37_777_777_777	8#74_177_417	
		Hexadezimal	DW#16#0000_0000 bis DW#16#FFFF_FFFF, 16#0000_0000 bis 16#FFFF_FFFF	DW#16#20_F30A, 16#B_01F6	

* Der Unterstrich "_" ist ein Tausendertrennzeichen, um die Lesbarkeit von Zahlen mit mehr als acht Ziffern zu verbessern.

5.4.2 Ganzzahlige Datentypen

Tabelle 5-30 Ganzzahlige Datentypen (U = vorzeichenlos, S = kurz, D = doppelt)

Daten- typ	Bitgrö- ße	Zahlenbereich	Beispiele für Konstan- ten	Beispiele für Ad- ressen
USInt	8	0 bis 255	78, 2#01001110	MB0, DB1.DBB4, Variablenname
SInt	8	-128 bis 127	+50, 16#50	
UInt	16	0 bis 65.535	65295, 0	MW2, DB1.DBW2, Variablenname
Int	16	-32.768 bis 32.767	30000, +30000	
UDInt	32	0 bis 4.294.967.295	4042322160	MD6, DB1.DBD8, Variablenname
DInt	32	-2.147.483.648 bis 2.147.483.647	-2131754992	

5.4.3 Gleitpunktzahl/Realzahl-Datentypen

Realzahlen (bzw. Gleitpunktzahlen) werden als einfachgenaue 32-Bit-Zahlen (Real) oder als doppeltgenaue 64-Bit-Zahlen (LReal) dargestellt, wie in der Norm ANSI/IEEE7541985 beschrieben. Einfachgenaue Gleitpunktzahlen sind bis zu 6 signifikante Ziffern genau und doppeltgenaue Gleitpunktzahlen sind bis zu 15 signifikante Ziffern genau. Sie können maximal 6 (Real) oder 15 (LReal) signifikante Ziffern angeben, wenn Sie eine Gleitpunktkonstante eingeben.

Tabelle 5-31 Gleitpunktzahl/Realzahl-Datentypen (L = lang)

Daten- typ	Bitgrö- ße	Zahlenbereich	Beispiele für Konstan- ten	Beispiele für Ad- ressen
Real	32	-3.402823e+38 bis -1.175 495e-38, ±0, +1.175 495e-38 bis +3.402823e +38	123.456, -3.4, 1,0e-5	MD100, DB1.DBD8, Variablenname
LReal	64	-1.7976931348623158e+308 bis -2.2250738585072014e-308, ±0, +2.2250738585072014e-308 bis +1.7976931348623158e+308	12345,123456789e40, 1,2E+40	DB-Name.Var-Name Regeln: <ul style="list-style-type: none"> • Direkte Adres- sierung wird nicht unter- stützt • Kann in der Bausteinschnitt- stelle eines OBs, FBs oder FCs zu- gewiesen wer- den

Bei Berechnungen, die eine lange Reihe von Werten einschließlich sehr großen und sehr kleinen Zahlen benötigen, kann es zu ungenauen Ergebnissen kommen. Dies kann auftreten, wenn sich die Zahlen um 10 hoch x unterscheiden, wobei x > 6 (Real) oder 15 (LReal) ist. Beispiel (Real): 100 000 000 + 1 = 100 000 000.

5.4.4 Uhrzeit- und Datums-Datentypen

Tabelle 5-32 Uhrzeit- und Datums-Datentypen

Datentyp	Größe	Bereich	Beispiele für konstanten Eintrag
Time	32 Bits	T#-24d_20h_31m_23s_648ms bis T#24d_20h_31m_23s_647ms Gespeichert als: -2.147.483.648 ms bis +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 bis D#2168-12-31	D#2009-12-31 DATE#2009-12-31 2009-12-31
Time_of_Day	32 Bit	TOD#0:0:0.0 bis TOD#23:59:59.999	TOD#10:20:30.400 TIME_OF_DAY#10:20:30.400 23:10:1
DTL (Date and Time Long)	12 Byte	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2262-04-11:23:47:16.854 775 807	DTL#2008-12-16-20:30:20.25 0

Zeit

TIME-Daten werden als vorzeichenbehaftete doppelte Ganzzahl gespeichert, die als Millisekunden ausgewertet wird. Das Editorformat kann Angaben für Tag (d), Stunden (h), Minuten (m), Sekunden (s) und Millisekunden (ms) verarbeiten.

Es ist nicht erforderlich, alle Zeiteinheiten anzugeben. So sind z. B. T#5h10s und 500h gültig.

Der Gesamtwert aller angegebenen Werte in den einzelnen Einheiten darf den oberen bzw. unteren Grenzwert in Millisekunden für den Zeit-Datentyp nicht über- bzw. unterschreiten (-2.147.483.648 ms bis +2.147.483.647 ms).

Datum

DATE-Daten werden als vorzeichenloser Ganzzahlenwert gespeichert, der als Anzahl der Tage ausgewertet wird, die zum Basisdatum 01.01.1990 addiert werden, um das angegebene Datum zu erhalten. Das Editorformat muss ein Jahr, einen Monat und einen Tag angeben.

TOD

TOD-Daten (TIME_OF_DAY) werden als vorzeichenlose doppelte Ganzzahl gespeichert, die als Anzahl der Millisekunden seit Mitternacht für die angegebene Uhrzeit ausgewertet wird (Mitternacht = 0 ms). Es müssen die Stunden (24 h/Tag), Minuten und Sekunden angegeben werden. Die Angabe der Nachkommastellen der Sekunde ist optional.

DTL

Der DTL-Datentyp (Date and Time Long) nutzt eine 12-Byte-Struktur, um Angaben zum Datum und zur Uhrzeit zu speichern. Sie können den Datentyp DTL entweder im temporären Speicher eines Bausteins oder in einem DB definieren. Für alle Komponenten muss in der Spalte für den Startwert im DB-Editor ein Wert eingegeben werden.

Tabelle 5-33 Größe und Bereich von DTL

Länge (Bytes)	Format	Wertebereich	Beispiel für Werteingabe
12	Uhrzeit und Kalender Jahr-Monat-Tag:Stunde:Minute: Sekunde.Nanosekunden	Min.: DTL#1970-01-01-00:00:00.0 Max.: DTL#2554-12-31-23:59:59.999 999	DTL#2008-12-16-20:30:20.250

Jede Komponente des Datentyps DTL enthält einen unterschiedlichen Datentyp und Wertebereich. Der Datentyp eines angegebenen Werts muss dem Datentyp der jeweiligen Komponenten entsprechen.

Tabelle 5-34 Elemente der DTL-Struktur

Byte	Komponente	Datentyp	Wertebereich
0	Jahr	UINT	1970 bis 2554
1			
2	Monat	USINT	1 bis 12
3	Tag	USINT	1 bis 31
4	Wochentag ¹	USINT	1 (Sonntag) bis 7 (Samstag) ¹
5	Stunde	USINT	0 bis 23
6	Minute	USINT	0 bis 59
7	Sekunde	USINT	0 bis 59
8	Nanosekunden	UDINT	0 bis 999.999.999
9			
10			
11			

¹ Das Format Jahr-Monat-Tag:Stunde:Minute:
Sekunde.Nanosekunde enthält nicht den Wochentag.

5.4.5 Zeichen- und Zeichenfolge-Datentypen

Tabelle 5-35 Zeichen- und Zeichenfolge-Datentypen

Datentyp	Größe	Bereich	Beispiel für konstanten Eintrag
Char	8 Bit	16#00 bis 16#FF	'A', 't', '@', 'ä', 'Σ'
WChar	16 Bits	16#0000 bis 16#FFFF	'A', 't', '@', 'ä', 'Σ', Asiatische Zeichen, kyrillische Zeichen und andere

Datentyp	Größe	Bereich	Beispiel für konstanten Eintrag
String	n+ 2 Byte	n = (0 bis 254 Bytes)	"ABC"
WString	n+ 2 Wörter	n = (0 bis 65534 Wörter)	"ä123@XYZ.COM"

Char und WChar

Ein Char (Zeichen) belegt ein Byte im Speicher und speichert ein einzelnes Zeichen im ASCII-Format, einschließlich der erweiterten ASCII-Zeichencodes. Ein WChar belegt ein Wort im Speicher und kann eine beliebige Doppelbyte-Zeichendarstellung enthalten.

In der Editorsyntax wird ein einzelnes Anführungszeichen vor und hinter dem Zeichen eingegeben. Sie können sichtbare Zeichen und Steuerzeichen verwenden.

String und WString

Die CPU unterstützt den Datentyp String zum Speichern einer Folge von Einzelbyte-Zeichen. Der Datentyp String enthält die Gesamtzeichenzahl (Anzahl der Zeichen in der Zeichenkette) und die tatsächliche Zeichenzahl. Der Datentyp String bietet bis zu 256 Bytes zum Speichern der maximalen Gesamtzeichenzahl (1 Byte), der tatsächlichen Zeichenzahl (1 Byte) und bis zu 254 Bytes in der Zeichenfolge. Jedes Byte in einem Datentyp String kann einen beliebigen Wert von 16#00 bis 16#FF annehmen.

Der Datentyp WString stellt längere Zeichenfolgen für Werte mit einem Wort (Doppelbyte) bereit. Das erste Wort enthält die maximale Gesamtzeichenzahl; das nächste Wort enthält die tatsächliche Gesamtzeichenzahl, die folgende Zeichenfolge kann bis zu 65334 Wörter enthalten. Jedes Wort in einem Datentyp WString kann ein beliebiger Wert von 16#0000 bis 16#FFFF sein.

Sie können literale Zeichenketten (Konstanten) für Anweisungsparameter vom Typ IN in einzelnen Hochkommata angeben. 'ABC' zum Beispiel ist eine Zeichenkette aus drei Zeichen, die als Eingang für Parameter IN der Anweisung S_CONV genutzt werden kann. Sie können auch Zeichenkettenvariablen erstellen, indem Sie in den Bausteinschnittstelleneditoren für OB, FC, FB und DB den Datentyp "String" oder "WString" auswählen. Im PLC-Variableneditor können Sie keine Zeichenkette erstellen.

Sie können die maximale Zeichenfolgröße in Bytes (String) oder in Wörtern (WString) mit Hilfe von eckigen Klammern nach dem Schlüsselwort "String" oder "WString" angeben, nachdem Sie den jeweiligen Datentyp in der Klappliste für die Datentypauswahl ausgewählt haben. Beispiel: "MeinString String[10]" gibt eine maximale Größe von 10 Byte für die Zeichenfolge MeinString an. Wenn Sie die eckigen Klammern mit der Angabe der maximalen Größe weglassen, wird die Größe 254 für Datentyp String und 65534 für Datentyp WString angenommen. "MyWString WString[1000]" stünde somit für einen Datentyp WString mit 1000 Wörtern.

Das folgende Beispiel zeigt eine Zeichenfolge mit der maximalen Zeichenzahl 10 und der tatsächlichen Zeichenzahl 3. Die Zeichenfolge enthält somit 3 Zeichen mit je einem Byte, kann aber bis auf 10 Zeichen mit je einem Byte erweitert werden.

Tabelle 5-36 Beispiel für einen Datentyp String

Gesamtzeichenzahl	Tatsächliche Zeichenzahl	Zeichen 1	Zeichen 2	Zeichen 3	...	Zeichen 10
10	3	'C' (16#43)	'A' (16#41)	'T' (16#54)	...	-
Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	...	Byte 11

Das folgende Beispiel zeigt einen Datentyp WString mit der maximalen Zeichenzahl 500 und der tatsächlichen Zeichenzahl 300. Die Zeichenfolge enthält somit 300 Zeichen mit je einem Wort, kann aber bis auf 500 Zeichen mit je einem Wort erweitert werden.

Tabelle 5-37 Beispiel für einen Datentyp WString

Gesamtzeichenzahl	Tatsächliche Zeichenzahl	Zeichen 1	Zeichen 2 bis 299	Zeichen 300	...	Zeichen 500
500	300	'ä' (16#0084)	ASCII-Zeichenwörter	'M' (16#004D)	...	-
Wort 0	Wort 1	Wort 2	Wörter 3 bis 300	Wort 301	...	Wort 501

ASCII-Steuerzeichen können in Char-, WChar-, String- und WString-Daten verwendet werden. Die folgende Tabelle zeigt Beispiele für die Syntax von Steuerzeichen.

Tabelle 5-38 Gültige ASCII-Steuerzeichen

Steuerzeichen	ASCII-Hexadezimalwert (Char)	ASCII-Hexadezimalwert (WChar)	Regelungsfunktion	Beispiele
\$L oder \$l	16#0A	16#000A	Zeilenvorschub	'\$LText', '\$0AText'
\$N oder \$n	16#0A und 16#0D	16#000A und 16#000D	Zeilenumbruch Die neue Zeile zeigt zwei Zeichen in der Zeichenfolge.	'\$NText', '\$0A \$0DText'
\$P oder \$p	16#0C	16#000C	Formularvorschub	'\$PText', '\$0CText'
\$R oder \$r	16#0D	16#000D	Zeilenschaltung (CR)	'\$RText', '\$0DText'
\$T oder \$t	16#09	16#0009	Tabulator	'\$TText', '\$09Text'
\$ \$	16#24	16#0024	Dollarzeichen	'100\$\$', '100\$24'
\$'	16#27	16#0027	Einzelnes Hochkomma	'\$'Text\$', '\$27Text \$27'

5.4.6 Datentyp ARRAY

Arrays

Sie können ein Array erstellen, das mehrere Elemente des gleichen Datentyps enthält. Arrays können in der Bausteinschnittstelle von OB, FC, FB und DB angelegt werden. Im PLC-Variableneditor können Sie kein Array erstellen.

Um ein Array in der Bausteinschnittstelle zu erstellen, benennen Sie das Array und wählen den Datentyp "Array [lo .. hi] of type", dann ändern Sie "lo", "hi" und "type" wie folgt:

- lo - Anfangsindex (niedrigster Index) für Ihr Array
- hi - Abschlussindex (höchster Index) für Ihr Array
- type - einer der Datentypen wie BOOL, SINT, UDINT

Tabelle 5-39 Regeln für den Datentyp ARRAY

Datentyp	Array-Syntax		
ARRAY	Name [index1_min..index1_max, index2_min..index2_max] of <Datentyp>		
	<ul style="list-style-type: none"> • Alle Array-Elemente müssen den gleichen Datentyp haben. • Der Index kann negativ sein, doch der untere Grenzwert muss kleiner oder gleich dem oberen Grenzwert sein. • Arrays können bis zu sechs Dimensionen umfassen. • Multidimensionale Indexdeklarationen min..max werden durch Kommazzeichen voneinander getrennt. • Verschachtelte Arrays bzw. Arrays aus Arrays sind nicht zulässig. • Die Speichergröße eines Arrays = (Größe eines Elements * Gesamtzahl der Elemente im Array) 		
	Arrayindex	Gültige Index-Datentypen	Regeln für den Arrayindex
Konstante oder Variable	USInt, SInt, UInt, Int, UDInt, DInt	<ul style="list-style-type: none"> • Grenzwerte: -32768 bis +32767 • Gültig: Gemischte Konstanten und Variablen • Gültig: Konstante Ausdrücke • Ungültig: Variable Ausdrücke 	

Beispiel: Array-Deklarationen

ARRAY[1..20] of REAL
 ARRAY[-5..5] of INT
 ARRAY[1..2, 3..4] of CHAR

Eine Dimension, 20 Elemente
 Eine Dimension, 11 Elemente
 Zwei Dimensionen, 4 Elemente

Beispiel: Array-Adressen

ARRAY1[0]
 ARRAY2[1,2]
 ARRAY3[i,j]

ARRAY1 Element 0
 ARRAY2 Element [1,2]
 Wenn i = 3 und j = 4, dann wird ARRAY3 Element [3, 4] angesprochen

5.4.7 Datentyp Struktur

Mit dem Datentyp "Struct" können Sie eine aus anderen Datentypen bestehende Datenstruktur definieren. Der Datentyp Struct kann genutzt werden, um eine Gruppe zusammengehöriger Prozessdaten als eine Dateneinheit zu behandeln. Der Datentyp Struct wird benannt und die interne Datenstruktur im Datenbausteineditor oder in einem Bausteinschnittstelleneditor deklariert.

Arrays und Strukturen können auch zu einer größeren Struktur zusammengefügt werden. Eine Struktur kann bis zu acht Ebenen tief verschachtelt werden. Sie können z. B. eine Struktur aus Strukturen erstellen, die wiederum Arrays enthalten.

5.4.8 PLC-Datentyp

Im PLC-Datentypeditor können Sie Datenstrukturen definieren, die Sie mehrmals in Ihrem Programm verwenden können. Sie erstellen einen PLC-Datentyp durch Öffnen von "PLC-Datentypen" in der Projektnavigation und Doppelklick auf den Befehl "Neuen Datentyp hinzufügen". Klicken Sie zweimal einzeln auf den neu erstellen PLC-Datentyp, um den Standardnamen zu ändern. Doppelklicken Sie dann, um den PLC-Datentypeditor zu öffnen.

Zum Erstellen einer benutzerdefinierten PLC-Datentypstruktur können Sie auf dieselbe Weise vorgehen wie im Datenbausteineditor. Fügen Sie für alle zusätzlich erforderlichen Datentypen neue Zeilen ein, um die gewünschte Datenstruktur anzulegen.

Wenn Sie einen PLC-Datentyp anlegen, erscheint der Name des neuen PLC-Datentyps in der Auswahl-Klappliste im DB-Editor und im Codebaustein-Schnittstelleneditor.

Sie können PLC-Datentypen potentiell wie folgt verwenden:

- Als Datentyp in einer Codebausteinschnittstelle oder in Datenbausteinen
- Als Vorlage für die Erstellung mehrerer globaler Datenbausteine mit der gleichen Datenstruktur
- Als Datentyp für PLC-Variablendeklarationen in den Speicherbereichen E und A der CPU

Ein PLC-Datentyp kann beispielsweise ein Rezept zum Mischen von Farben sein. Sie können diesen PLC-Datentyp dann mehreren Datenbausteinen zuweisen. Sie können die Variablen in jedem Datenbaustein anpassen, um eine bestimmte Farbe zu erzeugen.

5.4.9 Pointer-Datentyp "Variant"

Der Datentyp Variant kann auf Variablen verschiedener Datentypen oder Parameter verweisen. Der Pointer Variant kann auf Strukturen und einzelne Strukturkomponenten zeigen. Der Pointer Variant belegt keinen Platz im Speicher.

Tabelle 5-40 Eigenschaften des Pointers Variant

Länge (Byte)	Darstellung	Format	Beispieleintrag
0	Symbolischer	Operand	MeineVariable
		DB-Name.Strukturname.Elementname	MeinDB.Strukt1.Druck1
	Absoluter	Operand	%MW10
		DB-Nummer.Operand Typ Länge	P#DB10.DBX10.0 INT 12

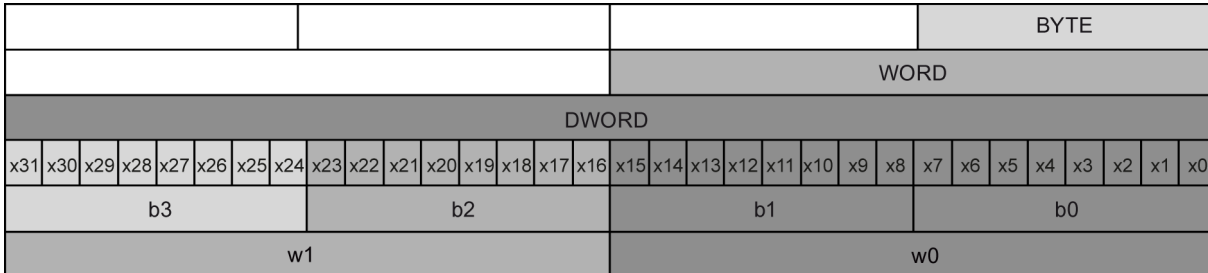
5.4.10 Zugriff auf eine "Slice" eines Variablentyps

Auf PLC-Variablen und Datenbausteinvariablen kann je nach ihrer Größe auf Bit-, Byte- oder Wortebene zugegriffen werden. Die Syntax für den Zugriff auf eine Daten-Slice lautet wie folgt:

- "<PLC-Variablenname>".xn (Bitzugriff)
- "<PLC-Variablenname>".bn (Bytezugriff)
- "<PLC-Variablenname>".wn (Wortzugriff)

- "<Datenbausteinname>.<Variablenname>.xn (Bitzugriff)
- "<Datenbausteinname>.<Variablenname>.bn (Bytezugriff)
- "<Datenbausteinname>.<Variablenname>.wn (Wortzugriff)

Auf eine Variable von der Größe eines Datendoppelworts kann über Bits 0 - 31, Bytes 0 - 3 oder Wörter 0 - 1 zugegriffen werden. Auf eine Variable von der Größe eines Worts kann über Bits 0 - 15, Bytes 0 - 1 oder Wort 0 zugegriffen werden. Auf eine Variable von der Größe eines Bytes kann über Bits 0 - 7 oder Byte 0 zugegriffen werden. Bit-, Byte- und Wort-Slices können überall dort genutzt werden, wo Bits, Bytes oder Wörter erwartete Operanden sind.



Hinweis

Auf folgende Datentypen kann über Slices zugegriffen werden: 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 und Word. Auf PLC-Variablen vom Datentyp Real kann über Slices zugegriffen werden, auf Datenbausteinvariablen vom Typ Real jedoch nicht.

Beispiele

In der PLC-Variablentabelle ist "DW" eine deklarierte Variable vom Typ DWORD. Die Beispiele zeigen den Zugriff in Form von Bit-, Byte- und Wort-Slices:

	KOP	FUP	SCL
Bitzugriff	<p>"DW".x11</p>		<pre>IF "DW".x11 THEN ... END_IF;</pre>
Bytezugriff	<p>"DW".b2</p> <p>==</p> <p>Byte</p> <p>"DW".b3</p>		<pre>IF "DW".b2 = "DW".b3 THEN ... END_IF;</pre>
Wortzugriff			<pre>out:= "DW".w0 AND "DW".w1;</pre>

5.4.11 Zugriff auf eine Variable mit einer AT-Überlagerung

Mit Hilfe der AT-Variablenüberlagerung können Sie mit einer überlagerten Deklaration eines anderen Datentyps auf eine bereits deklarierte Variable eines Bausteins zugreifen. Sie können beispielsweise die einzelnen Bits einer Variable vom Datentyp Byte, Word oder DWord mit einem Bool-Array adressieren. AT-Überlagerungen stehen für die folgenden Variablentypen zur Verfügung:

- Variablen in einem Standardzugriffsbaustein
- Remanente Variablen in einem optimierten Baustein

Deklaration

Um einen Parameter zu überlagern, deklarieren Sie einen zusätzlichen Parameter direkt nach dem zu überlagernden Parameter und wählen den Datentyp "AT". Der Editor legt die Überlagerung an und Sie können dann den Datentyp, die Struktur oder das Array für die Überlagerung wählen.

Beispiel

Dieses Beispiel zeigt die Eingangsparameter eines FBs mit Standardzugriff. Die Bytevariable B1 wird mit einem Booleschen Array überlagert:


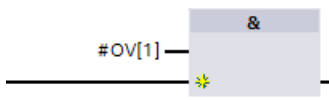
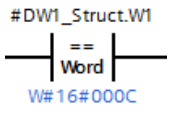
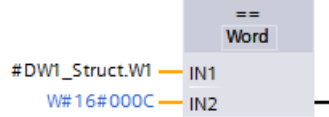
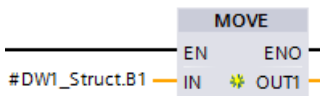
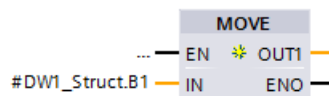
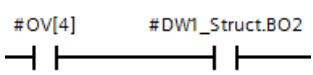
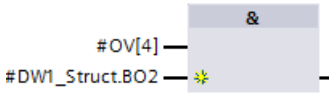
☐	B1	Byte	0.0
☐	▼ OV	AT*B1*	Array[0..7] of Bool
☐	☐ OV[0]	Bool	0.0
☐	☐ OV[1]	Bool	0.1
☐	☐ OV[2]	Bool	0.2
☐	☐ OV[3]	Bool	0.3
☐	☐ OV[4]	Bool	0.4
☐	☐ OV[5]	Bool	0.5
☐	☐ OV[6]	Bool	0.6
☐	☐ OV[7]	Bool	0.7

Ein weiteres Beispiel ist eine Variable vom Typ DWord, die mit einem Datentyp Struct überlagert wird. Struct enthält ein Wort, ein Byte und zwei Boolesche Werte:

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

Die Offset-Spalte der Bausteinschnittstelle zeigt die Lage der überlagerten Datentypen relativ zur Originalvariablen.

Die Überlagerungstypen können in der Programmlogik direkt angesprochen werden:

KOP	FUP	SCL
		<pre>IF #OV[1] THEN ... END_IF;</pre>
		<pre>IF #DW1_Struct.W1 = W#16#000C THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.B1;</pre>
		<pre>IF #OV[4] AND #DW1_Struct.BO2 THEN ... END_IF;</pre>

Regeln

- In FB- und FC-Bausteinen mit Standardzugriff (kein optimierter Zugriff) ist die Überlagerung von Variablen möglich.
- In optimierten FB- und FC-Bausteinen ist die Überlagerung von Variablen möglich, die remanent sind.
- Sie können Parameter für alle Baustein Typen und alle Deklarationsabschnitte überlagern.
- Ein überlagerter Parameter kann wie jeder andere Bausteinparameter verwendet werden.
- Parameter vom Typ VARIANT können Sie nicht überlagern.
- Die Größe des überlagernden Parameters muss kleiner oder gleich der Größe des überlagerten Parameters sein.
- Die überlagernde Variable muss sofort nach der Variablen deklariert werden, die sie überlagert, und muss durch das Schlüsselwort "AT" als Ausgangsdatentyp gekennzeichnet sein.

5.5 Memory Card verwenden

Hinweis

Die CPU unterstützt nur die vorformatierten SIMATIC Memory Cards (Seite 1430).

Löschen Sie vor dem Kopieren von Programmen auf die formatierte Memory Card alle zuvor gespeicherten Programme von der Memory Card.

Sie können eine Memory Card wie folgt verwenden:

- Sie können die Memory Card als Übertragungskarte oder als Programmkarte nutzen. Übertragungskarten und Programmkarten enthalten alle Codebausteine und Datenbausteine, alle Technologieobjekte und die Gerätekonfiguration. Auf Übertragungskarten und Programmkarten **nicht** enthalten sind beispielsweise Force-Tabellen, Beobachtungstabellen oder PLC-Variablen Tabellen.
 - Mit einer Übertragungskarte (Seite 122) kopieren Sie ein Programm in den internen Ladespeicher der CPU, ohne dafür STEP 7 zu verwenden.
 - Mit einer leeren Übertragungskarte können Sie auf eine passwortgeschützte CPU zugreifen, wenn Sie das Passwort verloren oder vergessen haben (Seite 134).
 - Eine Programmkarte (Seite 124) nutzen Sie als externen Ladespeicher für die CPU.
- Sie verwenden eine Memory Card auch zum Herunterladen von Firmware-Updates (Seite 131).
- Sie verwenden eine Memory Card zum Festlegen oder Ändern des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten.

5.5.1 Memory Card in die CPU stecken

ACHTUNG

Schutz von Memory Card und Schacht vor elektrostatischer Entladung

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.



Stellen Sie sicher, dass die Memory Card nicht schreibgeschützt ist. Schieben Sie dazu den Schutzschalter aus der Verriegelungsposition heraus.

Wenn Sie eine schreibgeschützte Memory Card in die CPU stecken, zeigt STEP 7 beim nächsten Anlauf eine Diagnosemeldung an, um Sie auf diesen Zustand hinzuweisen. Die CPU läuft dennoch fehlerfrei hoch, aber Anweisungen mit Rezepten oder Datenprotokollen führen bei einer schreibgeschützte Karte zu Fehlermeldungen.

 **WARNUNG**

Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt.

Wenn Sie eine Memory Card (unabhängig davon, ob als Programm-, Übertragungs- oder Firmware-Aktualisierungskarte genutzt) in eine laufende CPU stecken, geht die CPU sofort in den Betriebszustand STOP, was zu Prozessunterbrechung und dadurch zu Tod oder schweren Personenschäden führen kann.

Stellen Sie vor dem Stecken oder Ziehen einer Memory Card stets sicher, dass die CPU nicht aktiv eine Maschine oder einen Prozess steuert. Installieren Sie einen NOT-AUS-Schaltkreis für Ihre Anwendung bzw. Ihren Prozess.

Hinweis

Stecken Sie keine Übertragungskarte mit einem Programm V3.0 in S7-1200 CPUs V4.x.

Übertragungskarten mit einem Programm der Version 3.0 sind nicht mit S7-1200 CPUs der Version 4.x kompatibel. Wenn Sie eine Memory Card mit einem Programm V3.0 stecken, wird dadurch ein CPU-Fehler verursacht.

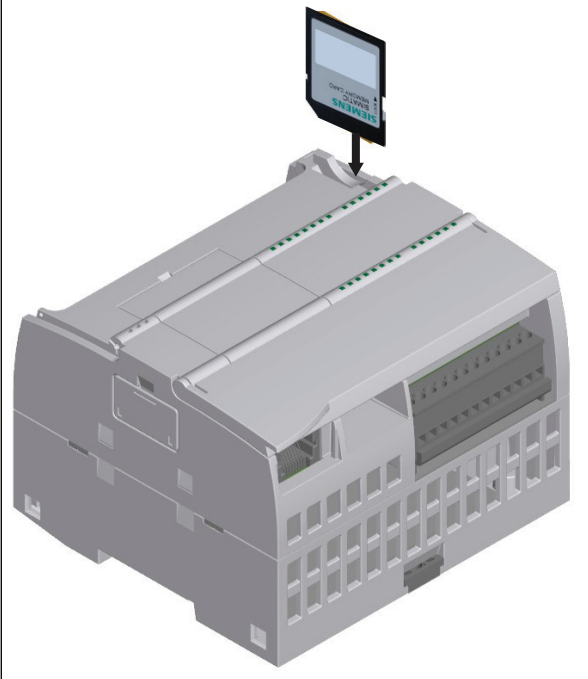
Wenn Sie eine Übertragungskarte (Seite 122) mit einer ungültigen Programmversion stecken, müssen Sie die Karte entnehmen und die CPU von STOP in RUN versetzen, ein Urlöschen (MRES) durchführen oder die CPU aus- und wieder einschalten. Nachdem Sie den Fehlerzustand der CPU behoben haben, können Sie ein gültiges CPU-Programm V4.x laden.

Um ein V3.0-Programm in ein V4.x-Programm zu übertragen, müssen Sie das TIA Portal verwenden und in der Hardware-Konfiguration das Gerät ändern.

Hinweis

Wenn Sie eine Memory Card im Betriebszustand STOP der CPU stecken, zeigt der Diagnosepuffer die Meldung an, dass die Auswertung der Memory Card gestartet wurde. Die CPU wertet die Memory Card aus, wenn Sie entweder die CPU in den Betriebszustand RUN versetzen, den Speicher der CPU über MRES urlöschen oder die CPU aus- und wieder einschalten.

Tabelle 5-41 Memory Card einsetzen

	<p>Zum Einsetzen einer Memory Card öffnen Sie die obere Abdeckung der CPU und stecken die Memory Card in den Steckplatz. Ein Steckverbinder ermöglicht einfaches Stecken und Ziehen des Moduls.</p> <p>Die Memory Card ist so geformt, dass sie nur in eine Richtung in den Schacht gesteckt werden kann.</p>
---	---

CPU-Verhalten beim Stecken einer Memory Card

Wenn Sie eine Memory Card in die CPU stecken, führt die CPU folgende Schritte aus:

1. Wechsel in den Betriebszustand STOP (sofern noch nicht in STOP)
2. Anzeige einer Eingabeaufforderung mit den folgenden Optionen:
 - Ausschalten und Wiedereinschalten
 - Wechsel in den Betriebszustand RUN
 - Durchführen von Urlöschen
3. Auswertung der Karte

Auswertung der Memory Card durch die CPU

Wenn Sie in den Schutzzeigenschaften der Gerätekonfiguration (Seite 167) der CPU die Einstellung "Kopieren vom internen Ladespeicher in den externen Ladespeicher deaktivieren" nicht auswählen, ermittelt die CPU, welche Art von Memory Card Sie stecken:

- **Leere Memory Card:** Eine leere Memory Card hat keine Auftragsdatei (S7_JOB.S7S). Wenn Sie eine leere Memory Card stecken, fügt die CPU eine Programmauftragsdatei hinzu. Sie kopiert dann den internen Ladespeicher in den externen Ladespeicher (die Programmdatei auf der Memory Card) und löscht den internen Ladespeicher.
- **Leere Programmkarte:** Eine leere Programmkarte enthält eine Programmauftragsdatei, die leer ist. In diesem Fall kopiert die CPU den internen Ladespeicher in den externen Ladespeicher (die Programmdatei auf der Memory Card) und löscht den internen Ladespeicher.

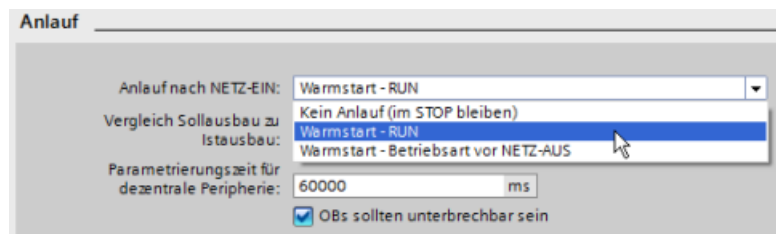
Wenn Sie in den Schutzeigenschaften der Gerätekonfiguration der CPU die Einstellung "Kopieren vom internen Ladespeicher in den externen Ladespeicher deaktivieren" ausgewählt haben, verhält sich die CPU wie folgt:

- **Leere Memory Card:** Eine leere Memory Card hat keine Auftragsdatei (S7_JOB.S7S). Wenn Sie eine leere Memory Card stecken, führt die CPU nichts durch. Sie erstellt keine Programmauftragsdatei und sie kopiert nicht den internen Ladespeicher in den externen Ladespeicher (die Programmdatei auf der Memory Card). Sie löscht nicht den internen Ladespeicher.
- **Leere Programmkarte:** Eine leere Programmkarte enthält eine Programmauftragsdatei, die leer ist. In diesem Fall führt die CPU keine Aktion durch. Sie kopiert nicht den internen Ladespeicher in den externen Ladespeicher (die Programmdatei auf der Memory Card). Sie löscht nicht den internen Ladespeicher.

Wenn Sie eine Programmkarte (Seite 124), Übertragungskarte (Seite 122) oder eine Karte mit einer Firmware-Aktualisierung (Seite 131) in die CPU stecken, wirkt sich die Konfigurationseinstellung "Kopieren vom internen Ladespeicher in den externen Ladespeicher deaktivieren" nicht darauf aus, wie die CPU die Memory Card auswertet.

5.5.2 Anlaufparameter der CPU vor dem Kopieren des Projekts auf die Memory Card konfigurieren

Wenn Sie ein Programm auf eine Übertragungskarte oder eine Programmkarte kopieren, enthält das Programm die Anlaufparameter für die CPU. Stellen Sie stets vor dem Kopieren des Programms auf die Memory Card sicher, dass Sie den Betriebszustand der CPU nach dem Ausschalten und Wiedereinschalten konfiguriert haben. Sie können wählen, ob die CPU im Betriebszustand STOP oder RUN oder im letzten Betriebszustand (vor dem Neustart) starten soll.



5.5.3 Übertragungskarte

ACHTUNG**Schutz von Memory Card und Schacht vor elektrostatischer Entladung**

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Für den sicheren Umgang mit der Memory Card stellen Sie sicher, dass eine oder beide der folgenden Bedingungen erfüllt sind:

- Stellen Sie sich auf einen geerdeten, leitfähigen Untergrund.
- Tragen Sie ein geerdetes Armband, wenn Sie mit der Memory Card arbeiten.

Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.

Übertragungskarte anlegen

Denken Sie daran, die Anlaufparameter der CPU zu konfigurieren (Seite 121), bevor Sie ein Programm auf die Übertragungskarte kopieren. Um eine Übertragungskarte anzulegen, gehen Sie folgendermaßen vor:

1. Stecken Sie eine leere, nicht schreibgeschützte SIMATIC Memory Card in einen an Ihren Computer angeschlossenen SD-Kartenleser. (Falls die Memory Card schreibgeschützt ist, schieben Sie den Schutzschalter aus der Verriegelungsposition heraus.)
Wenn Sie eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm, Datenprotokolle, Rezepte oder ein Firmware-Update enthält, **müssen** Sie die Dateien löschen, bevor Sie die Karte erneut verwenden. Rufen Sie den Windows Explorer auf und zeigen Sie die Inhalte der Memory Card an. Löschen Sie die Datei "S7_JOB.S7S" sowie alle vorhandenen Ordner (wie "SIMATIC.S7S", "FWUPDATE.S7S", "DataLogs" und "Recipes").

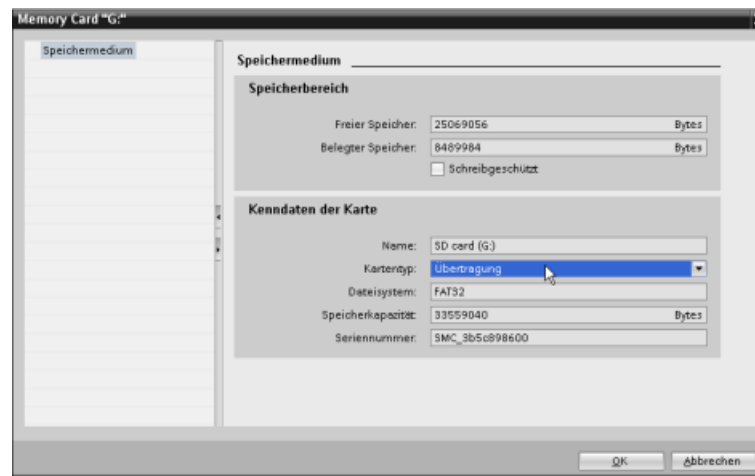
ACHTUNG**Löschen Sie NICHT die versteckten Dateien "__LOG__" und "crdinfo.bin" von der Memory Card.**

Die Dateien "__LOG__" und "crdinfo.bin" werden von der Memory Card benötigt. Wenn Sie diese Dateien löschen, können Sie die Memory Card nicht mehr mit der CPU nutzen.

2. Erweitern Sie in der Projektnavigation (Projektansicht) den Ordner "SIMATIC Kartenleser" und wählen Sie Ihren Kartenleser aus.
3. Klicken Sie mit der rechten Maustaste auf den Laufwerksbuchstaben der Memory Card im Kartenleser und wählen Sie im Kontextmenü die Option "Eigenschaften". Daraufhin wird der Dialog "Memory Card" angezeigt.

- Wählen Sie im Dialog "Memory Card" in der Klappliste "Kartentyp" die Option "Übertragen" aus.

Daraufhin erstellt STEP 7 die leere Übertragungskarte. Wenn Sie eine leere Übertragungskarte anlegen, weil Sie Ihr CPU-Passwort verloren haben (Seite 134), entnehmen Sie die Übertragungskarte aus dem Kartenleser.



- Fügen Sie das Programm hinzu, indem Sie in der Projektnavigation die CPU (z.B. PLC_1 [CPU 1214C DC/DC/DC]) auswählen und mit der Maus auf die Memory Card ziehen. (Alternativ können Sie die CPU kopieren und in die Memory Card einfügen.) Durch Kopieren der CPU in die Memory Card wird der Dialog "Vorschau laden" geöffnet.
- Klicken Sie im Dialog "Vorschau laden" auf die Schaltfläche "Laden", um die CPU in die Memory Card zu kopieren.
- Wenn der Dialog eine Meldung anzeigt, dass die CPU (das Programm) fehlerfrei geladen wurde, klicken Sie auf die Schaltfläche "Fertig stellen".

Übertragungskarte verwenden

WARNUNG

Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt.

Durch das Einlegen einer Memory Card geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unvorhersehbarer Betrieb eines Prozesses oder einer Maschine kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Stellen Sie vor dem Einlegen einer Übertragungskarte sicher, dass die CPU im Betriebszustand STOP ist und sich der Prozess in einem sicheren Zustand befindet.

Hinweis

Stecken Sie keine Übertragungskarte mit einem Programm V3.0 in CPU-Varianten höherer Versionen.

Übertragungskarten der Version 3.0 sind nicht mit S7-1200 CPUs höherer Versionen kompatibel. Wenn Sie eine Memory Card mit einem Programm V3.0 stecken, wird dadurch ein CPU-Fehler verursacht.

Wenn Sie eine Übertragungskarte mit einer ungültigen Programmversion stecken, entnehmen Sie die Karte und versetzen die CPU von STOP in RUN, führen ein Urlöschen (MRES) durch oder schalten die CPU aus und wieder ein. Nachdem Sie den Fehlerzustand der CPU behoben haben, können Sie ein gültiges CPU-Programm laden.

Um das Programm in die CPU zu übertragen, gehen Sie folgendermaßen vor:

1. Stecken Sie die Übertragungskarte in die CPU (Seite 118). Wenn sich die CPU in RUN befindet, geht die CPU in den Betriebszustand STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss. Das vorhandene Programm befindet sich jetzt immer noch in der CPU.
2. Schalten Sie die CPU aus und wieder ein, um die Memory Card auszuwerten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.
3. Nach dem Neustart wertet die CPU die Memory Card aus und kopiert das Programm in den internen Ladespeicher der CPU.
Die RUN/STOP-LED blinkt abwechselnd grün und gelb, um kenntlich zu machen, dass das Programm kopiert wird. Wenn die RUN/STOP-LED dauerhaft gelb leuchtet und die MAINT-LED gelb blinkt, ist der Kopiervorgang beendet. Nun können Sie die Memory Card entnehmen.
4. Starten Sie die CPU neu (entweder durch Einschalten oder durch eine der alternativen Methoden zum Neustarten), um das neue Programm, das in den internen Ladespeicher übertragen wurde, auszuwerten.

Die CPU geht dann in den Betriebszustand für den Anlauf (RUN oder STOP), den Sie für das Projekt eingerichtet haben.

Hinweis

Sie müssen die Übertragungskarte ziehen, bevor Sie die CPU in RUN versetzen.

5.5.4 Programmkarte

ACHTUNG

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.



Stellen Sie sicher, dass die Memory Card nicht schreibgeschützt ist. Schieben Sie dazu den Schutzschalter aus der Verriegelungsposition heraus.

Löschen Sie vor dem Kopieren von Programmelementen auf die Programmkarte alle zuvor gespeicherten Programme von der Memory Card.

Programmkarte anlegen

Bei Einsatz als Programmkarte funktioniert die Memory Card als externer Ladespeicher der CPU. Wenn Sie die Programmkarte ziehen, ist der interne Ladespeicher der CPU leer.

Hinweis

Wenn Sie eine leere Memory Card in die CPU stecken und eine Auswertung der Memory Card durchführen, indem Sie die CPU aus- und wieder einschalten, die CPU von STOP nach RUN versetzen oder ein Urlöschen (MRES) durchführen, werden das Programm und die geforderten Werte aus dem internen Ladespeicher der CPU in die Memory Card kopiert. (Die Memory Card ist nun eine Programmkarte.) Nach Abschluss des Kopiervorgangs wird das Programm im internen Ladespeicher der CPU gelöscht. Die CPU geht dann in den konfigurierten Betriebszustand für den Anlauf (RUN oder STOP).

Denken Sie daran, die Anlaufparameter der CPU zu konfigurieren (Seite 121), bevor Sie ein Projekt auf die Programmkarte kopieren. Um eine Programmkarte anzulegen, gehen Sie folgendermaßen vor:

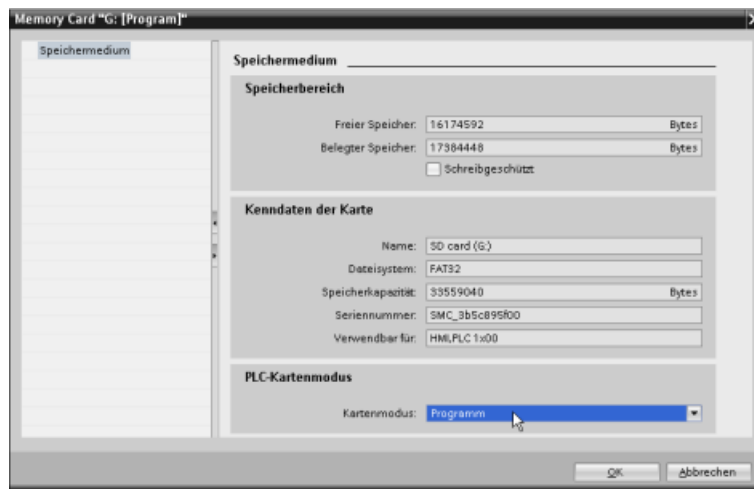
1. Stecken Sie eine leere, nicht schreibgeschützte SIMATIC Memory Card in einen an Ihren Computer angeschlossenen SD-Kartenleser. (Falls die Memory Card schreibgeschützt ist, schieben Sie den Schutzschalter aus der Verriegelungsposition heraus.)
Wenn Sie eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm, Datenprotokolle, Rezepte oder ein Firmware-Update enthält, **müssen** Sie die Dateien löschen, bevor Sie die Karte erneut verwenden. Mit dem Windows Explorer können Sie den Inhalt der Memory Card anzeigen und gegebenenfalls die folgenden Dateien und Ordner löschen:
 - S7_JOB.S7S
 - SIMATIC.S7S
 - FWUPDATE.S7S
 - DataLogs
 - Recipes
 - UserFiles

ACHTUNG

Löschen Sie NICHT die versteckten Dateien "__LOG__" und "crdinfo.bin" von der Memory Card.


Die Dateien "__LOG__" und "crdinfo.bin" werden von der Memory Card benötigt. Wenn Sie diese Dateien löschen, können Sie die Memory Card nicht mehr mit der CPU nutzen.

2. Erweitern Sie in der Projektnavigation (Projektansicht) den Ordner "Kartenleser/USB-Speicher" und wählen Sie Ihren Kartenleser aus.
3. Klicken Sie mit der rechten Maustaste auf den Laufwerksbuchstaben der Memory Card im Kartenleser und wählen Sie im Kontextmenü die Option "Eigenschaften". Daraufhin wird der Dialog "Memory Card" angezeigt.
4. Wählen Sie im Dialog "Memory Card" im Kontextmenü die Option "Programm" aus.



5. Fügen Sie das Programm hinzu, indem Sie in der Projektnavigation die CPU (z.B. PLC_1 [CPU 1214C DC/DC/DC]) auswählen und mit der Maus auf die Memory Card ziehen. (Alternativ können Sie die CPU kopieren und in die Memory Card einfügen.) Durch Kopieren der CPU in die Memory Card wird der Dialog "Vorschau laden" geöffnet.
6. Klicken Sie im Dialog "Vorschau laden" auf die Schaltfläche "Laden", um die CPU in die Memory Card zu kopieren.
7. Wenn der Dialog eine Meldung anzeigt, dass die CPU (das Programm) fehlerfrei geladen wurde, klicken Sie auf die Schaltfläche "Fertig stellen".

Programmkarte als externen Ladespeicher für die CPU nutzen

 WARNUNG
Risiken beim Stecken einer Programmkarte
Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt.
Durch das Einlegen einer Memory Card geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unvorhersehbarer Betrieb eines Prozesses oder einer Maschine kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.
Stellen Sie vor dem Einlegen einer Memory Card sicher, dass die CPU offline und in einem sicheren Zustand ist.

Um mit Ihrer CPU eine Programmkarte zu nutzen, gehen Sie folgendermaßen vor:

1. Schieben Sie die Programmkarte in die CPU. Befindet sich die CPU im Betriebszustand RUN, geht sie in STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Schalten Sie die CPU aus und wieder ein, um die Memory Card auszuwerten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.
3. Nach dem Neustart der CPU und der Auswertung der Programmkarte löscht die CPU den internen Ladespeicher der CPU.

Die CPU geht dann in den Betriebszustand für den Anlauf (RUN oder STOP), den Sie für die CPU eingerichtet haben.

Die Programmkarte muss in der CPU gesteckt bleiben. Wenn Sie die Programmkarte ziehen, hat die CPU kein Programm mehr im internen Ladespeicher.



WARNUNG

Risiken beim Ziehen einer Programmkarte

Wenn Sie die Programmkarte ziehen, verliert die CPU den externen Ladespeicher und erzeugt einen Fehler. Die CPU geht dann in den Betriebszustand STOP und die Fehler-LED blinkt.

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen. Daraus resultiert ein unvorhersehbarer Betrieb des Automatisierungssystems, der zu lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.

Ziehen Sie die Programmkarte nur, wenn Sie sich darüber im Klaren sind, dass Sie damit das Programm aus der CPU entfernen.

Lebensdauer einer SIMATIC Memory Card

Die Lebensdauer einer SIMATIC Memory Card ist von Faktoren wie den folgenden abhängig:

- Anzahl der Lösch- und Schreibvorgänge pro Speicherblock
- Anzahl der geschriebenen Bytes
- Externe Einflüsse wie Umgebungstemperatur

Hinweis

Auswirkungen von Schreib- und Löschvorgängen auf die Lebensdauer der SIMATIC Memory Card

Schreib- und Löschvorgänge, insbesondere wiederholte (zyklische) Schreib-/Löschvorgänge, verkürzen die Lebensdauer der SIMATIC Memory Card.

Die zyklische Ausführung der folgenden Aktionen verkürzt die Lebensdauer der Memory Card je nach Anzahl der Schreibvorgänge und Daten:

- Variablenarchivvorgänge (z.B. DataLogWrite)
 - Rezeptvorgänge (z.B. RecipeExport)
 - Systemfunktionsaufrufe (SFCs), die im Dateisystem schreiben/löschen (z.B. WRIT_DBL, CREATE)
 - Systemfunktionsbausteine (SFBs), die im Dateisystem schreiben/löschen (z.B. FileWriteC, FileDelete)
 - Alle anderen zyklischen Aktionen, die im persistenten Speicher Daten ändern (z.B. Tracing, SET-TimeZone)
-

5.5.5 Memory Card zum Schützen vertraulicher PLC-Konfigurationsdaten nutzen

Sie können eine SIMATIC Memory Card zum Festlegen oder Ändern des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten verwenden.

ACHTUNG**Schutz von Memory Card und Schacht vor elektrostatischer Entladung**

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen. Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.

ACHTUNG**Verwenden Sie zum Neuformatieren der Memory Card nicht das Windows-Formatierungsprogramm oder ein anderes Formatierungsprogramm.**

Wenn eine SIMATIC Memory Card mit dem Microsoft Windows-Formatierungsprogramm neu formatiert wird, kann die Memory Card nicht mehr von einer S7-1200 CPU verwendet werden.

Risiken im Zusammenhang mit der Außerbetriebsetzung

S7-1200 CPUs unterstützen nicht das sichere Löschen der Memory Card und des internen Flash-Speichers. Deshalb müssen Sie bei der Außerbetriebsetzung die CPU und die Memory Card sicher entsorgen, um den Verlust proprietärer und vertraulicher Informationen zu verhindern.

So erstellen Sie eine Memory Card mit dem Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten

Um die Memory Card mit diesem Passwort zu erstellen, gehen Sie folgendermaßen vor:

1. Stecken Sie eine leere SIMATIC Memory Card ohne Schreibschutz in einen SD-Kartenleser, der an Ihren Computer angeschlossen ist. Ist die Karte schreibgeschützt, schieben Sie den Schutzschalter aus der Verriegelungsposition heraus.
Sie können eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm oder ein Firmware-Update enthält, doch Sie müssen einige der Dateien auf der Memory Card löschen. Um eine Memory Card wiederzuverwenden, müssen Sie die Datei "S7_JOB.S7S" löschen, bevor Sie die Datei zum Schutz vertraulicher PLC-Konfigurationsdaten erstellen. Mit dem Windows Explorer können Sie den Inhalt der Memory Card anzeigen und die Datei "S7_JOB.S7S" und Ordner löschen.

ACHTUNG

Löschen Sie NICHT die versteckten Dateien "__LOG__" und "crdinfo.bin" von der Memory Card.

Die Dateien "__LOG__" und "crdinfo.bin" werden von der Memory Card benötigt. Wenn Sie diese Dateien löschen, können Sie die Memory Card nicht mehr mit der CPU nutzen.

2. Erstellen Sie eine Datei im Stammverzeichnis der Memory Card mit dem Namen "S7_JOB.S7S". Öffnen Sie die Datei mit einem Texteditor und geben Sie Folgendes ein:
SET_PWD.
3. Erstellen Sie einen Ordner im Stammverzeichnis der Memory Card mit dem Namen:
SET_PWD.S7S.
4. Erstellen Sie im Ordner "SET_PWD.S7S" eine Textdatei mit dem Namen "PWD.TXT". Die Datei muss "PWD.TXT" heißen. Geben Sie Ihr Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten als Textinhalt der Datei ein. Die Datei muss eine einzelne Textzeile mit dem Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten enthalten. Befolgen Sie beim Erstellen des Passworts die STEP 7-Regeln für Passwörter und verwenden Sie diese Zeichen:
 - 0123456789
 - A...Z a...z
 - !#\$%&()*+,-./:;<=>?@ [\]_{ }~^
5. Um Zurücksetzen des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten auf der PLC muss die Datei leer sein.
6. Nehmen Sie die Karte vorsichtig aus dem Lesegerät.

So legen Sie das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten fest

Zum Festlegen des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten gehen Sie folgendermaßen vor:

ACHTUNG

Vergewissern Sie sich vor dem Festlegen des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten, dass die CPU keinen Prozess ausführt.

Das Festlegen des Passworts zum Schutz vertraulicher PLC-Konfigurationsdaten kann bewirken, dass das vorhandene PLC-Programm nicht geladen werden kann. Bevor Sie die Memory Card stecken, stellen Sie unbedingt sicher, dass die CPU offline und in einem sicheren Zustand ist.

1. Schieben Sie die Memory Card in die CPU. Befindet sich die CPU im Betriebszustand RUN, geht sie in STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Starten Sie die CPU neu, um den Vorgang zu starten. Nach dem Neustart der CPU wird das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten festgelegt. Wenn die RUN/STOP-LED dauerhaft gelb leuchtet und die MAINT-LED blinkt, ist der Vorgang beendet. Anschließend müssen Sie die Memory Card entnehmen.
3. Nach dem Entnehmen der Memory Card starten Sie die CPU noch einmal neu, um das neue Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten zu verwenden.

Wenn das vorhandene Anwenderprogramm ein anderes Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten benötigt, wird es nach dem Neustart nicht geladen. Sie müssen das vorhandene Programm löschen und ein Programm ins Zielsystem laden, das das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten, das Sie in den vorherigen Schritten festgelegt haben, verwendet.

Wenn das vorhandene Programm das angegebene Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten benötigt, kann die PLC entsprechend der Projektkonfiguration in RUN gehen.

5.5.6 Firmware-Update

Mit einer SIMATIC Memory Card können Sie auch eine Firmware-Aktualisierung durchführen.

ACHTUNG

Schutz von Memory Card und Schacht vor elektrostatischer Entladung

Elektrostatische Entladungen können die Memory Card oder den dafür vorgesehenen Schacht in der CPU beschädigen.

Sie müssen auf einem leitfähigen, geerdeten Boden stehen und/oder ein geerdetes Armband tragen, wenn Sie mit der Memory Card arbeiten. Die Memory Card ist in einem leitfähigen Behälter aufzubewahren.

Sie verwenden die SIMATIC Memory Card auch zum Herunterladen von Firmware-Updates von der Siemens Industry-Website für Online-Support (<https://support.industry.siemens.com/cs/ww/de>). Gehen Sie auf dieser Website zu "Downloads". Von dort suchen Sie nach dem spezifischen Modultyp, den Sie aktualisieren möchten.

Alternativ können Sie die S7-1200 Website für Downloads (<https://support.industry.siemens.com/cs/ww/de/ps/13683/dl>) direkt aufrufen.

Hinweis

Sie können eine S7-1200 CPU V3.0 oder eine Vorgängerversion davon nicht auf die Firmware S7-1200 V4.0 (oder später) aktualisieren.

Ein Firmware-Update können Sie auch auf eine der folgenden Arten durchführen:

- Mit den Online- und Diagnosewerkzeugen von STEP 7 (Seite 1212)
- Über die Standard-Webseite "Modulinformationen" des Webservers (Seite 872)
- Mit dem SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/de/view/98161300>)

ACHTUNG

Verwenden Sie zum Neuformatieren der Memory Card nicht das Windows-Formatierungsprogramm oder ein anderes Formatierungsprogramm.

Wenn eine Siemens Memory Card mit dem Windows-Formatierungsprogramm neu formatiert wird, kann die Memory Card nicht mehr von einer S7-1200 CPU verwendet werden.
--

Um das Firmware-Update auf Ihre Memory Card zu laden, gehen Sie folgendermaßen vor:

1. Stecken Sie eine leere SIMATIC Memory Card ohne Schreibschutz in einen SD-Kartenleser, der an Ihren Computer angeschlossen ist. (Ist die Karte schreibgeschützt, schieben Sie den Schutzschalter aus der Verriegelungsposition heraus.)
Sie können eine SIMATIC Memory Card verwenden, die bereits ein Anwenderprogramm oder ein anderes Firmware-Update enthält. Um Verwirrung zu vermeiden, müssen Sie auch die Dateien S7_JOB.SYS, SIMATIC.S7S und FWUPDATE.S7S löschen, sofern sie vorhanden sind.


ACHTUNG

Löschen Sie NICHT die versteckten Dateien "__LOG__" und "crdinfo.bin" von der Memory Card.

Die Dateien "__LOG__" und "crdinfo.bin" werden von der Memory Card benötigt. Wenn Sie diese Dateien löschen, können Sie die Memory Card nicht mehr mit der CPU nutzen.
--

2. Wählen Sie die Zip-Datei des Ihrem Modul entsprechenden Firmware-Updates aus und laden Sie diese Zip-Datei auf Ihren Computer herunter. Doppelklicken Sie auf die Datei, geben Sie als Zielpfad das Stammverzeichnis der SIMATIC Memory Card an und starten Sie die Extraktion. Wenn die Extraktion beendet ist, enthält das Stammverzeichnis (Ordner) der Memory Card das Verzeichnis "FWUPDATE.S7S" und die Datei "S7_JOB.S7S".
3. Nehmen Sie die Karte vorsichtig aus dem Lesegerät.

Zum Installieren des Firmware-Updates gehen Sie folgendermaßen vor:

<p> WARNUNG</p> <p>Stellen Sie vor der Installation des Firmware-Updates sicher, dass die CPU keinen aktiven Prozess ausführt.</p> <p>Durch die Installation des Firmware-Updates geht die CPU in den Betriebszustand STOP, was sich auf den Betrieb eines Online-Prozesses oder einer Maschine auswirken kann. Unerwarteter Betrieb eines Prozesses oder einer Maschine kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p> <p>Bevor Sie die Memory Card stecken, stellen Sie unbedingt sicher, dass die CPU offline und in einem sicheren Zustand ist.</p>
--

1. Schieben Sie die Memory Card in die CPU. Befindet sich die CPU im Betriebszustand RUN, geht sie in STOP. Die Wartungs-LED (MAINT) blinkt, um anzuzeigen, dass die Memory Card ausgewertet werden muss.
2. Starten Sie die CPU neu, um das Firmware-Update zu starten. Alternative Methoden zum Neustarten der CPU sind ein Wechsel von STOP in RUN oder ein Urlöschen (MRES) in STEP 7.

Hinweis

Damit das Firmware-Upgrade für das Modul abgeschlossen werden kann, achten Sie darauf, dass die externe 24-V-DC-Versorgung des Moduls eingeschaltet bleibt.

Nach dem Neustart der CPU beginnt das Firmware-Update. Die RUN/STOP-LED blinkt grün und gelb, um anzuzeigen, dass das Update kopiert wird. Wenn die RUN/STOP-LED dauerhaft gelb leuchtet und die MAINT-LED blinkt, ist der Kopiervorgang beendet. Anschließend müssen Sie die Memory Card entnehmen.

3. Nachdem Sie die Memory Card entnommen haben, starten Sie die CPU erneut neu (entweder durch Einschalten oder mit einem anderen Verfahren zum Neustarten), um die neue Firmware zu laden.

Das Anwenderprogramm und die Hardware-Konfiguration sind vom Firmware-Update nicht betroffen. Wenn die CPU eingeschaltet wird, geht die CPU in den konfigurierten Anlaufzustand. (Wenn als Anlaufbetriebszustand für die CPU "Warmstart - Betriebsart vor NETZ-AUS" konfiguriert wurde, befindet sich die CPU im Betriebszustand STOP, weil STOP der letzte Betriebszustand der CPU war.)

Während der Aktualisierung werden vom Firmware-Update-Vorgang die UPD-Dateien ignoriert, die keinen Stations-Hardwaremodulen entsprechen. Dadurch können Sie eine Memory Card für ein Master-Firmware-Update erstellen, bei dem alle S7-1200 CPU-Stationen in Ihrer Anlage aktualisiert werden. Es wird für die ignorierten UPD-Dateien kein Eintrag im Diagnosepuffer erstellt. Dadurch wird verhindert, dass der Diagnosepuffer durch nicht benötigte, meist bedeutungslose Einträge verstopft wird, welche die Einträge zum Aktualisierungsvorgang, die für Sie interessant sind, möglicherweise verdecken könnten. Beispielsweise wird ein Eintrag in den Diagnosepuffer geschrieben, wenn versucht wird, ein Firmware-Update durchzuführen, ob erfolgreich oder nicht. Sie können dann den Diagnosepuffer mühelos auf unerwartete Anomalien durchsuchen, ohne dass dieser Vorgang durch nicht benötigte Einträge verkompliziert wird.

Alle Versuche, die Firmware zu aktualisieren, werden unabhängig vom Ergebnis im Diagnosepuffer protokolliert. Ist beispielsweise ein Firmware-Update erfolgreich, wird eine

entsprechende Meldung in den Diagnosepuffer geschrieben. Ebenso wird ein fehlgeschlagenes Firmware-Update mit einer entsprechenden Erläuterung protokolliert.

Auf einer Firmware-Update-Karte ist nur eine UPD-Datei einer MLFB (Bestellnummer) zulässig. Findet das Update-Programm beispielsweise UPD-Dateien mit doppelten MLFBs auf der Firmware-Update-Karte, wird kein Firmware-Update durchgeführt und eine entsprechende Meldung in den Diagnosepuffer eingetragen. Dies kann der Fall sein, wenn die UPD-Dateien die gleiche MLFB tragen, jedoch verschiedene Versionen haben.

5.6 Vorgehensweise bei verlorenem Passwort

Wenn Sie das Passwort für eine passwortgeschützte CPU verloren haben, löschen Sie das passwortgeschützte Programm mit einer leeren Übertragungskarte. Die leere Übertragungskarte löscht den internen Ladespeicher der CPU. Dann können Sie ein neues Anwenderprogramm aus STEP 7 in die CPU laden.

Informationen zum Erstellen und Nutzen einer leeren Übertragungskarte finden Sie im Abschnitt Übertragungskarten (Seite 122).



WARNUNG

Vergewissern Sie sich vor dem Einlegen der Memory Card, dass die CPU keinen Prozess ausführt

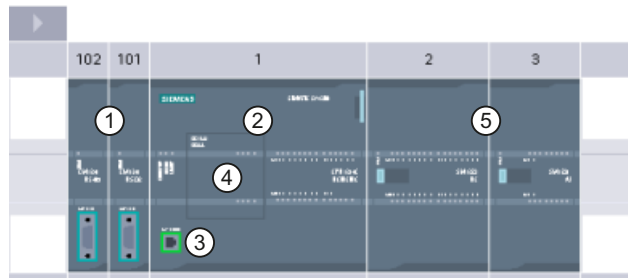
Wenn Sie eine Übertragungskarte in eine laufende CPU stecken, geht die CPU in STOP. Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch den unkontrollierten Betrieb der gesteuerten Geräte verursachen. Daraus resultiert ein unvorhersehbarer Betrieb des Automatisierungssystems, der zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

Stellen Sie vor dem Einlegen einer Übertragungskarte sicher, dass die CPU sich im Betriebszustand STOP und Ihr Prozess sich in einem sicheren Zustand befindet.

Sie müssen die Übertragungskarte ziehen, bevor Sie die CPU in RUN versetzen.

Gerätekonfiguration

Sie können die Gerätekonfiguration für Ihr PLC-Gerät durch Hinzufügen einer CPU und weiterer Module zu Ihrem Projekt erstellen.



- ① Kommunikationsmodul (CM) oder Kommunikationsprozessor (CP): bis zu 3, in Steckplätzen 101, 102 und 103
- ② CPU: Steckplatz 1
- ③ PROFINET-Port der CPU
- ④ Signalboard (SB), Kommunikationsboard (CB) oder Batterieboard (BB): max. 1, in CPU gesteckt
- ⑤ Signalmodul (SM) für digitale oder analoge E/A: bis zu 8, in Steckplätzen 2 bis 9
(8 bei CPU 1214C, CPU 1215C und CPU 1217C, 2 bei CPU 1212C, keines bei der CPU 1211C)

Konfigurationssteuerung

Die Gerätekonfiguration für die S7-1200 unterstützt auch die "Konfigurationssteuerung (Seite 141)", mit der eine Maximalconfiguration für ein Projekt mit Modulen projektiert werden kann, die möglicherweise nicht alle verwendet werden. Mit dieser Funktion, manchmal auch "Optionsverwaltung" genannt, kann ein Maximalausbau projektiert werden, der die Verwendung variabler Module in zahlreichen Anwendungen gestattet.

6.1 Einfügen einer CPU

Eine CPU können Sie entweder in der Portalansicht oder in der Projektansicht von STEP 7 in Ihr Projekt einfügen:

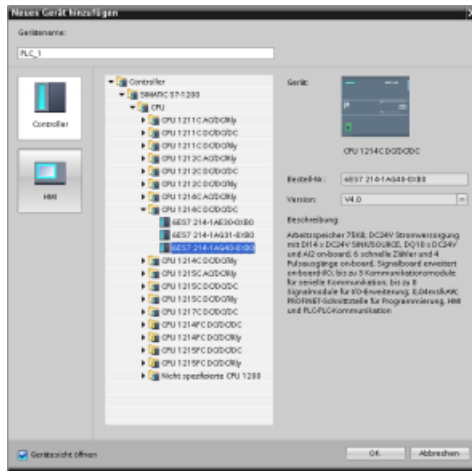
- Wählen Sie in der Portalansicht das Portal "Geräte & Netze" und klicken Sie auf "Neues Gerät hinzufügen".



- Doppelklicken Sie in der Projektansicht unter dem Projektnamen auf "Neues Gerät hinzufügen".



Wählen Sie im Dialog "Neues Gerät hinzufügen" die richtige Variante und Firmware-Version in der Liste aus.

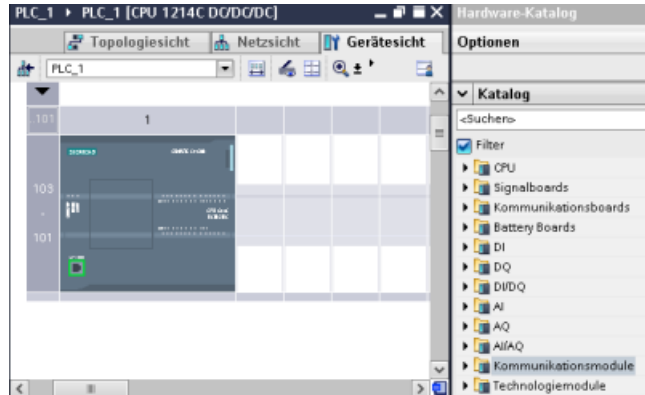


PLC-Security-Einstellungen für Ihre eingefügte CPU konfigurieren

Wenn Sie eine S7-1200 CPU V4.5 einfügen, ruft STEP 7 den Security-Assistenten (Seite 160) auf, in dem Sie Ihre PLC-Security-Einstellungen einrichten können. Folgen Sie den Schritten im Assistenten, um Ihre PLC-Security-Einstellungen festzulegen.

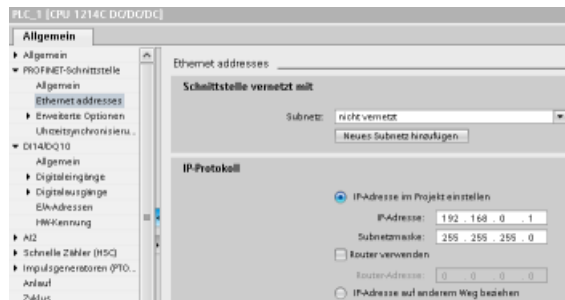
Gerätekonfiguration für die gesteckte CPU

Nachdem Sie eine CPU hinzugefügt haben, erstellt STEP 7 den Baugruppenträger und zeigt die CPU in der Gerätesicht an:



Nach dem Anklicken der CPU in der Gerätesicht werden die Eigenschaften der CPU im Inspektorfenster angezeigt.

Sie können der CPU bei der Gerätekonfiguration eine IP-Adresse zuweisen. Ist Ihre CPU an einen Router im Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.



6.2 Konfiguration aus einer angeschlossenen CPU laden

STEP 7 ermöglicht das Laden der Hardwarekonfiguration einer angeschlossenen CPU auf zweierlei Art:

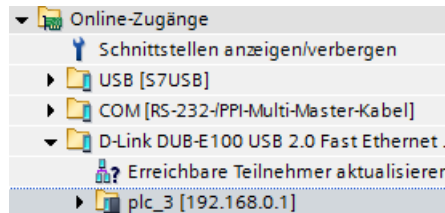
- das Laden des angeschlossenen Geräts als neue Station
- das Konfigurieren einer nicht spezifizierten CPU mittels Hardwareerkennung der angeschlossenen CPU

Zu beachten ist jedoch, dass im ersten Fall sowohl die Hardwarekonfiguration als auch die Software der angeschlossenen CPU geladen werden.

Gerät als neue Station laden

Um ein angeschlossenes Gerät als neue Station zu laden, gehen Sie wie folgt vor:

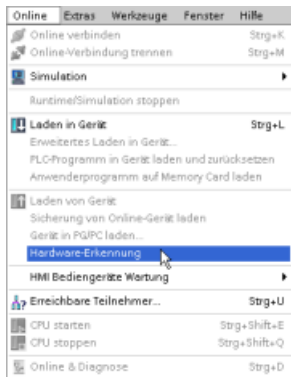
1. Erweitern Sie in der Projektnavigation Ihre Kommunikationsschnittstelle über den Knoten "Online-Zugänge".
2. Doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren".
3. Wählen Sie die PLC aus den erkannten Geräten.



4. Wählen Sie im Online-Menü von STEP 7 Befehl "Gerät als neue Station laden (Hardware und Software)".

STEP 7 lädt die Hardwarekonfiguration und die Programmbausteine.

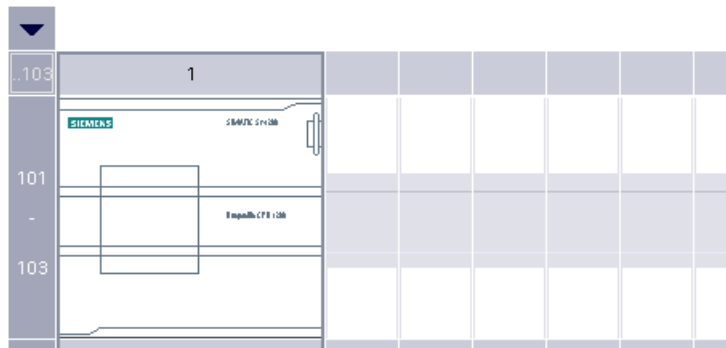
Hardwarekonfiguration einer nicht spezifizierten CPU erkennen



Wenn eine Verbindung zu einer CPU besteht, können Sie die Konfiguration dieser CPU einschließlich evtl. vorhandener Module aus dem Gerät in Ihr Projekt laden. Legen Sie dazu einfach ein neues Projekt an und wählen Sie anstelle einer bestimmten CPU die "nicht spezifizierte CPU". (Sie können auch die Gerätekonfiguration ganz umgehen, indem Sie unter "Erste Schritte" auf "Ein PLC-Programm erstellen" klicken. STEP 7 legt dann automatisch eine nicht spezifizierte CPU an.)

Wählen Sie im Programmiereditor im Menü "Online" den Befehl "Hardwareerkennung".

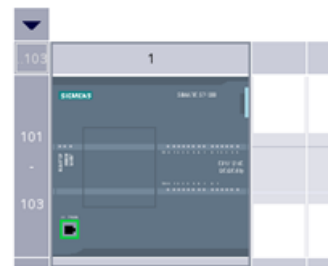
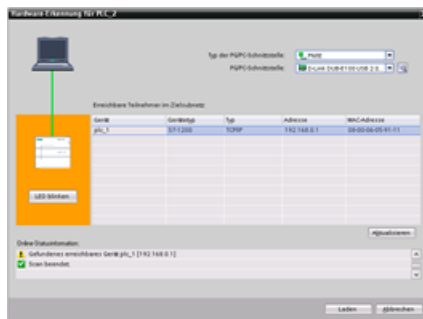
Wählen Sie im Gerätekonfigurationseditor die Option zum Erkennen der Konfiguration des angeschlossenen Geräts.



Das Gerät ist nicht spezifiziert.

- Bitte verwenden Sie den [Hardware-Katalog](#) um die CPU zu spezifizieren,
- oder [ermitteln](#) Sie die Konfiguration des angeschlossenen Gerätes.

Nachdem Sie im Online-Dialog die CPU ausgewählt und auf die Schaltfläche zum Laden geklickt haben, lädt STEP 7 die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Sie können dann die Parameter für die CPU und die Module (Seite 153) konfigurieren.



6.3 Module zur Konfiguration hinzufügen

Im Hardwarekatalog können Sie Module zur CPU hinzufügen:

- Signalmodule (SMs) für zusätzliche digitale oder analoge Ein- und Ausgänge. Diese Module werden an der rechten Seite der CPU angeschlossen.
- Signalboards (SBs) bieten eine begrenzte Zahl von zusätzlichen Ein-/Ausgängen für die CPU. Das SB wird auf der Vorderseite der CPU gesteckt.
- Das Batterieboard 1297 (BB) bietet eine Langzeitpufferung der Echtzeituhr. Das BB wird auf der Vorderseite der CPU gesteckt.

6.3 Module zur Konfiguration hinzufügen

- Kommunikationsboards (CBs) bieten einen zusätzlichen Kommunikationsanschluss (z. B. RS485). Das CB wird auf der Vorderseite der CPU gesteckt.
- Kommunikationsmodule (CMs) und Kommunikationsprozessoren (CPs) bieten einen zusätzlichen Kommunikationsanschluss, z. B. für PROFIBUS oder GPRS. Diese Module werden an der linken Seite der CPU angeschlossen.

Um ein Modul in die Gerätekonfiguration einzufügen, wählen Sie das Modul im Hardwarekatalog aus und klicken doppelt darauf oder ziehen Sie es in den markierten Steckplatz. Sie müssen die Module in die Gerätekonfiguration aufnehmen und die Hardwarekonfiguration in die CPU laden, damit die Module funktionsfähig sind.

Tabelle 6-1 Modul zur Gerätekonfiguration hinzufügen

Modul	Modul auswählen	Modul einsetzen	Ergebnis
SM			
SB, BB oder CB			
CM oder CP			

Mit der Funktion "Konfigurationssteuerung" (Seite 141) können Signalmodule und Signalboards in der Gerätekonfiguration hinzugefügt werden, die möglicherweise nicht der aktuellen Hardware für eine bestimmte Anwendung entsprechen, aber in anderen Anwendungen verwendet werden, die das gleiche Anwenderprogramm, CPU-Modell und u. U. auch einige der konfigurierten Module teilen.

6.4 Konfigurationssteuerung

6.4.1 Vorteile und Nutzung der Konfigurationssteuerung

Die Konfigurationssteuerung kann hilfreich sein, wenn eine Automatisierungslösung entwickelt wird, die mit Variationen in unterschiedlichen Aufbauten eingesetzt werden soll.

Eine STEP 7-Gerätekonfiguration und ein Anwenderprogramm können in verschiedene installierte PLC-Konfigurationen geladen werden. Dafür sind lediglich einige wenige Anpassungen des STEP 7-Projekts für den neuen Aufbau erforderlich.

6.4.2 Den zentralen Aufbau und optionale Module konfigurieren

Die Konfigurationssteuerung in STEP 7 und die S7-1200 ermöglichen die Projektierung einer Maximalkonfiguration für ein Standardsystem mit unterschiedlichen Versionen (Optionen), die jeweils eine Teilmenge dieser Konfiguration nutzen. Im Handbuch PROFINET mit STEP 7 (<https://support.industry.siemens.com/cs/ww/de/view/49948856>) werden diese Projekttypen als "Serienmaschinen-Projekte" bezeichnet.

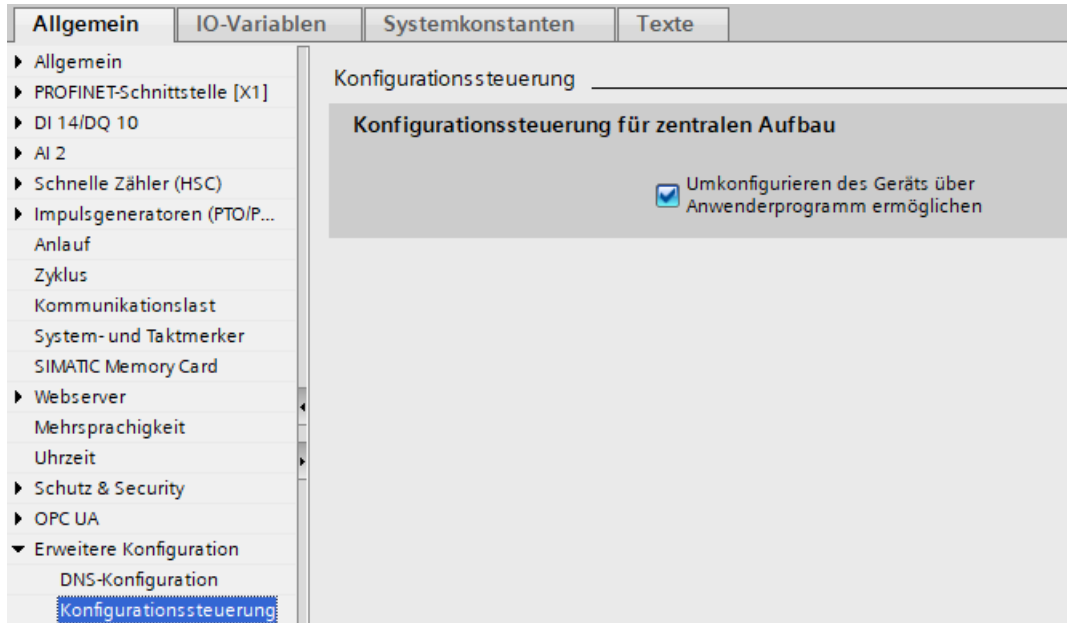
Ein Steuerdatensatz, der im Anlauf-OB programmiert wird, teilt der CPU mit, welche Module im realen Aufbau abweichend von der Projektierung fehlen oder welche Module sich abweichend von der Projektierung in einem anderen Steckplatz befinden. Die Konfigurationssteuerung hat keine Auswirkung auf die Parametrierung der Module.

Mit der Konfigurationssteuerung kann der Aufbau flexibel verändert werden, solange der reale Aufbau in der maximalen Gerätekonfiguration in STEP 7 enthalten ist.

Gehen Sie wie folgt vor, um die Konfigurationssteuerung zu aktivieren und den erforderlichen Steuerdatensatz zu strukturieren:

1. Optional kann die CPU auf die Werkseinstellungen zurückgesetzt werden, um sicherzustellen, dass kein inkompatibler Steuerdatensatz in der CPU vorhanden ist.
2. Wählen Sie die CPU in der Gerätekonfiguration in STEP 7.

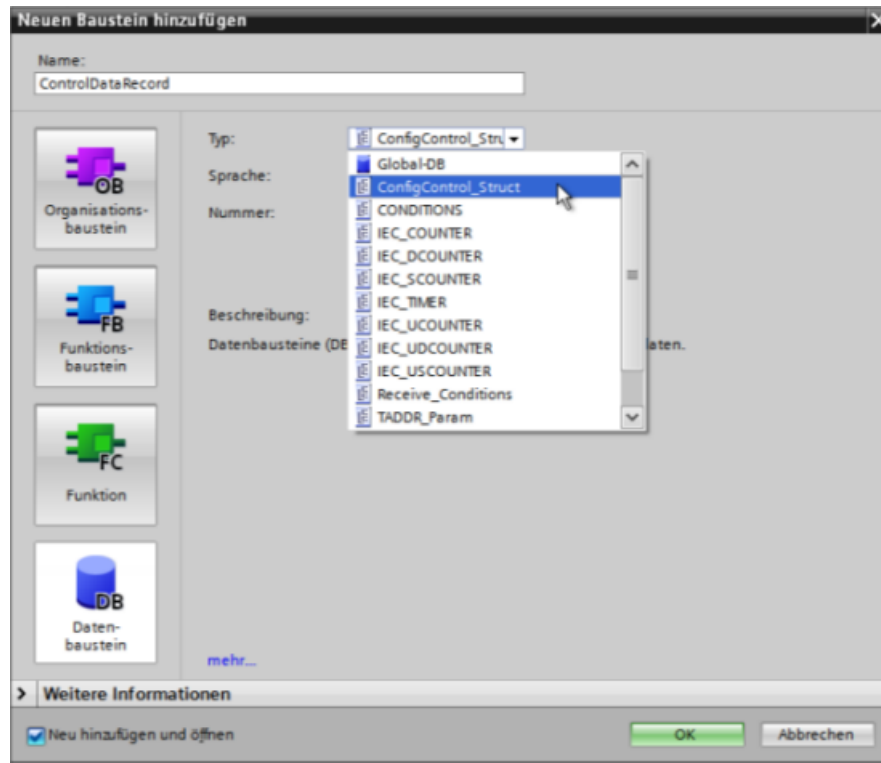
3. Im Teilnehmer "Konfigurationssteuerung" in den CPU-Eigenschaften aktivieren Sie Option "Umkonfigurieren des Geräts über Anwenderprogramm ermöglichen".



4. Erstellen Sie einen PLC-Datentyp für den Steuerdatensatz. Konfigurieren Sie ihn als Struct mit vier USInts für die Konfigurationssteuerungsdaten und weiteren USInts entsprechend der Anzahl der Steckplätze für den S7-1200 Maximalausbau wie folgt:

ConfigControl_Struct				
	Name	Datentyp	Defaultwert	Kommentar
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. Legen Sie einen Datenbaustein mit dem gewählten PLC-Datentyp an.



6. Konfigurieren Sie in diesem Datenbaustein die Parameter Block_length, Block_ID, Version und Subversion wie unten gezeigt. Konfigurieren Sie die Werte für die Steckplätze je nachdem, ob sie belegt sind oder nicht und nach ihrer Lage im tatsächlichen Aufbau:
 - 0: Konfiguriertes Modul in der tatsächlichen Konfiguration nicht vorhanden. (Steckplatz ist leer.)
 - 1 bis 9, 101 bis 103: Tatsächliche Position des konfigurierten Steckplatzes
 - 255: Die STEP 7-Gerätekonfiguration enthält kein Modul an diesem Steckplatz.

Hinweis

Konfigurationssteuerung nicht verfügbar für HSCs und PTOs auf dem Signalboard

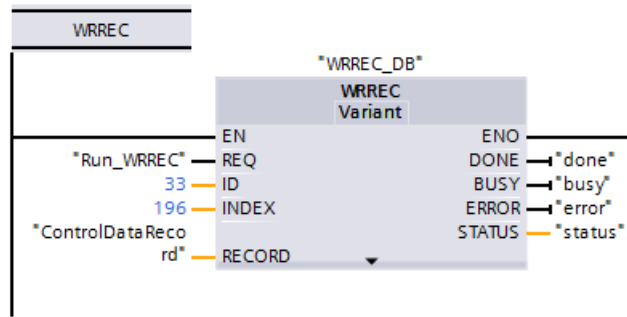
Wenn Sie in der CPU ein Signalboard haben, das Sie für HSCs oder PTOs konfigurieren, dürfen Sie dieses Signalboard nicht mit einer "0" in Steckplatz_1 des Steuerdatensatzes für die Konfiguration deaktivieren. Konfigurierte HSC- und PTO-Geräte der CPU sind für die Konfigurationssteuerung obligatorisch.

ControlDataRecord					
	Name	Datentyp	Startwert	Kommentar	
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	

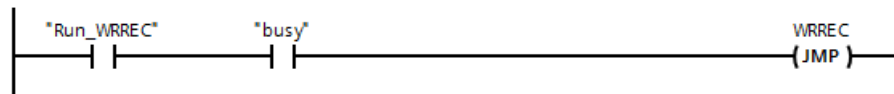
Zur Zuweisung der Werte für die Steckplätze, siehe Beispiel für die Konfigurationssteuerung (Seite 148).

7. Rufen Sie im Anlauf-OB die erweiterte Anweisung WRREC (Datensatz schreiben) auf, um den Steuerdatensatz zu Index 196 der Hardware-ID 33 zu übertragen. Verwenden Sie eine Sprungmarke (JMP), um zu warten, bis die Anweisung WRREC beendet ist.

Netzwerk 1:



Netzwerk 2:



Hinweis

Die Konfigurationssteuerung ist erst wirksam, wenn die Anweisung WRREC den Steuerdatensatz in den Anlauf-OB übertragen hat. Ist die Konfigurationssteuerung aktiviert und der Steuerdatensatz in der CPU nicht vorhanden, geht diese nach dem Anlauf direkt in STOP. Daher ist es wichtig, dass der Anlauf-OB so programmiert ist, dass der Steuerdatensatz übertragen wird.

Modulanordnung

Die folgende Tabelle zeigt die Zuweisung der Steckplätze:

Steckplatz	Module
1	Signalboard oder Kommunikationsboard (CPU-Zusatzkarte)
2 bis 9	Signalmodule
101 bis 103	Kommunikationsmodule

Steuerdatensatz

Ein Steuerdatensatz 196 enthält die Steckplatzbelegung und die tatsächliche Konfiguration wie unten gezeigt:

Byte	Element	Wert	Erklärung
0	Bausteinlänge	16	Header
1	Baustein-ID	196	
2	Version	5	
3	Subversion	0	

Byte	Element	Wert	Erklärung
4	Zuweisung einer CPU-Zusatzkarte	Tatsächliche Zusatzkarte, 0 oder 255*	Steuerelement Beschreibt in jedem Element, welcher reale Steckplatz im Gerät dem konfigurierten Steckplatz zugewiesen ist.
5	Zuweisung konfigurierter Steckplatz 2	Tatsächlicher Steckplatz, 0 oder 255*	
...	
12	Zuweisung konfigurierter Steckplatz 9	Tatsächlicher Steckplatz, 0 oder 255*	Anders als bei Signalmodulen muss der tatsächliche Steckplatz bei physisch vorhandenen Kommunikationsmodulen mit dem konfigurierten Steckplatz übereinstimmen.
13	Zuweisung konfigurierter Steckplatz 101	Tatsächlicher Steckplatz oder 255*	
14	Zuweisung konfigurierter Steckplatz 102	Tatsächlicher Steckplatz oder 255*	
15	Zuweisung konfigurierter Steckplatz 103	Tatsächlicher Steckplatz oder 255*	

***Steckplatzwerte:**

0: Konfiguriertes Modul in der tatsächlichen Konfiguration nicht vorhanden. (Steckplatz ist leer.)

1 bis 9, 101 bis 103: Tatsächliche Position des konfigurierten Steckplatzes

255: Die STEP 7-Gerätekonfiguration enthält kein Modul an diesem Steckplatz.

Hinweis**Alternative zum Erstellen eines PLC-Variablentyps**

Als Alternative zum Erstellen eines benutzerdefinierten PLC-Variablentyps können Sie einen Datenbaustein direkt mit sämtlichen Strukturelementen eines Steuerdatensatzes erstellen. Sie könnten sogar mehrere Strukturen in diesem Datenbaustein konfigurieren, die als verschiedene Steuerdatensatzkonfigurationen dienen. Beide Implementierungen sind ein effektiver Weg, um den Steuerdatensatz während des Anlaufs zu übertragen.

Regeln

Die folgenden Regeln sind zu beachten:

- Die Konfigurationssteuerung unterstützt keine Änderung der Position für Kommunikationsmodule. Sie können auch die Konfigurationssteuerung nicht dazu verwenden, Kommunikationsmodule zu deaktivieren. Die im Steuerdatensatz angegebenen Positionen der Steckplätze 101 bis 103 müssen dem tatsächlichen Aufbau entsprechen. Ist für den Steckplatz in Ihrer Gerätekonfiguration kein Modul konfiguriert, geben Sie für diese Position den Wert 255 in den Steuerdatensatz ein. Ist für den Steckplatz ein Modul konfiguriert, geben Sie den konfigurierten Steckplatz als tatsächlichen Steckplatz für diese Position ein.
- F-E/A-Module unterstützen die Konfigurationssteuerung nicht. Die Steckplatzpositionen für ein F-E/A-Modul im Steuerdatensatz müssen der konfigurierten Steckplatzposition für das F-E/A-Modul entsprechen. Wenn Sie versuchen, ein konfiguriertes F-E/A-Modul mit dem Steuerdatensatz zu verschieben oder zu löschen, melden alle tatsächlich installierten F-E/A-Module einen Parametrierungsfehler und gestatten den Austausch nicht.

- Es sind keine unbenutzten Steckplätze zwischen belegten Steckplätzen zulässig. Ist in der tatsächlichen Konfiguration beispielsweise ein Modul in Steckplatz 4 vorhanden, müssen auch Module in den Steckplätzen 2 und 3 gesteckt sein. Ist in der tatsächlichen Konfiguration ein Kommunikationsmodul in Steckplatz 102 vorhanden, muss dementsprechend auch ein Modul in Steckplatz 101 gesteckt sein.
- Ist die Konfigurationssteuerung aktiviert, ist die CPU erst betriebsbereit, wenn ein Steuerdatensatz geladen wurde. Die CPU wechselt aus dem Anlauf in STOP, wenn der Anlauf-OB keinen gültigen Steuerdatensatz überträgt. Die zentralen E/A werden dann von der CPU nicht initialisiert und der Grund für den Wechsel in STOP wird in den Diagnosepuffer geschrieben.
- Ein erfolgreich übertragener Steuerdatensatz wird von der CPU im remanenten Speicher abgelegt; der Steuerdatensatz 196 muss daher nach einem Neustart nicht neu geschrieben werden, sofern die Konfiguration nicht geändert wurde.
- Jeder reale Steckplatz darf nur einmal im Steuerdatensatz enthalten sein.
- Jedem konfigurierten Steckplatz kann nur ein realer Steckplatz zugewiesen werden.

Hinweis

Ändern einer Konfiguration

Wird ein Steuerdatensatz mit einer geänderten Konfiguration geschrieben, wird in der CPU der folgende Ablauf automatisch angestoßen: Urlöschen und anschließender Neustart mit der geänderten Konfiguration.

Die CPU löscht daraufhin den ursprünglichen Steuerdatensatz und speichert den neuen Datensatz im remanenten Speicher.

Verhalten im Betrieb

Für die Online-Anzeige und die Anzeige im Diagnosepuffer (Modul OK oder Modul fehlerhaft) nutzt STEP 7 die Gerätekonfiguration, nicht die abweichende reale Konfiguration.

Beispiel: Ein Modul gibt Diagnosedaten aus. Dieses Modul steckt laut Konfiguration in Steckplatz 4, tatsächlich aber in Steckplatz 3. In der Online-Ansicht wird ein Fehler in Steckplatz 4 angezeigt. In der realen Konfiguration meldet das Modul in Steckplatz 3 einen Fehler an seiner LED-Anzeige.

Wurden im Steuerdatensatz Module als fehlend (0) konfiguriert, verhält sich das Automatisierungssystem wie folgt:

- Module, die im Steuerdatensatz als nicht vorhanden konfiguriert sind, melden keine Diagnosedaten und ihr Status ist immer OK. Der Wertstatus ist OK.
- Ein direkter Schreibzugriff auf die Ausgänge oder Schreibzugriff auf das Prozessbild von Ausgängen, die nicht vorhanden sind, hat keine Wirkung; die CPU meldet keinen Zugriffsfehler.
- Ein direkter Lesezugriff auf die Eingänge oder Lesezugriff auf das Prozessbild von Eingängen, die nicht vorhanden sind, bewirkt einen Wert "0" für jeden Eingang; die CPU meldet keinen Zugriffsfehler.

- Das Schreiben eines Datensatzes in ein Modul, das nicht vorhanden ist, hat keine Wirkung; die CPU meldet keinen Fehler.
- Wird versucht, einen Datensatz aus einem nicht vorhandenen Modul zu lesen, wird eine Fehlermeldung erzeugt, weil die CPU keinen gültigen Datensatz zurückliefern kann.

Fehlermeldungen

Die CPU gibt die folgenden Fehlermeldungen zurück, wenn beim Schreiben des Steuerdatensatzes ein Fehler auftritt:

Fehlercode	Bedeutung
16#80B1	Ungültige Länge; im Steuerdatensatz ist eine falsche Länge angegeben.
16#80B5	Konfigurationssteuerungsparameter nicht zugewiesen
16#80E2	Der Datensatz wurde im falschen OB-Kontext übertragen. Der Datensatz muss im Anlauf-OB übertragen werden.
16#80B0	Bausteintyp (Byte 2) des Steuerdatensatzes ist nicht 196.
16#80B8	Parameterfehler; Modul meldet ungültige Parameter, zum Beispiel: <ul style="list-style-type: none"> • Der Steuerdatensatz versucht, die Konfiguration eines Kommunikationsmoduls oder einer Kommunikationszusatzkarte zu ändern. Die tatsächliche Konfiguration für Kommunikationsmodule und eine Kommunikationszusatzkarte muss mit der Konfiguration in STEP 7 übereinstimmen. • Der zugewiesene Wert für einen nicht konfigurierten Steckplatz im STEP 7-Projekt ist nicht 255. • Der zugewiesene Wert für einen konfigurierten Steckplatz liegt nicht im zulässigen Bereich. • Die zugewiesene Konfiguration hat einen leeren "internen" Steckplatz, z. B. Steckplatz n ist zugewiesen und Steckplatz n-1 ist nicht zugewiesen.

6.4.3 Beispiel für die Konfigurationssteuerung

Dieses Beispiel beschreibt eine Konfiguration bestehend aus einer CPU und drei E/A-Modulen. Das Modul an Steckplatz 3 ist im ersten tatsächlichen Aufbau nicht vorhanden und wird daher mit Hilfe der Konfigurationssteuerung "verborgen".

Im zweiten Aufbau wird das anfangs verborgene Modul zwar verwendet, aber nun im letzten Steckplatz konfiguriert. Ein geänderter Steuerdatensatz überträgt die Informationen über die Zuordnung der Module zu den Steckplätzen.

Beispiel: Tatsächlicher Aufbau mit einem konfigurierten, aber nicht benutzten Modul

Die Gerätekonfiguration enthält alle Module, die in einem tatsächlichen Aufbau vorhanden sein können (Maximalausbau). In diesem Fall ist das in der Gerätekonfiguration Steckplatz 3 zugewiesene Modul im realen Aufbau nicht vorhanden.



Bild 6-1 Gerätekonfiguration für Maximalausbau mit drei Signalmodulen



Bild 6-2 Tatsächlicher Aufbau mit fehlendem Modul für Steckplatz 3 und dem für Steckplatz 4 konfigurierten Modul in Steckplatz 3

Um anzuzeigen, dass ein Modul fehlt, muss Steckplatz 3 im Steuerdatensatz mit 0 konfiguriert werden.

ControlDataRecord				
	Name	Datentyp	Startwert	Kommentar
1	▼ Static			
2	▼ ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ca.
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	0	Configured slot 3 / Assigned "real" slot
10	Slot_4	USInt	3	Configured slot 4 / Assigned "real" slot
11	Slot_5	USInt	255	Configured slot 5 / Assigned "real" slot
12	Slot_6	USInt	255	Configured slot 6 / Assigned "real" slot
13	Slot_7	USInt	255	Configured slot 7 / Assigned "real" slot
14	Slot_8	USInt	255	Configured slot 8 / Assigned "real" slot
15	Slot_9	USInt	255	Configured slot 9 / Assigned "real" slot
16	Slot_101	USInt	255	Configured slot 101 / Assigned "real" slot
17	Slot_102	USInt	255	Configured slot 102 / Assigned "real" slot
18	Slot_103	USInt	255	Configured slot 103 / Assigned "real" slot

Beispiel: Tatsächlicher Aufbau mit einem Modul, das nachträglich einem anderen Steckplatz hinzugefügt wurde

Im zweiten Beispiel ist das Modul in Steckplatz 3 der Gerätekonfiguration im tatsächlichen Aufbau vorhanden, es befindet sich jedoch in Steckplatz 4.

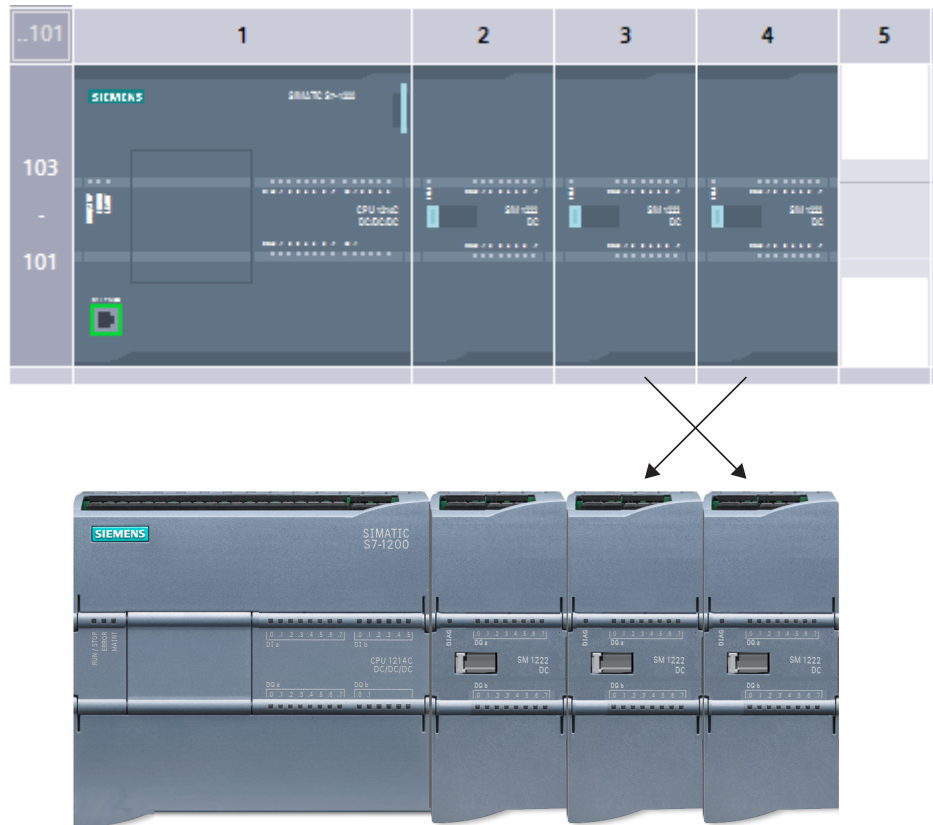


Bild 6-3 Module in Steckplätzen 3 und 4 aus Gerätekonfiguration im tatsächlichen Aufbau vertauscht
Um die Gerätekonfiguration an den tatsächlichen Aufbau anzupassen, sind die Module durch Bearbeiten des Steuerdatensatzes den richtigen Steckplätzen zuzuordnen.

ControlDataRecord				
	Name	Datentyp	Startwert	Kommentar
1	Static			
2	ConfigControl	Struct		
3	Block_length	USInt	16	Length of control data record, including header
4	Block_ID	USInt	196	Data record number
5	Version	USInt	5	
6	Subversion	USInt	0	
7	Slot_1	USInt	255	Assignment for CPU annex card/Actual annex ca..
8	Slot_2	USInt	2	Configured slot 2 / Assigned "real" slot
9	Slot_3	USInt	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 Ändern eines Geräts

Sie können den Gerätetyp einer konfigurierten CPU oder eines Moduls ändern. Klicken Sie in der Gerätekonfiguration mit der rechten Maustaste auf das Gerät und wählen Sie im Kontextmenü den Befehl "Gerät ändern". Wählen Sie im Dialog die CPU oder das Modul aus, die/das Sie ersetzen möchten. Im Dialog "Gerät ändern" werden Ihnen Kompatibilitätsinformationen für die beiden betroffenen Geräte angezeigt.

Wenn Sie die CPU-Version von Geräten wechseln möchten, finden Sie weitere Informationen hierzu unter Ersatz einer CPU V3.0 durch eine CPU V4.x (Seite 1456).

6.6 Konfigurieren des CPU-Betriebs

6.6.1 Übersicht

Um die Betriebsparameter der CPU zu konfigurieren, wählen Sie die CPU in der Gerätesicht aus (blauer Rahmen um die gesamte CPU) und öffnen dann im Inspektorfenster das Register "Eigenschaften".

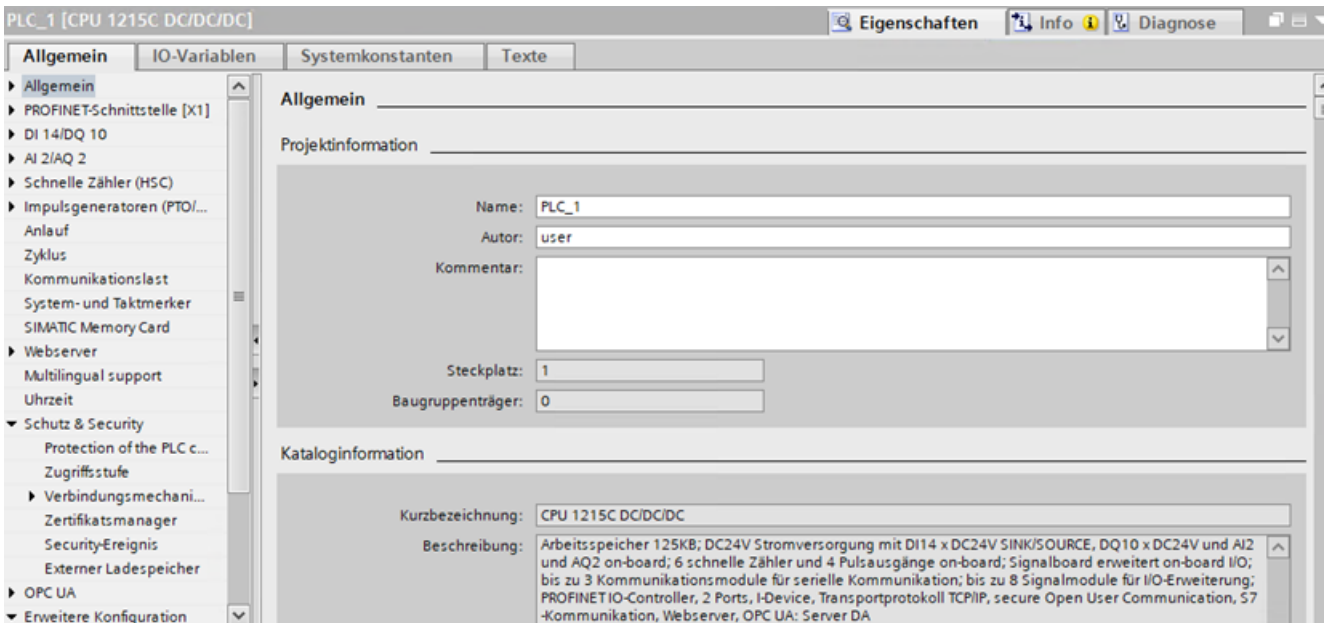


Tabelle 6-2 CPU-Eigenschaften

Eigenschaft	Beschreibung
PROFINET-Schnittstelle	Einstellen der IP-Adresse für die CPU und die Uhrzeitsynchronisation
DE, DA und AE	Einstellen des Verhaltens der lokalen (integrierten) digitalen und analogen E/A (z.B. Filterzeiten für die Digitaleingänge und die Reaktion von Digitalausgängen auf das Versetzen der CPU in den Betriebszustand STOP).
Schnelle Zähler (Seite 543) und Impulsgeneratoren (Seite 475)	Aktivieren und Einstellen der schnellen Zähler (HSC) und der Impulsgeneratoren für die Impulsfolgen (PTO) und die Impulsdauermodulation (PWM) Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Anwenderprogramm verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Eigenschaft	Beschreibung
Anlauf (Seite 71)	<p>Anlauf nach NETZ-EIN: Einstellen des Verhaltens der CPU nach dem Aus- und Wiedereinschalten, z. B. für das Anlaufen im Betriebszustand STOP oder den Wechsel in RUN nach einem Warmstart</p> <p>Hardwarekompatibilität: Konfiguration der Ersatzstrategie für alle Systemkomponenten (SM, SB, CM, CP und CPU):</p> <ul style="list-style-type: none"> • Akzeptablen Ersatz zulassen • Beliebigen Ersatz zulassen (Standard) <p>Jedes Modul enthält interne Anforderungen an die Ersatzkompatibilität basierend auf der Anzahl der E/A, der elektrischen Verträglichkeit und anderen entsprechenden Vergleichspunkten. Beispiel: Ein SM mit 16 Kanälen kann ein akzeptabler Ersatz für ein SM mit 8 Kanälen sein, doch ein SM mit 8 Kanälen kann kein akzeptabler Ersatz für ein SM mit 16 Kanälen sein. Wenn Sie "Akzeptablen Ersatz zulassen" wählen, setzt STEP 7 die Ersetzungsregeln um. Ansonsten gestattet STEP 7 jeden Ersatz.</p> <p>Zuweisungszeit der Parameter für dezentrale E/A: Konfiguriert einen maximalen Zeitraum (Voreinstellung: 60000 ms), in dem die dezentrale Peripherie online gebracht werden muss. (Die CMs und CPs werden während des Anlaufs von der CPU mit Spannung und Kommunikationsparametern versorgt. Diese Zuweisungszeit gestattet einen Zeitraum, während dessen die an CM oder CP angeschlossenen E/A-Module online gebracht werden müssen.)</p> <p>Die CPU geht in RUN, sobald die dezentrale Peripherie online ist, unabhängig von der Zuweisungszeit. Wurde die dezentrale Peripherie während dieses Zeitraums nicht online gebracht, geht die CPU trotzdem in RUN, ohne die dezentrale Peripherie.</p> <p>Hinweis: Wenn Ihre Konfiguration ein CM 1243-5 (PROFIBUS-Master) enthält, legen Sie für diesen Parameter keinen Wert unter 15 Sekunden (15000 ms) fest, um sicherzustellen, dass das Modul online gebracht werden kann.</p> <p>OBs müssen unterbrechbar sein: Legt fest, ob die OB-Ausführung (sämtlicher OBs) in der CPU unterbrechbar oder nicht unterbrechbar (Seite 86) ist.</p>
Zyklus (Seite 90)	Festlegen einer maximalen Zykluszeit oder einer festen Mindestzykluszeit
Kommunikationslast	Zuweisen eines prozentualen Anteils der CPU-Zeit für Kommunikationsaufgaben
System- und Taktmerker (Seite 94)	Setzen eines Bytes für die Systemmerkerfunktionen und Setzen eines Bytes für die Taktmerkerfunktionen (dabei schaltet jedes Bit in einer vorgegebenen Frequenz ein und aus).
Webserver (Seite 847)	Aktiviert und konfiguriert die Webserverfunktion
Tageszeit	Auswählen der Zeitzone und Einstellen der Sommer-/Winterzeit
Mehrsprachiger Support (Seite 158)	Weist eine Projektsprache für den Webserver zu. Diese Sprache wird zum Anzeigen von Diagnosepuffermeldungen in der Benutzeroberfläche des Webserver verwendet.
Schutz (Seite 163)	Einstellen des Lese-/Schreibschutzes und der Passwörter für den Zugriff auf die CPU
Konfigurationssteuerung (Seite 141)	Ermöglicht die Konfiguration einer Mastergerätekonfiguration, die Sie für verschiedene tatsächliche Gerätekonfigurationen steuern können.
Verbindungsressourcen (Seite 595)	Bietet einen Überblick über die Kommunikationsverbindungsressourcen der CPU und die Anzahl der konfigurierten Verbindungsressourcen.
Übersicht über Adressen	Bietet einen Überblick über die E/A-Adressen, die für die CPU konfiguriert sind.

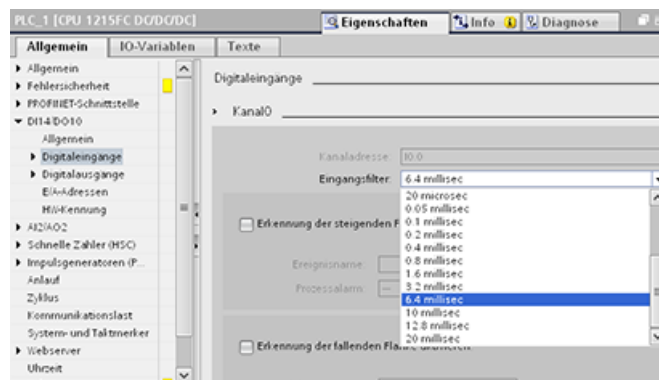
6.6.2 Filterzeiten für Digitaleingänge einrichten

Die Filter der digitalen Eingänge schützen Ihr Programm davor, auf unerwünschte schnelle Veränderungen der Eingangssignale zu reagieren, wie sie z. B. durch Kontaktprellen oder Rauschen verursacht werden können. Die voreingestellte Filterzeit von 6,4 ms verhindert unerwünschte Schalthandlungen durch typische mechanische Kontakte. Bestimmte Funktionen Ihrer Anwendung können kürzere Filterzeiten für die Erkennung und Reaktion auf Eingänge schneller Sensoren erfordern oder längere Filterzeiten für die Ausfilterung langsamer Kontaktpreller oder längeren Impulsrauschens erfordern.

Eine Eingangsfilterszeit von 6,4 ms bedeutet, dass ein einziger Signalwechsel von '0' auf '1' oder von '1' auf '0' etwa 6,4 ms lang anstehen muss, um erkannt zu werden, und dass einzelne hohe oder niedrige Impulse, die kürzer als 6,4 ms anstehen, nicht erkannt werden. Schaltet ein Eingangssignal schneller als die eingestellte Filterzeit zwischen '0' und '1', so kann sich der Wert des Eingangs im Anwenderprogramm ändern, wenn die Gesamtdauer der Impulse mit neuen Werten gegenüber den alten Werten die Filterzeit übersteigt.


Der Filter für Digitaleingänge funktioniert wie folgt:

- Wird "1" eingegeben, wird die Zeit hochgezählt und bei Erreichen der Filterzeit angehalten. Der Punkt im Prozessabbild wechselt von "0" auf "1", wenn die Filterzeit erreicht wird.
- Wird "0" eingegeben, wird die Zeit heruntergezählt und bei Erreichen von "0" angehalten. Der Punkt im Prozessabbild wechselt von "1" auf "0", wenn ein Wert "0" erreicht wird.
- Pendelt der Eingang, wird die Zeit teils hoch-, teils heruntergezählt. Das Prozessabbild ändert sich, wenn die Nettosumme der Zählwerte entweder die Filterzeit oder "0" erreicht.
- Ein schnell veränderliches Signal mit mehr "0" als "1" geht dann möglicherweise auf "0"; sind mehr "1" als "0" vorhanden, wechselt das Prozessabbildregister möglicherweise auf "1".



Jeder Eingang hat eine Filterkonfiguration, die für alle Zwecke gilt: Prozesseingänge, Alarmer, Impulsabgriff und HSC-Eingänge. Um die Zeiten für den Eingangsfilter einzurichten, wählen Sie "Digitaleingänge" aus.

Die Standardfilterzeit für die digitalen Eingänge beträgt 6,4 ms. In der Klappliste "Eingangsfiler" können Sie eine Filterzeit auswählen. Gültige Filterzeiten liegen zwischen 0,1 µs und 20,0 ms.

 WARNUNG
<p>Gefahren beim Ändern der Filterzeit für digitale Eingangskanäle</p> <p>Wenn Sie die bisherige Einstellung der Filterzeit eines Digitaleingangskanals ändern, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel "0" vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.</p> <p>Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.</p> <p>Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.</p>

Konfiguration von Filterzeiten für als HSCs verwendete Digitaleingänge

Bei Eingängen, die als schnelle Zähler (HSCs) verwendet werden, sollten Sie die Eingangsfilerzeit auf einen geeigneten Wert einstellen, mit dem Sie Zählausfälle vermeiden.

Siemens empfiehlt folgende Einstellungen:

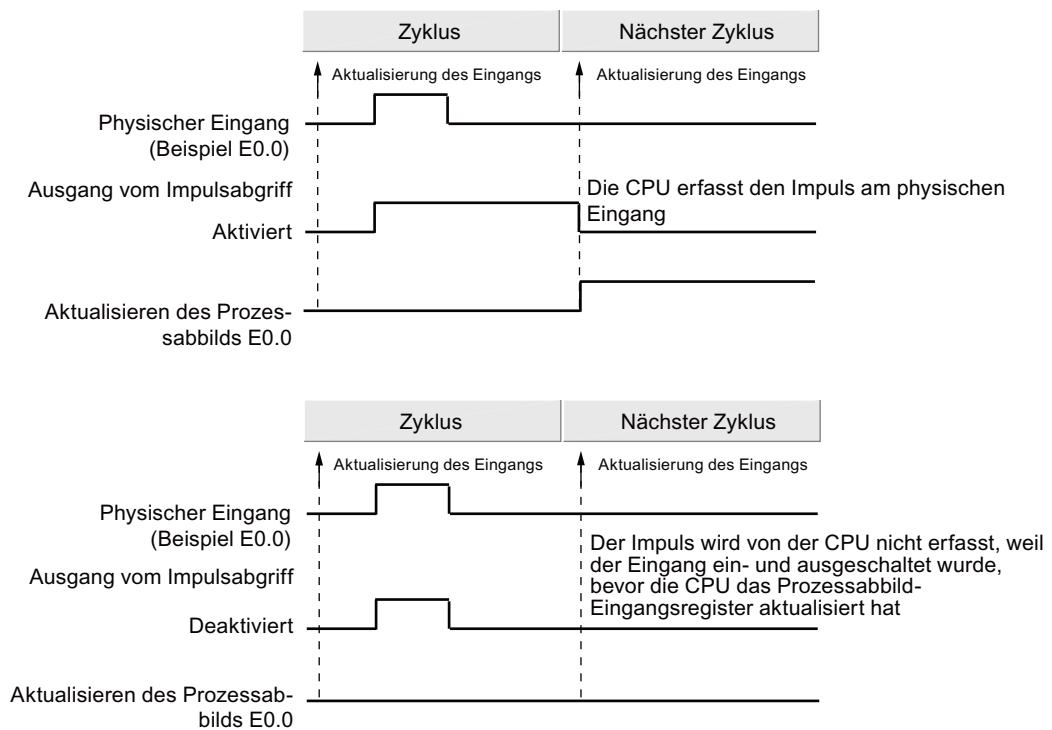
Typ des schnellen Zählers	Empfohlene Eingangsfilerzeit
1 MHz	0,1 Mikrosekunden
100 kHz	0,8 Mikrosekunden
30 kHz	3,2 Mikrosekunden

6.6.3 Impulsabgriff

Die S7-1200 CPU bietet den Impulsabgriff für digitale Eingänge. Mit der Funktion "Impulsabgriff" können Sie hohe oder niedrige Impulse erfassen, die eine so kurze Dauer haben, dass sie von der CPU leicht übersehen werden können, wenn die Digitaleingänge zu Beginn eines Zyklus gelesen werden.

Wenn Sie den Impulsabgriff für einen Eingang aktivieren, wird ein Zustandswechsel des Eingangs bis zur nächsten Aktualisierung des Eingangs gespeichert und gehalten. Damit wird sichergestellt, dass ein nur kurz anstehender Impuls erfasst und gehalten wird, bis die CPU die Eingänge liest.

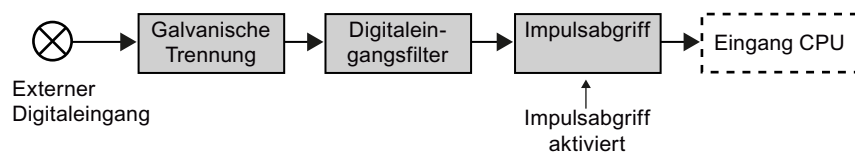
Die folgenden Abbildungen zeigen das Verhalten der S7-1200 CPU mit und ohne aktivierten Impulsabgriff:



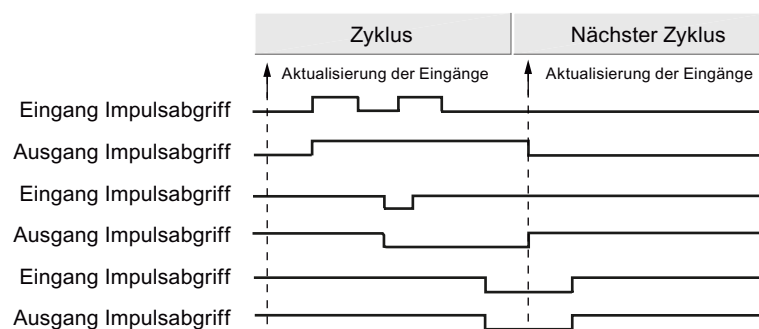
Hinweis

Da die Impulsabgrifffunktion auf den Eingang erst nach dem Passieren des Eingangsfilters wirkt, muss die Eingangsfilterszeit so eingestellt werden, dass der Impuls nicht ausgefiltert wird.

Das folgende Bild zeigt eine schematische Darstellung des Digitaleingangskreises:



Das folgende Bild zeigt die Reaktion des aktivierten Impulsabgriffs auf verschiedene Eingangsbedingungen. Gibt es mehr als einen Impuls in einem bestimmten Zyklus, wird nur der erste Impuls gelesen. Bei mehreren Impulsen in einem Zyklus sollten Sie die Interruptereignisse für steigende/fallende Flanken einsetzen:



6.7 Mehrsprachigen Support konfigurieren

Die Einstellungen für den mehrsprachigen Support ermöglichen Ihnen das Zuweisen von zwei Projektsprachen für jede Oberflächensprache des S7-1200 Webservers (Seite 847). Es besteht auch die Möglichkeit, keine Projektsprache für die Sprache einer Benutzeroberfläche festzulegen.

Was ist eine Projektsprache?

Die Projektsprache ist die Sprache, in der das TIA Portal benutzerdefinierte Projekttexte wie Netzwerkkommentare und Bausteinkommentare anzeigt.

Sie wählen für das in der Projektnavigation markierte Projekt die Projektsprachen im TIA Portal über den Menübefehl **Werkzeuge > Projektsprachen** aus.

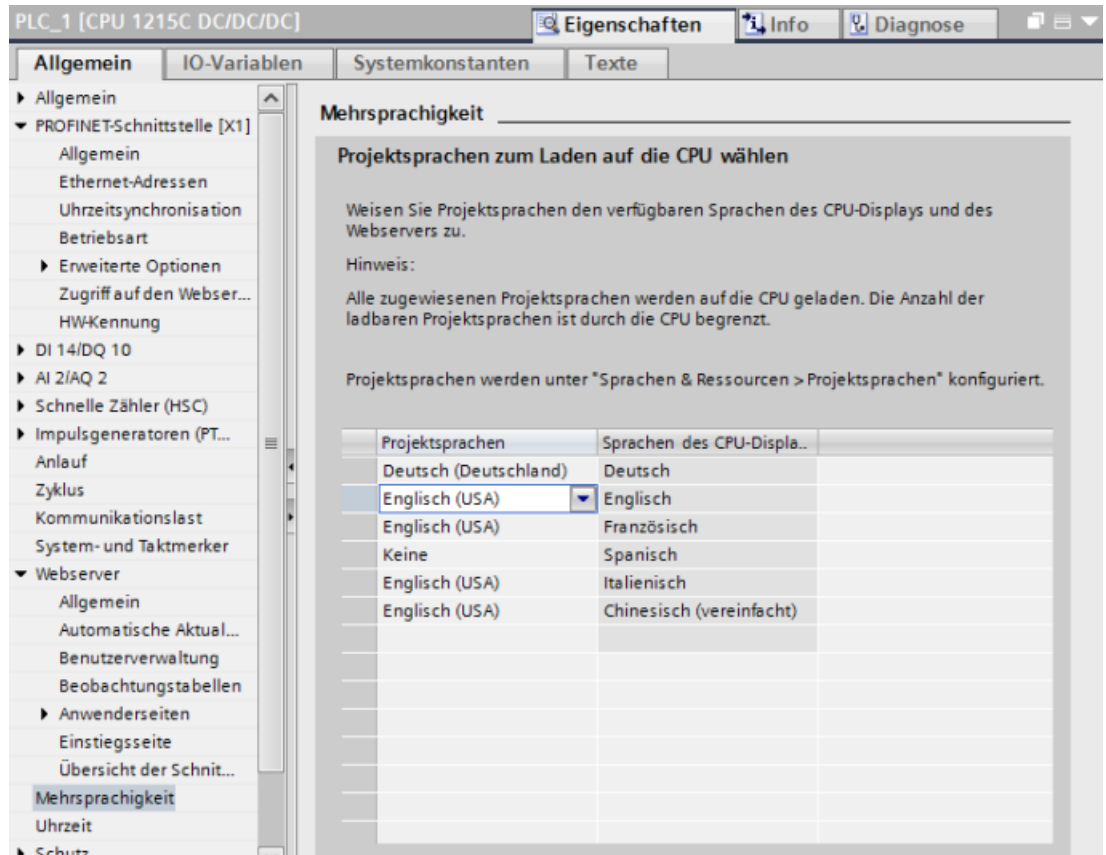
Anschließend können Sie über den Menübefehl **Werkzeuge > Projekttexte** Benutzertexte konfigurieren, z. B. Netzwerkkommentare in jeder Projektsprache. Wenn Sie die Oberflächensprache für das TIA Portal ändern, werden Netzwerkkommentare, Bausteinkommentare und andere mehrsprachige Projekttexte in der entsprechenden Projektsprache angezeigt. Die Oberflächensprache für das TIA Portal stellen Sie über den Menübefehl **Optionen > Einstellungen** auf die jeweilige Projektsprache ein.

Projektsprachen und Projekttexte lassen sich auch über den Knoten **Sprachen & Ressourcen** der Projektnavigation konfigurieren.

Der Webserver kann eine oder zwei der Projektsprachen für STEP 7 zum Anzeigen der Diagnosepuffermeldungen verwenden.

Projektsprache und Oberflächensprache des Webservers

Der Webserver unterstützt die gleichen Oberflächensprachen wie das TIA Portal, doch sind maximal zwei Projektsprachen möglich. Sie können den Webserver so konfigurieren, dass je nach Oberflächensprache des Webservers eine von zwei Projektsprachen für Diagnosepuffermeldungen verwendet wird. Sie konfigurieren diese Einstellungen über die Eigenschaften für "Mehrsprachiger Support" in der Gerätekonfiguration der CPU (Netzwerkcommentare, Bausteinkommentare und andere mehrsprachige Texte sind vom Webserver aus nicht sichtbar).



In den Eigenschaften für Mehrsprachiger Support sind die Oberflächensprachen auf der rechten Seite nicht editierbar. Es handelt sich um vordefinierte Sprachen, die sowohl für das TIA Portal als auch für die Benutzeroberfläche des Webservers verfügbar sind. Die Einstellung "Projektsprache zuweisen" ist konfigurierbar, wählen Sie eine von zwei Ihrer konfigurierten Projektsprachen oder "Keine" aus. Weil die S7-1200 CPU nur zwei Projektsprachen unterstützt, kann die Projektsprache nicht die gleiche Sprache wie die Oberflächensprache sein. Dies gilt für alle unterstützten Oberflächensprachen.

In der unten dargestellten deutschen Konfiguration Webserver zeigt Diagnosepuffereinträge an (Seite 870) (Benutzeroberfläche des Webservers in Deutsch) werden keine Texte für Diagnosepufferereignisse angezeigt, wenn die Benutzeroberfläche des Webservers auf Spanisch eingestellt ist. Diagnosepuffereinträge für alle anderen Sprachen werden in Englisch angezeigt.

6.8 Schutz & Security

6.8.1 PLC-Security-Einstellungen im Security-Assistenten festlegen

Der Security-Assistent im TIA Portal V17 und höher bietet Ihnen eine zentrale Stelle zum Konfigurieren der PLC-Security-Einstellungen. Wenn Sie eine S7-1200 CPU V4.5 oder höher (Seite 136) in Ihr Projekt einfügen, ruft das TIA Portal den Security-Assistenten auf.

Der Security-Assistent besteht aus vier Teilen:

- Schutz vertraulicher PLC-Daten
- Betriebsart für PG/PC- und HMI-Kommunikation
- PLC-Zugriffsschutz
- Übersicht

STEP 7 speichert die Einstellungen, die Sie im Assistenten vornehmen, im Projekt, wenn Sie im Assistenten auf "Fertigstellen" klicken. Wenn Sie auf "Abbrechen" klicken, speichert STEP 7 Ihre Änderungen nicht. Die Änderungen, die Sie im Assistenten vornehmen, wirken sich nur auf das STEP 7-Projekt aus.

Schutz vertraulicher PLC-Daten

Mit der Funktion "Schutz vertraulicher PLC-Konfigurationsdaten" können Sie jede CPU in Ihrem Projekt einzeln schützen. Im Security-Assistenten können Sie diesen Schutz aktivieren und ein Passwort für den Schutz der vertraulichen PLC-Konfigurationsdaten festlegen.

- Wenn das Gerät dieses Passwort nicht hat, werden Sie vom TIA Portal beim ersten Laden ins Zielsystem aufgefordert, das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten einzugeben.
- Wenn das Gerät dieses Passwort bereits hat, müssen das Passwort im STEP 7-Projekt und im Gerät übereinstimmen. Stimmen die Passwörter nicht überein, können Sie das Projekt nicht in die CPU laden. Sie müssen das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten löschen oder dafür das gleiche Passwort wie das im Gerät festlegen.

Sie können den Schutz vertraulicher PLC-Konfigurationsdaten auch in der Gerätekonfiguration der CPU (Seite 161) konfigurieren.

Betriebsart für PG/PC- und HMI-Kommunikation

Die Betriebsart für PG/PC- und HMI-Kommunikation ermöglicht Ihnen die Verwendung eines PLC-Kommunikationszertifikats (Seite 591) zum Schutz der Kommunikation zwischen Ihrer CPU und anderen Geräten:

- Programmiergeräte (PGs) wie TIA Portal und das SIMATIC Automation Tool
- HMI-Geräte

Wählen Sie im Assistenten "Nur Secure PG/PC- und HMI-Kommunikation zulassen" aus, um nur sichere Kommunikation zuzulassen.

Wenn Sie mit Geräten kommunizieren müssen, die Secure Communication nicht unterstützen, wählen Sie "Nur Secure PG/PC- und HMI-Kommunikation" ab. Bei dieser Auswahl kann Ihre PLC

entweder über Secure Communication oder über Legacy-Kommunikation (Seite 591) kommunizieren.

Sie können die Betriebsart für PG/PC- und HMI-Kommunikation (Seite 166) auch in den Verbindungsmechanismen in der Gerätekonfiguration der CPU konfigurieren.

PLC-Zugriffsschutz

Mit dem Security-Assistenten können Sie auch die Passwörter für die Schutzstufe (Seite 163) der CPU festlegen. Dies ist die gleiche Schutzstufenkonfiguration wie in der Gerätekonfiguration. Der Security-Assistent bietet vereinfachten Zugriff darauf. TIA Portal V17 und die S7-1200 CPUs V4.5 haben eine verbesserte Verschlüsselung dieser Passwörter.

Übersicht

Die Übersicht im Security-Assistenten zeigt Ihre Einstellungen für die folgenden Bereiche an:

- Schutz vertraulicher PLC-Daten (Seite 161)
- Betriebsart für PG/PC-Kommunikation und HMI-Kommunikation (Seite 166)
- PLC-Zugriffsschutz (Seite 163)

Überprüfen Sie Ihre Einstellungen und verwenden Sie gegebenenfalls die Schaltfläche "Zurück", um Änderungen vorzunehmen. Wenn Sie mit Ihren Einstellungen zufrieden sind, klicken Sie auf "Fertig stellen". STEP 7 speichert Ihre Einstellungen im Projekt.

Aufrufen des Security-Assistenten aus der Gerätekonfiguration der CPU

Sie können den Security-Assistenten für eine CPU ab V4.5 auch über die Gerätekonfiguration der CPU im Abschnitt "Schutz & Security" aufrufen.

6.8.2 Schutz vertraulicher PLC-Konfigurationsdaten

Mit der Funktion "Schutz vertraulicher PLC-Konfigurationsdaten" können Sie die Konfiguration jeder CPU in Ihrem Projekt schützen. Im Abschnitt "Schutz & Security" der Gerätekonfiguration können Sie diesen Schutz aktivieren und ein Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten festlegen.

Beachten Sie Folgendes, wenn Sie den Schutz vertraulicher PLC-Konfigurationsdaten festlegen:

- Wenn das Gerät dieses Passwort nicht hat, werden Sie vom TIA Portal beim ersten Laden ins Zielsystem aufgefordert, das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten einzugeben.
- Wenn das Gerät dieses Passwort bereits hat, müssen das Passwort im STEP 7-Projekt und im Gerät übereinstimmen. Stimmen die Passwörter nicht überein, können Sie das Projekt nicht in die CPU laden. Sie müssen das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten löschen oder dafür das gleiche Passwort wie das im Gerät festlegen. Sie können das Passwort im Gerät über "Online & Diagnose (Seite 1213)" festlegen oder löschen.

Security-Assistent

Sie können diese Funktion auch im Security-Assistenten (Seite 160) aktivieren und das Passwort festlegen. Der Security-Assistent wird gestartet, wenn Sie eine CPU V4.5 oder höher erstmalig einfügen. Sie können den Security-Assistenten auch über den Abschnitt "Schutz & Security" der Gerätekonfiguration aufrufen.

Vorteil des Schutzes vertraulicher PLC-Konfigurationsdaten

Die CPUs ab V4.5 zusammen mit TIA Portal V17 bieten Schutz für jede einzelne CPU. Der Schutz vertraulicher PLC-Konfigurationsdaten bietet eine höhere Sicherheit für den Projektspeicher jeder PLC.

Funktionsweise des Schutzes

Der Schutz funktioniert ähnlich wie ein Schloss mit Schlüssel. Sie aktivieren den Schutz vertraulicher PLC-Konfigurationsdaten im TIA Portal und legen ein Passwort fest. Durch Laden des Projekts ins Zielsystem wird das Passwort für den "Schutz vertraulicher PLC-Konfigurationsdaten" in der CPU festgelegt. Die CPU muss das Passwort entweder durch das Laden des Projekts ins Zielsystem oder über eine Memory Card (Seite 129) erhalten, um Projektdateien zu lesen. Die Projektdateien enthalten Ihre vertraulichen Konfigurationsdaten.



Verschlüsselung:

Sie aktivieren den Schutz vertraulicher PLC-Konfigurationsdaten im TIA Portal und legen ein Passwort fest. Dieses Passwort ist das Mittel zum Schutz vertraulicher PLC-Konfigurationsdaten.

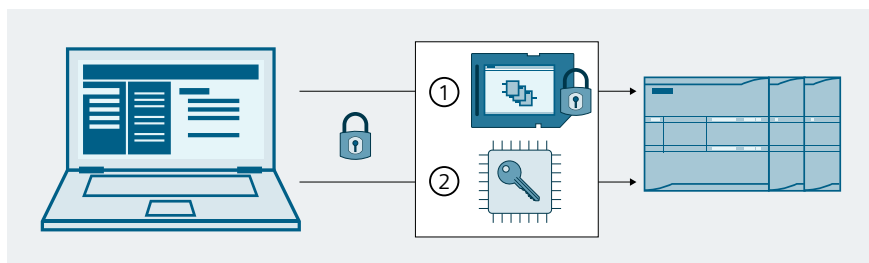


Entschlüsselung:

Sie laden das Projekt in die CPU oder alternativ von einer Memory Card (Seite 129).

Die CPU kann jetzt Projektdateien lesen (entriegeln).

Die Passwortverschlüsselung im Projekt und die nachfolgende Entschlüsselung in der CPU bieten somit eine hohe Schutzart für Ihre vertraulichen PLC-Konfigurationsdaten.



- ① Projekt mit passwortgeschützten vertraulichen Daten auf der Memory Card der CPU
- ② Schlüsselinformationen, generiert vom Passwort, im Speicherbereich der CPU, die den Zugriff auf die geschützten vertraulichen Konfigurationsdaten aktivieren.

Hinweis**Verantwortung für CPUs**

Wenn Sie den Schutz für vertrauliche PLC-Konfigurationsdaten konfiguriert und in eine CPU geladen haben, gewährleisten Sie eine sichere Entsorgung der CPU bei ihrer Außerbetriebsetzung.

Wenn Sie den Schutz für vertrauliche PLC-Konfigurationsdaten konfiguriert und in eine CPU geladen haben und später ein Projekt in die CPU laden, bei dem die Firmware-Version der CPU im STEP 7-Projekt vor V4.5 ist, enthält die CPU trotzdem das verschlüsselte Passwort. Gewährleisten Sie eine sichere Entsorgung dieser CPU bei Ihrer Außerbetriebsetzung.

Durch die sichere Entsorgung der außer Betrieb gesetzten CPUs stellen Sie sicher, dass keine Dritten Zugriff auf die geschützten vertraulichen Konfigurationsdaten erhalten.

Online-Tools

Wenn die CPU online ist, können Sie das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten auch im Tool "Online & Diagnose" festlegen, löschen oder ändern (Seite 1213).

Weitere Informationen

Weiterführende Informationen zur Funktionalität und Implementierung finden Sie im Kapitel "Secure Communication" im Informationssystem im TIA Portal.

Siehe auch

Eine CPU mit Schutz vertraulicher PLC-Konfigurationsdaten ersetzen (Seite 1455)

6.8.3 Zugriffsschutz für die CPU

Die CPU bietet verschiedene Sicherheitsstufen, um den Zugang zu bestimmten Funktionen einzuschränken. Mit dem Einrichten der Schutzstufe und des Passworts für eine CPU schränken Sie die Funktionen und Speicherbereiche ein, die ohne Eingabe eines Passworts zugänglich sind.

Jede Schutzstufe lässt auch ohne Eingabe eines Passworts den uneingeschränkten Zugriff auf bestimmte Funktionen zu. Die Standardeinstellung für die CPU ist "Kein Zugriff (kompletter Schutz)". Zur Verwendung einer Schutzstufe müssen Sie das Passwort für die entsprechende Stufe eingeben.

Wenn Sie ein Netzpasswort eingeben, wirkt sich dieses Passwort nicht auf den Passwortschutz der CPU aus. Der Passwortschutz gilt nicht für die Ausführung der Anweisungen des Anwenderprogramms einschließlich Kommunikationsfunktionen. Die Eingabe des richtigen Passworts ermöglicht den Zugriff auf alle Funktionen der jeweiligen Stufe.

Die Kommunikation zwischen CPUs (über die Kommunikationsfunktionen in den Codebausteinen) wird durch die Schutzstufe der CPU nicht eingeschränkt.

Tabelle 6-3 Schutzstufen der CPU

Schutzstufe	Zugangsbeschränkungen
Vollzugriff inkl. Fail-safe (kein Schutz)	Uneingeschränkter Zugriff auf eine F-CPU ohne Passwortschutz.
Vollzugriff (kein Schutz)	Uneingeschränkter Zugriff auf eine Standard-CPU ohne Passwortschutz.
Lesezugriff	Gestattet ohne Eingabe eines Passworts nur Lesezugriff auf Hardwarekonfiguration und Bausteine. Sie können die Hardwarekonfiguration und Bausteine ins Programmiergerät laden. Außerdem ist der HMI-Zugang und Zugriff auf Diagnosedaten möglich. Sie können Offline/Online-Vergleichsergebnisse anzeigen, den Betriebszustand (RUN/STOP) ändern und die Uhrzeit einstellen. Sie können keine Bausteine und keine Hardwarekonfiguration in die CPU laden. Firmware-Aktualisierungen sind nicht möglich.
HMI-Zugriff	Gestattet nur HMI-Zugriff.
Kein Zugriff (kompletter Schutz)	Gestattet keinen Zugriff ohne Passwort. Sie können Identifikationsdaten nur anzeigen, z.B. "Erreichbare Teilnehmer".

Beachten Sie, dass Sie auf jeder Schutzstufe eine (temporäre) Not-IP-Adresse (Seite 812) für die CPU einstellen können.

Bei Passwörtern wird Groß- und Kleinschreibung unterschieden. Um die Schutzstufe und die Passwörter zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der "Gerätekonfiguration" die CPU aus.
2. Wählen Sie im Inspektorfenster das Register "Eigenschaften".
3. Wählen Sie die Eigenschaft "Schutz & Security" aus, um die Zugriffsstufe auszuwählen und Passwörter einzugeben.

Zugriffsstufe für die PLC auswählen.

	Zugriffsstufe	Zugriff				Zugriffserla..
		HMI	Lesen	Schreiben	Fehlersi...	Passwort
<input type="radio"/>	Vollzugriff inkl. Fail-safe (kein Schutz)	✓	✓	✓	✓	*****
<input type="radio"/>	Vollzugriff (kein Schutz)	✓	✓	✓		*****
<input type="radio"/>	Lesezugriff	✓	✓			*****
<input checked="" type="radio"/>	HMI-Zugriff	✓				
<input type="radio"/>	Kein Zugriff (kompletter Schutz)					

V4.5 der S7-1200 CPU unterstützt eine verbesserte Speicherung von Passwörtern für die Zugriffsstufen. Wenn Sie eine S7-1200 CPU in V4.5 ändern, aktualisieren Sie mit der Schaltfläche "Passwortverschlüsselung aktualisieren" das Speicherformat bestehender Passwörter für die Zugriffsstufe.

Wenn Sie diese Konfiguration in die CPU laden, hat der Anwender HMI-Zugriff und kann ohne Passwort auf die HMI-Funktionen zugreifen. Um Daten zu lesen oder Offline-/Online-Codebausteine zu vergleichen, muss der Anwender das für den "Lesezugriff" konfigurierte Passwort oder das Passwort für "Vollzugriff (kein Schutz)" eingeben. Um Daten zu schreiben, muss der Anwender das für den "Vollzugriff (kein Schutz)" konfigurierte Passwort eingeben. Für

den vollständigen Zugriff auf fehlersichere CPUs muss der Anwender das konfigurierte Passwort für "Vollzugriff inkl. Fail-safe (kein Schutz)" eingeben.

**WARNUNG****Nicht berechtigter Zugriff auf eine geschützte CPU**

Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Schützen Sie die CPU-Zugriffsstufen und Webserver-Benutzer-IDs (Seite 852) durch starke Passwörter.
- Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden:
 - Großbuchstaben
 - Kleinbuchstaben
 - Ziffern
 - Sonderzeichen
- Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234.
- Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d
- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht.
- Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil die Benutzer von Webseiten für PLC-Variablen ungültige Werte festlegen können.

6.8.4 Verbindungsmechanismen konfigurieren

6.8.4.1 Zugriffsmechanismus für entfernte Partner festlegen

Auch für den Zugriff auf entfernte Verbindungspartner über die Anweisungen PUT/GET benötigt der Anwender die entsprechende Berechtigung.

Standardmäßig ist die Option "Zugriff über PUT/GET-Kommunikation erlauben" nicht aktiviert. In diesem Fall ist der Lese- und Schreibzugriff auf CPU-Daten nur mit

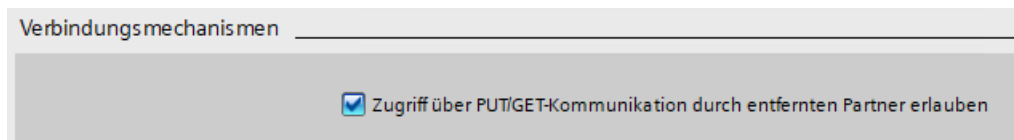
Kommunikationsverbindungen möglich, bei denen sowohl die lokale CPU als auch der Kommunikationspartner konfiguriert oder programmiert werden müssen. Beispielsweise ist der Zugriff über die Anweisungen BSEND/BRCV möglich.

Verbindungen, bei denen die lokale CPU lediglich ein Server ist (was bedeutet, dass in der lokalen CPU keine Konfiguration/Programmierung der Kommunikation mit dem Kommunikationspartner vorhanden ist) sind deshalb während des Betriebs der CPU nicht möglich. Das gilt zum Beispiel für:

- Zugriff mittels PUT/GET, FETCH/WRITE oder FTP über Kommunikationsmodule
- PUT/GET-Zugriff von anderen S7-CPU's
- HMI-Zugriff über PUT/GET-Kommunikation

Wenn Sie den clientseitigen Zugriff auf CPU-Daten erlauben möchten und somit die Kommunikationsdienste der CPU nicht einschränken möchten, gehen Sie wie folgt vor:

1. Konfigurieren Sie für den Zugriffsschutz eine beliebige Schutzstufe außer "Kein Zugriff (kompletter Schutz)".
2. Aktivieren Sie das Kontrollkästchen "Zugriff über PUT/GET-Kommunikation erlauben".



Wenn Sie diese Konfiguration in die CPU laden, erlaubt die CPU die PUT/GET-Kommunikation von entfernten Partnern.

6.8.4.2 Sichere PG/PC-Kommunikation und HMI-Kommunikation aktivieren und Zertifikate erstellen

Verwenden Sie die "Verbindungsmechanismen" in der Gerätekonfiguration Ihrer CPU, um zu konfigurieren, ob die CPU nur Secure Communication akzeptiert oder sowohl über Secure Communication als auch über Legacy-Kommunikation kommunizieren kann. Secure Communication nutzt X.509-Zertifikate über TLS (Transport Layer Security) 1.3. Die CPU verwendet diese Zertifikate, um Secure Communication zwischen der CPU und den Clients zu bieten. Client können sein:

- TIA Portal
- SIMATIC Automation Tool
- HMI-Geräte

Wählen Sie "Nur Secure PG/PC- und HMI-Kommunikation zulassen" aus, um Legacy-Kommunikation für PG/PC und HMI zu deaktivieren.

Sie können auch eigene Zertifikate erstellen. Klicken Sie neben dem PLC-Kommunikationszertifikat auf "...", um ein neues Zertifikat für die CPU hinzuzufügen oder ein vorhandenes Zertifikat auszuwählen. Unter "Zertifikate erstellen/verlängern" im Informationssystem im TIA Portal finden Sie weitere Informationen zu Parametern für die Konfiguration von Zertifikaten.

Die Konfigurationsabschnitte unter "Schutz & Security" der Gerätekonfiguration führen Sie auf dem Bildschirm durch Ihre Sicherheitsoptionen. In diesen Abschnitten finden Sie auch Links zu

Themen im Informationssystem im TIA Portal für jede Konfigurationssaufgabe und die zugehörigen Sicherheitskonzepte.

Legacy-Kommunikation

Wenn Sie mit einem Gerät kommunizieren müssen, das Secure Communication nicht unterstützt, wählen Sie "Nur Secure PG/PC- und HMI-Kommunikation" ab. Bei dieser Auswahl kann Ihre PLC entweder über Secure Communication oder über Legacy-Kommunikation (Seite 591) kommunizieren.

Ab TIA Portal V17 wird als Standardwert die höchste Stufe von Secure Communication genutzt. Aus Inbetriebnahmegründen können Sie jedoch das TIA Portal zwingen, Legacy-PG/PC-Kommunikation zu verwenden, indem Sie im Online-Menü "Nur Legacy-PG/PC-Kommunikation verwenden" auswählen.

Security-Assistent

Sie können die Funktion "Secure PG/PC- und HMI-Kommunikation" für CPUs ab V4.5 auch im Security-Assistenten (Seite 160) konfigurieren.

6.8.5 Externer Ladespeicher

Sie können das Kopieren aus dem internen Ladespeicher in einen externen Ladespeicher (SIMATIC Memory Card) verhindern. Um das Kopieren des internen Ladespeichers in den externen Ladespeicher zu verhindern, gehen Sie wie folgt vor:

1. Wählen Sie in der Gerätekonfiguration der CPU in STEP 7 unter den Eigenschaften "Allgemein" die Option "Schutz".
2. Wählen Sie im Bereich "Externer Ladespeicher" die Option "Kopieren vom internen Ladespeicher in den externen Ladespeicher deaktivieren".

Unter Memory Card in die CPU stecken (Seite 118) finden Sie eine Beschreibung dazu, wie sich diese Eigenschaft auf das Stecken einer Memory Card in die CPU auswirkt.

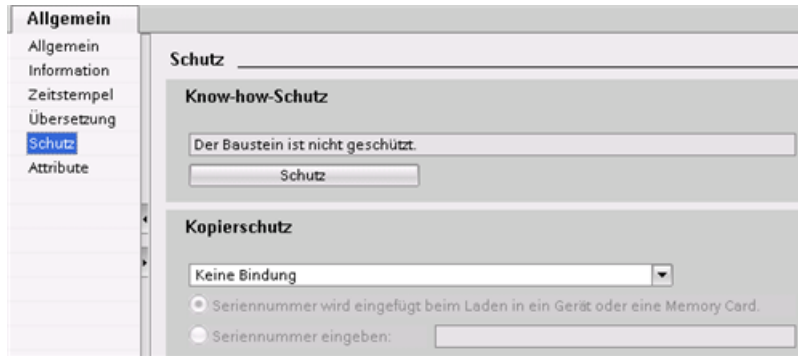
6.8.6 Knowhow-Schutz

Mit dem Knowhow-Schutz können Sie einen oder mehrere Codebausteine (OB, FB, FC oder DB) in Ihrem Programm vor unbefugtem Zugriff schützen. Sie können ein Passwort eingeben, um den Zugriff auf einen Codebaustein einzuschränken. Der Passwortschutz verhindert das unbefugte Lesen oder Ändern des Codebausteins. Ohne Passwort können nur die folgenden Informationen zum Codebaustein gelesen werden:

- Bausteintitel, Kommentar und Bausteineigenschaften
- Übertragungsparameter (IN, OUT, IN_OUT, Rückgabe)
- Aufrufstruktur des Programms
- Globale Variablen in den Querverweisen (ohne Information über die Verwendung), lokale Variablen sind jedoch verborgen

Wenn Sie einen Baustein für den Knowhow-Schutz konfigurieren, so ist der Code in diesem Baustein erst nach Eingabe des Passworts zugänglich.

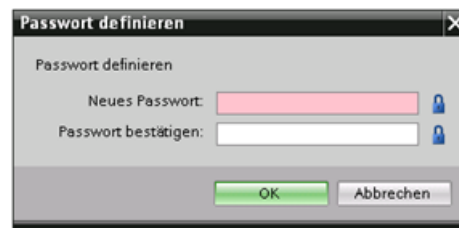
Den Knowhow-Schutz eines Codebausteins konfigurieren Sie in der Taskcard "Eigenschaften" des jeweiligen Codebausteins. Nach dem Öffnen des Codebausteins wählen Sie unter "Eigenschaften" die Option "Schutz".



1. In den Eigenschaften des Codebausteins klicken Sie auf die Schaltfläche "Eigenschaften", um den Dialog "Knowhow-Schutz" anzuzeigen.
2. Klicken Sie auf die Schaltfläche "Definieren", um das Passwort einzugeben.



Nachdem Sie das Passwort eingegeben und bestätigt haben, klicken Sie auf "OK".



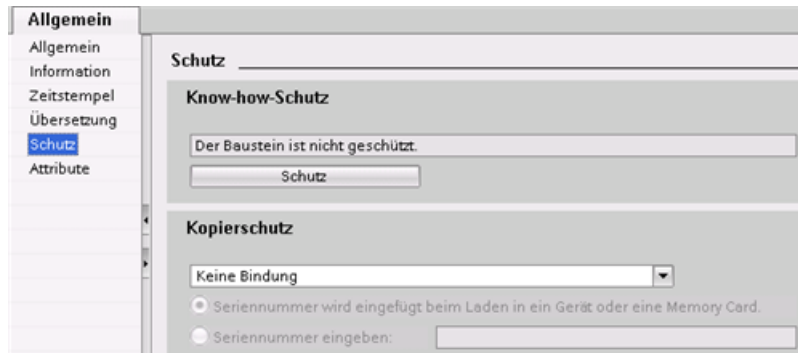
6.8.7 Kopierschutz

Eine weitere Sicherheitsfunktion ermöglicht Ihnen, Programmbausteine mit einer bestimmten Memory Card oder CPU zu verknüpfen. Diese Funktion ist vor allem zum Schutz geistigen Eigentums nützlich. Wird ein Programmbaustein mit einem bestimmten Gerät verknüpft, so ist die Verwendung des Programms oder Codebausteins nur in Verbindung mit einer bestimmten Memory Card oder CPU möglich. Diese Funktion ermöglicht Ihnen, einen Programm- oder Codebaustein elektronisch (z. B. über das Internet oder per E-Mail) oder durch Versenden einer Memory Card zu verteilen. Kopierschutz ist verfügbar für OBs (Seite 178), FBs (Seite 181) und FCs (Seite 180). Die S7-1200 CPU unterstützt drei Arten des Bausteinschutzes:

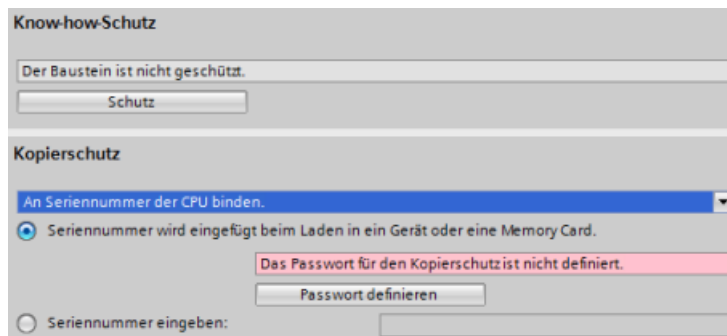
- Verknüpfung mit der Seriennummer einer CPU
- Verknüpfung mit der Seriennummer einer Memory Card
- Dynamische Verknüpfung mit einem obligatorischen Passwort

Um einen Baustein mit einer bestimmten CPU oder Memory Card zu verknüpfen, öffnen Sie die Taskcard "Eigenschaften" des jeweiligen Codebausteins.

1. Nach dem Öffnen des Codebausteins wählen Sie "Schutz".



2. Wählen Sie in der Klappliste "Kopierschutz" die Art des Kopierschutzes aus, die verwendet werden soll.



3. Für die Verknüpfung mit der Seriennummer einer CPU oder Memory Card können Sie wählen, ob die Seriennummer beim Laden eingefügt oder die Seriennummer von der Memory Card oder CPU eingegeben werden soll.

Hinweis

Bei der Seriennummer ist die Groß- und Kleinschreibung zu beachten.

Für die dynamische Verknüpfung mit einem obligatorischen Passwort ist das Passwort festzulegen, mit dem der Baustein geladen oder kopiert werden kann. Wird dann ein Baustein mit dynamischer Verknüpfung geladen (Seite 199), muss dieses Passwort eingegeben werden, um das Laden zu ermöglichen. Zu beachten ist, dass die Passwörter für den Kopierschutz und für den Knowhow-Schutz (Seite 167) verschieden sind.

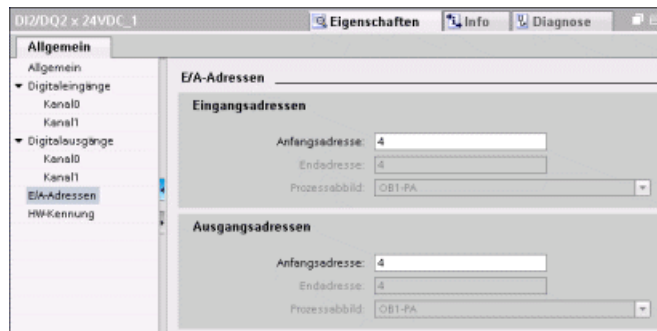
6.9 Modulparameter konfigurieren

Um die Betriebsparameter der Module zu konfigurieren, wählen Sie das Modul in der Gerätesicht aus und öffnen im Inspektorfenster das Register "Eigenschaften", um die Parameter für das Modul einzurichten.

Signalmodul (SM) oder Signalboard (SB) konfigurieren

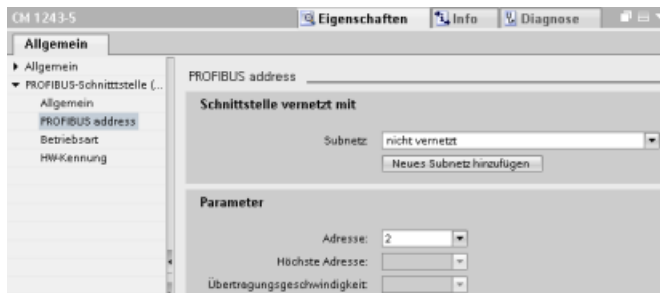
In der Gerätekonfiguration für Signalmodule und Signalboards können Sie Folgendes konfigurieren:

- Digitale E/A: Sie können Eingänge für die Erkennung steigender oder fallender Flanken (jeweils mit Zuweisung eines Ereignisses und eines Prozessalarms) oder für den "Impulsabgriff" (Eingang bleibt nach einem Impuls eingeschaltet) durch die nächste Aktualisierung des Prozessabbilds der Eingänge konfigurieren. Die Ausgänge können eingefroren werden oder es können Ersatzwerte aufgeschaltet werden.
- Analoge E/A: Für einzelne Eingänge konfigurieren Sie Parameter, wie z.B. Messart (Spannung oder Strom), Bereich und Glättung sowie Freigabe der Unter- oder Überlaufdiagnose. Die Analogausgänge bieten Parameter wie Ausgangsart (Spannung oder Strom) und Diagnose, wie z. B. Kurzschluss (bei Spannungsausgängen) oder Diagnose der oberen/unteren Grenzwerte. Die Bereiche von Analogeingängen und Analogausgängen in physikalischen Einheiten konfigurieren Sie nicht im Dialog "Eigenschaften". Dies nehmen Sie in Ihrer Programmlogik vor (siehe "Verarbeitung von Analogwerten (Seite 104)").
- E/A-Adressen: Sie konfigurieren die Startadresse für die Ein- und Ausgänge des Moduls. Sie können die Eingänge und Ausgänge auch einem Teilprozessabbild (TPA0, TPA1, TPA2, TPA3, TPA4), der automatischen Aktualisierung oder der Verwendung keines Teilprozessabbilds zuweisen. Eine Erläuterung von Prozessabbildern und Teilprozessabbildern finden Sie unter "Ausführung des Anwenderprogramms" (Seite 67).



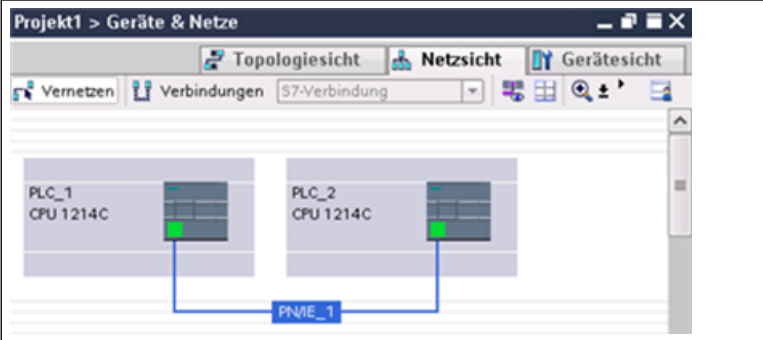
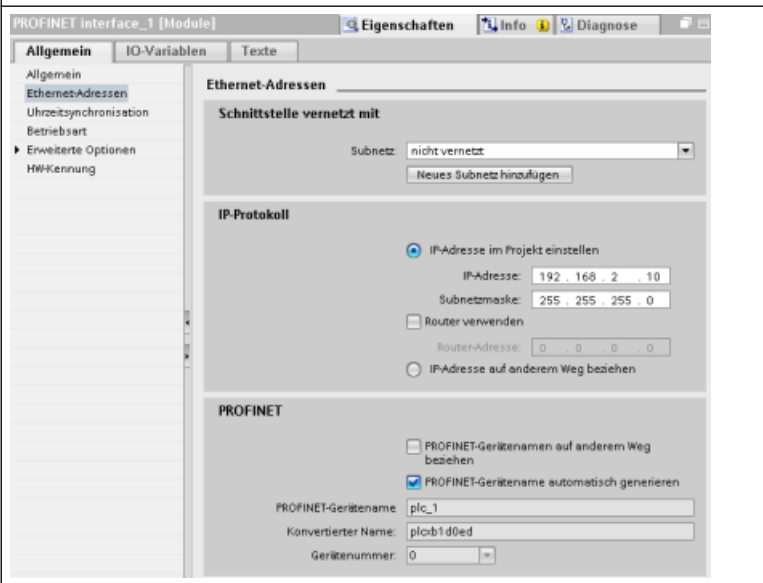
Kommunikationsschnittstelle (CM, CP oder CB) konfigurieren

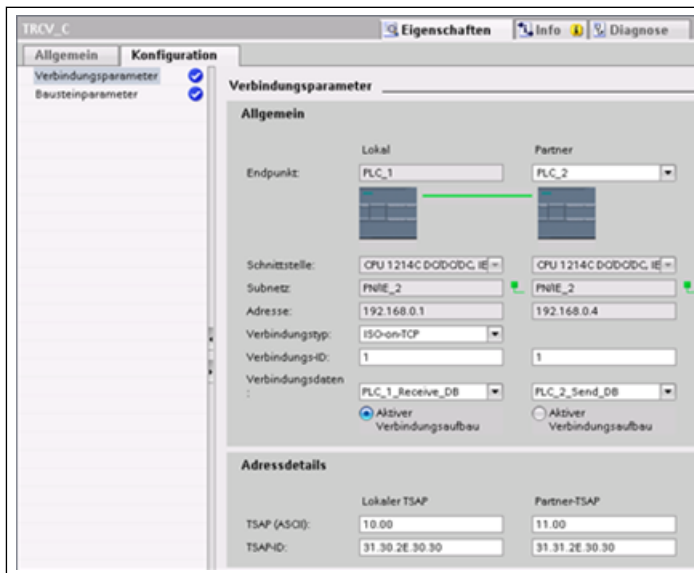
Sie konfigurieren die Parameter für das Netzwerk je nach Typ der Kommunikationsschnittstelle.



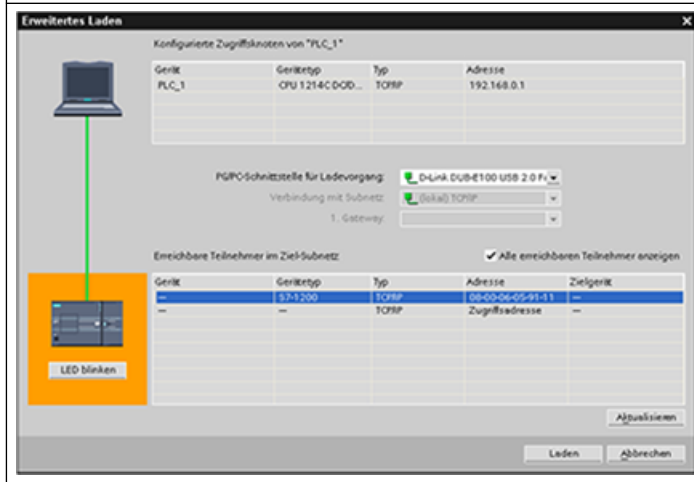
6.10 CPU für die Kommunikation konfigurieren

Die S7-1200 erfüllt Ihre Kommunikations- und Vernetzungsanforderungen durch Unterstützung sowohl einfacher als auch komplexer Netze. Die S7-1200 bietet außerdem Werkzeuge für die Kommunikation mit anderen Geräten, z.B. mit Druckern und Waagen, die über eigene Kommunikationsprotokolle verfügen.

	<p>In der Netzansicht der Gerätekonfiguration können Sie die Netzwerkverbindungen zwischen den Geräten in Ihrem Projekt herstellen. Nach dem Herstellen der Netzwerkverbindung können Sie im Register "Eigenschaften" des Inspektorfensters die Netzwerkparameter konfigurieren.</p> <p>Weitere Informationen finden Sie unter "Erstellen einer Netzwerkverbindung" (Seite 600).</p>
	<p>Wählen Sie im Dialog "Eigenschaften" den Eintrag "Ethernet-Adressen". STEP 7 zeigt den Dialog für die Konfiguration der Ethernet-Adresse an, in dem Sie dem Softwareprojekt die IP-Adresse der CPU zuweisen, in die das Projekt geladen wird.</p> <p>Hinweis: Die S7-1200 CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher manuell eine IP-Adresse zuweisen.</p> <p>Weitere Informationen finden Sie unter "Internet Protocol (IP)-Adressen zuweisen" (Seite 603).</p>



Bei den TCP-, ISO-on-TCP- und UDP-Ethernet-Protokollen konfigurieren Sie die Verbindungen der lokalen und der Partner-CPU in den "Eigenschaften" der Anweisung (TSEND_C, TRCV_C oder TCON). Die Abbildung zeigt die "Verbindungseigenschaften" im Register "Verbindung" einer ISO-on-TCP-Verbindung. Weitere Informationen finden Sie unter "Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren" (Seite 600).



Nach Abschluss der Konfiguration laden Sie das Projekt in die CPU. Alle IP-Adressen werden beim Laden des Projekts konfiguriert. Weitere Informationen finden Sie unter "Testen des PROFINET-Netzwerks" (Seite 612).

Hinweis

Für die Verbindung zu Ihrer CPU müssen Ihre Netzwerkkarte (NIC) und die CPU der gleichen Netzwerkkategorie und dem gleichen Subnetz angehören. Sie können entweder Ihre Netzwerkkarte nach der voreingestellten IP-Adresse der CPU einrichten oder die IP-Adresse der CPU passend zur Netzwerkkategorie und zum Subnetz der Netzwerkschnittstellenkarte ändern.

Weitere Informationen finden Sie unter "Internet Protocol (IP)-Adressen zuweisen" (Seite 603).

6.11 Uhrzeitsynchronisation

Ziel der Uhrzeitsynchronisation der Echtzeituhr ist es, eine Masteruhr zu haben, mit der alle anderen lokalen Uhren synchronisiert werden. Die Uhren werden durch die Masteruhr anfänglich synchronisiert und anschließend regelmäßig neu synchronisiert, um sich im Laufe der Zeit summierende Abweichungen zu verhindern.

Bei der S7-1200 und den lokalen Basiskomponenten haben nur die CPU und einige der CP-Module eine Echtzeituhr, die gegebenenfalls synchronisiert werden muss. Sie können die Echtzeituhr der CPU mit einer externen Masteruhr synchronisieren. Die externe Masteruhr liefert die Uhrzeit möglicherweise über einen NTP-Server oder ein CP-Modul auf dem lokalen Baugruppenträger der S7-1200, der mit einem SCADA-System verbunden ist, in dem sich eine Masteruhr befindet.

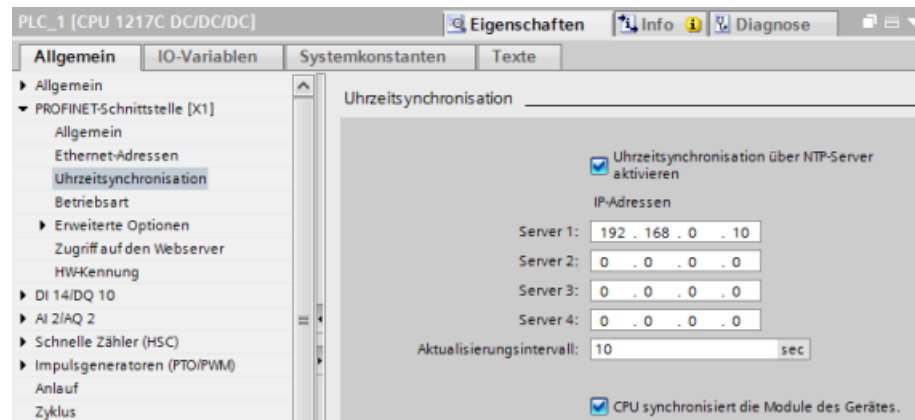
Weitere Informationen zu allen S7-1200 CPs (<https://support.industry.siemens.com/cs/de/de/ps>), die die Zeitsynchronisation unterstützen, finden Sie auf der Website Siemens Industry Online-Support, Produkt-Support.

Einstellen der Echtzeituhr

Sie können die Echtzeituhr in der S7-1200 CPU auf drei verschiedene Arten einstellen:

- Über den NTP-Server (Seite 615)
- Über STEP 7
- Über das Anwenderprogramm
- Über ein HMI-Panel

Sie konfigurieren die Uhrzeitsynchronisation der CP-Module anhand der Uhr der CPU, indem Sie das Kontrollkästchen "CPU synchronisiert die Module des Geräts" aktivieren, wie in der folgenden Abbildung dargestellt:



Standardmäßig ist weder die Uhrzeitsynchronisation über den NTP-Server noch die Uhrzeitsynchronisation der CP-Uhren mit der Uhr der CPU aktiviert.

Sie konfigurieren die Uhrzeitsynchronisation der CPU-Uhr und die Uhrzeitsynchronisation der CP-Uhren unabhängig voneinander. Von daher können Sie die Uhrzeitsynchronisation der CP-Uhren über die CPU aktivieren, wenn die Uhr der CPU über eine der oben beschriebenen Methoden eingestellt wird.

Wenn Sie den NTP-Server verwenden, können Sie das Aktualisierungsintervall auswählen. Für das Aktualisierungsintervall des NTP-Servers sind standardmäßig 10 Sekunden eingestellt.

Wenn Sie die Uhrzeitsynchronisation in einem Modul aktivieren, werden Sie von STEP 7 aufgefordert, das Kontrollkästchen "CPU synchronisiert die Module des Geräts" im Dialog "Uhrzeitsynchronisation" zu aktivieren, sofern noch nicht geschehen. STEP 7 zeigt auch eine Warnung an, wenn Sie mehrere Masteruhren als Quelle für die Uhrzeitsynchronisation

6.11 Uhrzeitsynchronisation

konfiguriert haben (wenn Sie beispielsweise die Uhrzeitsynchronisation auf mehreren CP-Modulen oder auf der CPU und einem Modul aktiviert haben).

Hinweis

Die Aktivierung der Uhrzeitsynchronisation auf einem CP-Modul bewirkt, dass dieses CP-Modul die Uhr der CPU stellt.

Wenn Sie im Dialog "Uhrzeitsynchronisation" der CPU das Kontrollkästchen "CPU synchronisiert die Module des Geräts" aktiviert haben, ist die CPU die Masteruhr. Die CP-Module synchronisieren dann die Uhr der CPU.

Hinweis

Konfigurieren Sie für die CPU nur eine Zeitquelle. Erhält die CPU die Uhrzeitsynchronisation von mehreren Quellen (NTP-Server und CP-Modul beispielsweise), kann dies zu Konflikten bei der Uhrzeitaktualisierung führen. Die Uhrzeitsynchronisation von mehreren Quellen kann auf der Uhrzeit basierende Anweisungen und Ereignisse beeinträchtigen.

Programmierkonzepte

7.1 Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät

Bei der Entwicklung eines PLC-Systems können Sie aus einer Vielzahl von Methoden und Kriterien auswählen. Die folgenden allgemeinen Richtlinien sind auf viele Projekte anwendbar. Dabei sollten Sie sich selbstverständlich an die Verfahrensanweisungen in Ihrem Unternehmen halten und Ihre eigenen Erfahrungen berücksichtigen.

Tabelle 7-1 Richtlinien für das Entwerfen einer Automatisierungslösung mit einem PLC-Gerät

Empfohlene Schritte	Aufgaben
Gliedern Ihres Prozesses bzw. Ihrer Anlage	Unterteilen Sie Ihren Prozess bzw. Ihre Anlage in Abschnitte, die voneinander unabhängig sind. Diese Abschnitte legen die Grenzen zwischen mehreren Automatisierungssystemen fest und beeinflussen die Beschreibungen der Funktionsbereiche sowie die Zuordnung der Betriebsmittel.
Beschreiben der Funktionsbereiche	Geben Sie die Funktionsbeschreibungen für jeden Abschnitt des Prozesses bzw. der Anlage ein, wie z. B. Ein-/Ausgänge, die funktionale Beschreibung des Ablaufs, die Zustände, die erreicht werden müssen, bevor ein Aktor (z. B. ein Magnetventil, ein Motor oder ein Antrieb) reagieren darf, eine Beschreibung der Benutzeroberfläche und aller Schnittstellen zu anderen Abschnitten des Prozesses oder der Anlage.
Entwerfen der Sicherheitsstromkreise	Bestimmen Sie die Geräte, die aus Sicherheitsgründen festverdrahtete Schaltungen benötigen. Steuerungsgeräte können unsichere Betriebszustände einnehmen, woraus unerwartete Anlauferscheinungen bzw. geänderte Funktionsabläufe der Anlage resultieren können. Besteht die Gefahr, dass bei unerwartetem bzw. fehlerhaftem Betrieb der Anlage schwere Körperverletzungen oder Sachschäden auftreten, sollten Sie die Einrichtung elektromechanischer Programmeingriffe in Erwägung ziehen, die unabhängig von der Steuerung arbeiten, um unsichere Betriebszustände zu vermeiden. Zum Entwerfen von Sicherheitsstromkreisen gehen Sie folgendermaßen vor: <ul style="list-style-type: none"> • Definieren Sie falschen bzw. unerwarteten Betrieb von Aktoren, die Gefahrenpotentiale bergen. • Definieren Sie die Bedingungen, unter denen der Betrieb ungefährlich ist, und legen Sie fest, wie diese Bedingungen unabhängig von der SPS erkannt werden. • Definieren Sie, wie die SPS den Prozess beeinflusst, wenn die Spannung eingeschaltet und wieder ausgeschaltet wird und wenn Fehler erkannt werden. Verwenden Sie diese Informationen nur für den Entwurf des normalen und erwarteten unnormalen Betriebs. Verlassen Sie sich aus Sicherheitsgründen nicht auf dieses "Best-Case"-Szenario. • Entwerfen Sie Korrekturen durch Handeingriff bzw. elektromechanische Programmeingriffe, mit denen gefährliche Abläufe unabhängig von der SPS gesperrt werden. • Lassen Sie von den unabhängigen Stromkreisen Statusinformationen an die SPS übermitteln, so dass das Programm und jede Bedienerschnittstelle über die erforderlichen Informationen verfügt. • Definieren Sie weitere Sicherheitsanforderungen, damit der Prozess sicher ablaufen kann.
Systemicherheit planen	Ermitteln Sie, welche Schutzstufe (Seite 160) Sie für den Zugriff auf Ihren Prozess benötigen. Sie können CPUs und Programmbausteine mittels Passwort vor unberechtigtem Zugriff schützen.

Empfohlene Schritte	Aufgaben
Definieren der Operator-Stationen	Erstellen Sie die folgenden Pläne der OperatorStationen anhand der Anforderungen in den Beschreibungen der Funktionsbereiche: <ul style="list-style-type: none"> • Position aller OperatorStationen in Bezug zum Prozess bzw. zur Anlage • Mechanische Anordnung der Geräte der Operator-Station, z. B. Display, Schalter und Lampen. • Verdrahtungspläne mit den zugehörigen Ein und Ausgängen der SPS und Signalmodule
Zeichnen der Konfigurationspläne	Erstellen Sie die Konfigurationspläne für das Automatisierungssystem anhand der Anforderungen in den Beschreibungen der Funktionsbereiche: <ul style="list-style-type: none"> • Positionen aller SPS in Bezug zum Prozess bzw. zur Anlage • Mechanische Anordnung aller SPS und E/A-Module (einschließlich Schränke usw.) • Verdrahtungspläne für alle SPS und E/A-Module, einschließlich Geräternummern, Kommunikationsadressen und Adressen der Ein und Ausgänge
Auflisten der symbolischen Namen	Erstellen Sie eine Liste der symbolischen Namen für die absoluten Adressen. Geben Sie nicht nur die physikalischen Ein und Ausgänge, sondern auch alle anderen Elemente (wie z. B. Variablennamen) an, die Sie in Ihrem Programm verwenden.

7.2 Strukturieren Ihres Anwenderprogramms

Beim Erstellen eines Anwenderprogramms für Automatisierungslösungen fügen Sie die Anweisungen für das Programm in Codebausteine ein:

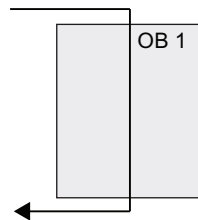
- Ein Organisationsbaustein (OB) reagiert auf ein bestimmtes Ereignis in der CPU und kann die Ausführung des Anwenderprogramms unterbrechen. Der Standardbaustein für die zyklische Ausführung des Anwenderprogramms (OB 1) stellt die Grundstruktur für Ihr Anwenderprogramm bereit. Fügen Sie weitere OBs in Ihr Programm ein, so unterbrechen diese OBs die Ausführung von OB 1. Die anderen OBs führen spezifische Funktionen aus, wie z. B. für Anlauf, Interrupt- und Fehlerbearbeitung, oder für die Ausführung eines spezifischen Programmcodes in bestimmten Zeitabständen.
- Ein Funktionsbaustein (FB) ist ein Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Der aufrufende Baustein übergibt dem FB die Parameter und definiert einen bestimmten Datenbaustein (DB), der die Daten für diesen Aufruf oder diese Instanz des FBs speichert. Wird der Instanz-DB geändert, so kann ein allgemeiner FB den Betrieb einer Gerätegruppe steuern. Beispielsweise kann ein FB mehrere Pumpen oder Ventile mit unterschiedlichen Instanz-DBs, die die speziellen Betriebsparameter jeder Pumpe bzw. jedes Ventils enthalten, steuern.
- Eine Funktion (FC) ist eine Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Einer FC ist kein Instanz-DB zugeordnet. Der aufrufende Baustein übergibt der FC die Parameter. Die Ausgangswerte der FC müssen in eine Speicheradresse oder in einen globalen DB geschrieben werden.

Art der Struktur für das Anwenderprogramm wählen

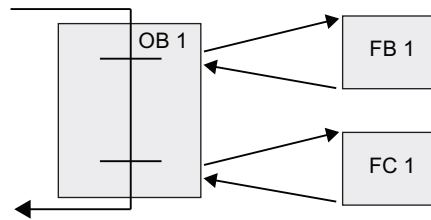
Je nach den Anforderungen Ihrer Anwendung können Sie eine lineare oder eine modulare Struktur für Ihr Anwenderprogramm wählen:

- Ein lineares Programm führt alle Anweisungen für Ihre Automatisierungsaufgaben nacheinander aus. Typischerweise werden bei einem linearen Programm alle Programmanweisungen im Zyklus-OB (OB 1) abgelegt.
- Ein modulares Programm ruft spezielle Codebausteine auf, die spezifische Aufgaben ausführen. Um eine modulare Programmstruktur aufzubauen, gliedern Sie die komplexe Automatisierungsaufgabe in kleinere Teilaufgaben, die den technologischen Funktionen des Prozesses entsprechen. Jeder Codebaustein enthält das Programmsegment für die jeweilige Teilaufgabe. Sie strukturieren Ihr Programm durch den Aufruf eines Codebausteins aus einem anderen Baustein.

Lineare Struktur:



Modulare Struktur:



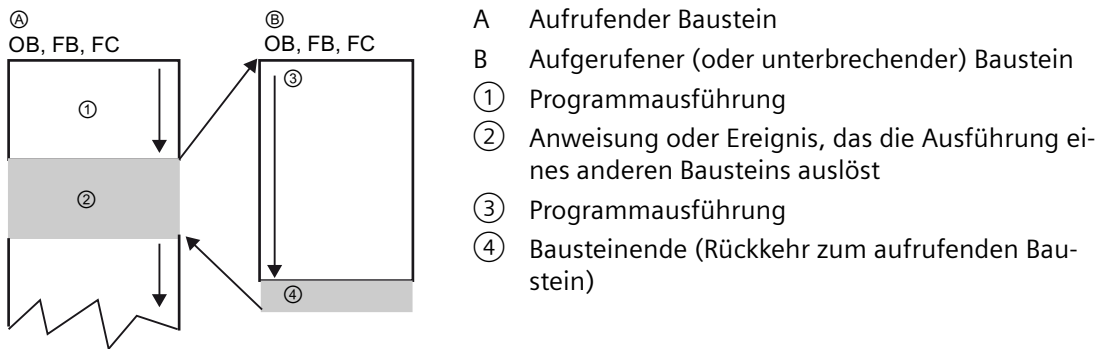
Durch das Anlegen allgemeiner Codebausteine, die im Anwenderprogramm mehrfach genutzt werden können, lassen sich Entwurf und Implementierung des Anwenderprogramms vereinfachen. Die Arbeit mit allgemeinen Codebausteinen hat mehrere Vorteile:

- Sie können wiederverwendbare Codebausteine für Standardaufgaben, wie z. B. für die Steuerung einer Pumpe oder eines Motors, erstellen. Sie können diese allgemeinen Codebausteine ferner in einer Bibliothek speichern, die für verschiedene Anwendungen oder Lösungen verwendet werden kann.
- Durch die modulare Gliederung des Anwenderprogramms in einzelne Komponenten, die sich auf funktionale Aufgaben beziehen, wird der Programmaufbau übersichtlicher und ist einfacher zu handhaben. Die modularen Komponenten ermöglichen nicht nur die Standardisierung des Programmentwurfs, sondern machen auch Anpassungen oder Änderungen des Programmcodes schneller und einfacher.
- Das Erstellen modularer Komponenten vereinfacht das Testen Ihres Programms. Wenn das komplette Programm in eine Folge modularer Programmsegmente gegliedert ist, können Sie die Funktionalität jedes Codebausteins direkt während der Entwicklung testen.
- Mit modularen Komponenten, die sich auf spezifische technologische Funktionen beziehen, wird die Inbetriebnahme der kompletten Anwendung vereinfacht und verkürzt.

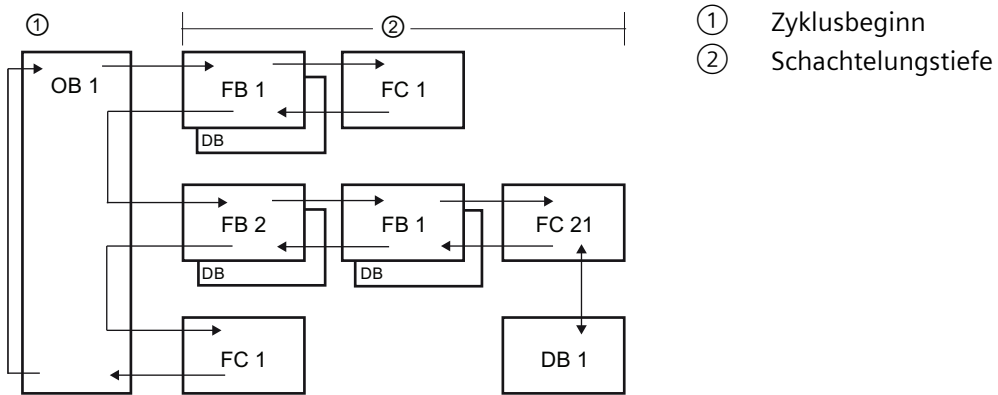
7.3 Verwendung von Bausteinen zum Strukturieren Ihres Programms

Modulare Codebausteine erstellen Sie durch den Entwurf von FBs und FCs für die Ausführung allgemeiner Aufgaben. Dann strukturieren Sie Ihr Programm, indem andere Codebausteine diese wiederverwendbaren Module aufrufen. Der aufrufende Baustein gibt gerätespezifische Parameter an den aufgerufenen Baustein weiter.

Wird ein Codebaustein von einem anderen Codebaustein aufgerufen, führt die CPU den Programmcode im aufgerufenen Baustein aus. Nachdem der aufgerufene Baustein abgearbeitet ist, setzt die CPU die Ausführung des aufrufenden Bausteins fort. Die Bearbeitung geht weiter mit der Ausführung der nächsten Anweisung nach dem Bausteinaufruf.



Die Bausteinaufrufe können verschachtelt werden, um die Struktur noch modularer zu gestalten. Im folgenden Beispiel umfasst die Schachtelungstiefe 3 Ebenen: den Programmzyklus-OB plus 3 Ebenen mit Aufrufen von Codebausteinen.



Hinweis: Die maximale Schachtelungstiefe sind sechs Ebenen. Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

7.3.1 Organisationsbaustein (OB)

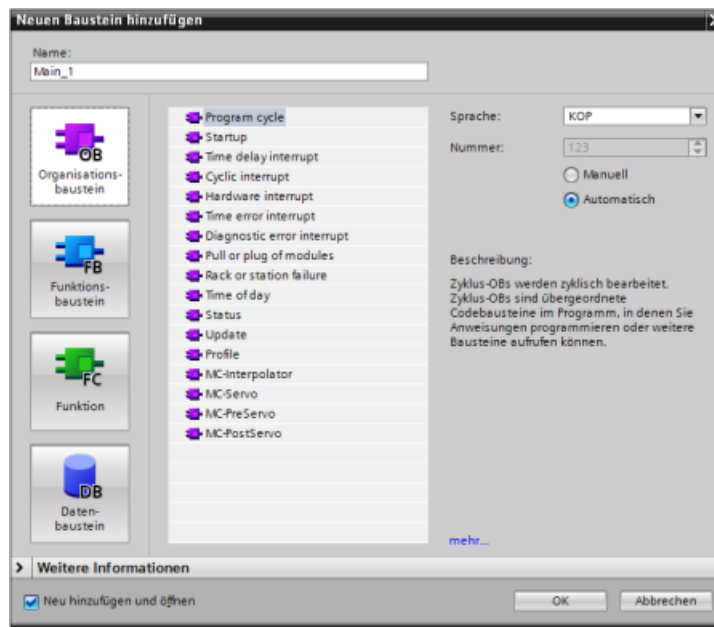
Organisationsbausteine dienen zur Strukturierung Ihres Programms. Sie bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. OBs sind ereignisgesteuert. Ein Ereignis, z. B. ein Diagnosealarm oder ein Zeitintervall, veranlasst die CPU zur Ausführung eines OB. Manche OBs haben vordefinierte Startereignisse und vordefiniertes Verhalten.

Der Programmzyklus-OB enthält das Hauptprogramm. Sie können mehrere Programmzyklus-OBs in Ihr Anwenderprogramm aufnehmen. Im Betriebszustand RUN werden die Programmzyklus-OBs mit der niedrigsten Prioritätsstufe ausgeführt und können durch alle anderen Ereignisarten unterbrochen werden. Der Anlauf-OB unterbricht den Programmzyklus-OB nicht, weil die CPU den Anlauf-OB vor dem Wechsel in RUN ausführt.

Nach Abarbeitung des Programmzyklus-OBs führt die CPU die Programmzyklus-OBs sofort erneut aus. Diese zyklische Ausführung ist die "normale" Ablaufart für speicherprogrammierbare Steuerungen. In vielen Anwendungen ist das gesamte Anwenderprogramm in einem einzigen Programmzyklus-OB enthalten.

Sie können andere OBs anlegen, um bestimmte Funktionen auszuführen, z. B. für die Alarm- und Fehlerbearbeitung oder für die Ausführung eines bestimmten Programmcodes in bestimmten Abständen. Diese OBs unterbrechen die Ausführung der Programmzyklus-OBs.

Neue OBs für Ihr Anwenderprogramm legen Sie im Dialog "Neuen Baustein hinzufügen" an.



Die Alarmbearbeitung ist immer ereignisgesteuert. Tritt ein solches Ereignis auf, so unterbricht die CPU die Ausführung des Anwenderprogramms und ruft den OB für die Behandlung dieses Ereignisses auf. Nach der Ausführung des unterbrechenden OBs setzt die CPU die Ausführung des Anwenderprogramms an der Stelle fort, an der es zuvor unterbrochen wurde.

Die CPU ermittelt die Reihenfolge für die Bearbeitung von Alarmereignissen anhand der Prioritäten. Sie können einer Prioritätsklasse mehrere Alarmereignisse zuweisen. Weitere Informationen finden Sie in den Themen zu Organisationsbausteinen (Seite 75) und zur Ausführung des Anwenderprogramms (Seite 67).

Zusätzliche OBs anlegen

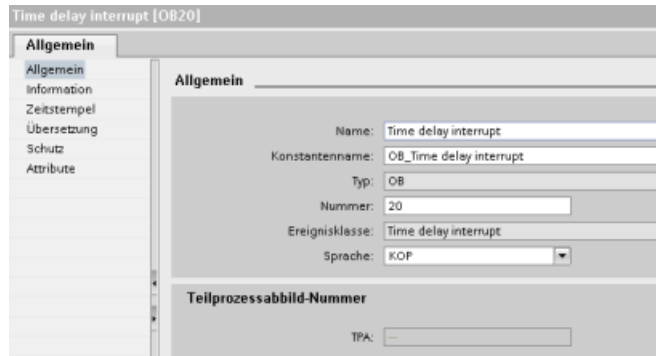
Sie können mehrere OBs für Ihr Anwenderprogramm anlegen. Dies gilt auch für die Programmzyklus- und Anlauf-OB-Ereignisse. Im Dialog "Neuen Baustein hinzufügen" können Sie einen OB anlegen und einen Namen für den neuen OB eingeben.

Wenn Sie mehrere Programmzyklus-OBs für Ihr Anwenderprogramm anlegen, führt die CPU die einzelnen Programmzyklus-OBs in numerischer Reihenfolge aus. Dabei wird mit dem Programmzyklus-OB mit der niedrigsten Nummer begonnen (z. B. OB 1). Beispiel: Nach dem

Ende des ersten Programmzyklus-OBs (z. B. OB 1) führt die CPU den Programmzyklus-OB mit der nächsthöheren Nummer aus.

Eigenschaften eines OBs konfigurieren

Sie können die Eigenschaften eines OBs, beispielsweise die OB-Nummer oder die Programmiersprache, konfigurieren.



Hinweis

Beachten Sie, dass Sie einem OB eine Teilprozessabbild-Nummer wie TPA0, TPA1, TPA2, TPA3 oder TPA4 zuweisen können. Wenn Sie für die Teilprozessabbild-Nummer eine Nummer eingeben, erstellt die CPU das Teilprozessabbild. Erläuterungen zu Teilprozessabbildern finden Sie unter "Ausführung des Anwenderprogramms (Seite 67)".

7.3.2 Funktion (FC)

Eine Funktion (FC) ist ein Codebaustein, der typischerweise eine bestimmte Operation mit einer Anzahl von Eingangswerten durchführt. Die FC speichert die Ergebnisse dieser Operation an bestimmten Speicherorten. Verwenden Sie beispielsweise FCs, um Standardoperationen und mehrfach verwendbare Operationen (z. B. mathematische Berechnungen) oder technologische Funktionen (z. B. für einzelne Steuerungsvorgänge über Bitverknüpfungsoperationen) durchzuführen. Eine FC kann auch mehrmals an verschiedenen Stellen eines Programms aufgerufen werden. Diese Wiederverwendung vereinfacht die Programmierung häufig wiederkehrender Aufgaben.

Einer FC ist kein zugehöriger Instanz-Datenbaustein(DB) zugeordnet. Die FC verwendet den lokalen Datenspeicher für die temporären Daten, die für die Berechnung der Operation benötigt werden. Die temporären Daten werden nicht gespeichert. Für die dauerhafte Datenspeicherung muss der Ausgangswert einem globalen Speicherplatz, wie z. B. dem Merkerspeicher, oder einem globalen DB zugewiesen werden.

7.3.3 Funktionsbaustein (FB)

Ein Funktionsbaustein (FB) ist ein Codebaustein, der für seine Parameter und statischen Daten einen Instanz-Datenbaustein nutzt. FBs haben einen variablen Speicher, der sich in einem Datenbaustein (DB) oder einem Instanz-DB befindet. Der Instanz-DB stellt einen Speicherbaustein bereit, der dieser Instanz (oder diesem Aufruf) des FBs zugewiesen ist und die Daten nach Ablauf des FBs speichert. Sie können verschiedenen Aufrufen des FBs verschiedene Instanz-DBs zuordnen. Der Instanz-DB ermöglicht es Ihnen, einen allgemeinen FB für die Steuerung mehrerer Geräte zu verwenden. Sie können Ihr Programm strukturieren, indem ein Codebaustein einen FB und einen Instanz-DB aufruft. Die CPU führt dann den Programmcode in diesem FB aus und speichert die Bausteinparameter und die statischen Lokaldaten im Instanz-DB. Wenn die Ausführung des FBs beendet ist, setzt die CPU die Ausführung mit dem Codebaustein fort, der den FB aufgerufen hatte. Der Instanz-DB speichert die Werte für diese Instanz des FBs. Diese Werte stehen nachfolgenden Aufrufen des Funktionsbausteins entweder in demselben Zyklus oder in anderen Zyklen zur Verfügung.

Wiederverwendbare Codebausteine mit zugewiesenem Speicher

Typischerweise wird ein FB für die Steuerung des Ablaufs von Tätigkeiten oder Geräten verwendet, deren Betrieb nicht innerhalb eines Zyklus endet. Für die Speicherung der Betriebsparameter, damit diese zwischen zwei Zyklen schnell zugänglich sind, hat jeder FB in Ihrem Anwenderprogramm einen oder mehrere Instanz-DBs. Mit dem Aufruf eines FBs geben Sie gleichzeitig einen Instanz-DB an, in dem die Bausteinparameter und die statischen Lokaldaten für diesen Aufruf oder diese "Instanz" des FBs enthalten sind. Der Instanz-DB speichert diese Werte nach der Ausführung des FBs.

Wird der FB für allgemeine Steuerungsaufgaben entworfen, so kann er für mehrere Geräte verwendet werden, indem verschiedene Instanz-DBs für die verschiedenen Aufrufe des FBs ausgewählt werden.

Ein FB speichert die Eingangs-, Ausgangs- und Durchgangsparameter sowie die statischen Parameter in einem Instanz-DB.

Sie können ferner die Funktionsbausteinschnittstelle in Betriebszustand RUN ändern und laden (Seite 1237).

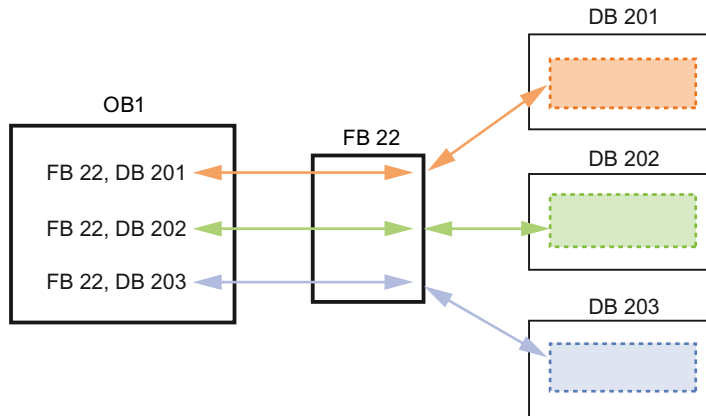
Startwert im Instanz-DB zuweisen

Der Instanz-DB speichert sowohl einen Standardwert als auch einen Startwert für jeden Parameter. Der Startwert gibt den Wert an, der verwendet werden soll, wenn der FB ausgeführt wird. Der Startwert kann dann während der Ausführung Ihres Anwenderprogramms geändert werden.

Die FB-Schnittstelle verfügt auch über eine Spalte für den Standardwert, in der Sie beim Schreiben des Programmcode einen neuen Startwert für den Parameter eingeben können. Dieser Standardwert im FB wird dann in den Startwert des zugehörigen Instanz-DBs übertragen. Wenn Sie in der FB-Schnittstelle einem Parameter keinen neuen Startwert zuweisen, wird der Standardwert aus dem Instanz-DB in den Startwert kopiert.

Einzelnen FB mit DBs verwenden

Die folgende Abbildung zeigt einen OB, der einen FB dreimal aufruft, wobei für jeden Aufruf ein anderer Datenbaustein verwendet wird. Durch diese Struktur kann ein allgemeiner FB für die Steuerung mehrerer gleichartiger Geräte wie z. B. Motoren verwendet werden, indem jedem Aufruf eines Geräts ein anderer Instanzdatenbaustein zugewiesen wird. Jeder Instanz-DB speichert die Daten (wie Drehzahl, Hochlaufzeit und Gesamtbetriebszeit) für ein einziges Gerät.



In diesem Beispiel steuert FB 22 drei verschiedene Geräte, DB 201 speichert die Betriebsdaten für das erste Gerät, DB 202 die Betriebsdaten für das zweite Gerät und DB 203 die Betriebsdaten für das dritte Gerät.

7.3.4 Datenbaustein (DB)

Sie können in Ihrem Anwenderprogramm Datenbausteine (DBs) zum Speichern der Daten für die Codebausteine anlegen. Alle Programmbausteine im Anwenderprogramm können auf die Daten in einem globalen DB zugreifen, doch ein Instanz-DB speichert Daten für einen spezifischen Funktionsbaustein (FB).

Die gespeicherten Daten in einem DB werden nach der Ausführung des zugehörigen Codebausteins nicht gelöscht. Es gibt zwei Arten von Datenbausteinen:

- In einem globalen DB werden die Daten für die Codebausteine in Ihrem Programm gespeichert. Jeder OB, FB oder FC kann auf die Daten in einem globalen DB zugreifen.
- In einem Instanz-DB werden die Daten für einen spezifischen FB gespeichert. Die Datenstruktur in einem Instanz-DB entspricht den Parametern (Input, Output und InOut) und den statischen Daten des FBs. (Der temporäre Speicher des FBs wird im Instanz-DB nicht gespeichert.)

Hinweis

Obwohl der Instanz-DB die Daten für einen bestimmten FB enthält, kann jeder Codebaustein auf die Daten eines Instanz-DBs zugreifen.

Sie können Datenbausteine auch im Betriebszustand RUN ändern und laden (Seite 1237).

Schreibgeschützte Datenbausteine

Ein DB kann so konfiguriert werden, dass er nur gelesen werden kann:

1. Klicken Sie mit der rechten Maustaste in der Projektnavigation auf den DB und wählen Sie im Kontextmenü "Eigenschaften".
2. Wählen Sie Dialog "Eigenschaften" das Element "Attribute".
3. Wählen Sie die Option "Datenbaustein im Gerät schreibgeschützt" und bestätigen Sie mit "OK".

Optimierte und Standard-Datenbausteine

Sie können den Zugriff auf den Datenbaustein auch als optimierten Zugriff konfigurieren. Wenn der Baustein optimiert ist, gilt er als Standard-Datenbaustein. Ein Standard-DB ist mit den Programmierwerkzeugen von STEP 7 Classic und den klassischen S7-300 und S7-400 CPUs kompatibel. Datenbausteine mit optimiertem Zugriff haben keine fest definierte Struktur. Die Datenelemente enthalten nur einen symbolischen Namen in der Deklaration und keine feste Adresse im Baustein. Die CPU speichert die Elemente automatisch im verfügbaren Speicherbereich des Bausteins, so dass keine Lücken im Speicher entstehen. Dadurch wird die Speicherkapazität optimal ausgenutzt.

Um für einen Datenbaustein den optimierten Zugriff festzulegen, gehen Sie wie folgt vor:

1. Erweitern Sie den Ordner der Programmbausteine in der STEP 7 Projektnavigation.
2. Klicken Sie mit der rechten Maustaste auf den Datenbaustein und wählen Sie im Kontextmenü den Befehl "Eigenschaften".
3. Wählen Sie in den Attributen die Option "Optimierter Bausteinzugriff".

Beachten Sie, dass bei neuen Datenbausteinen standardmäßig der optimierte Bausteinzugriff eingestellt wird. Wenn Sie "Optimierter Bausteinzugriff" deaktivieren, wird der Standardzugriff für den Baustein festgelegt.

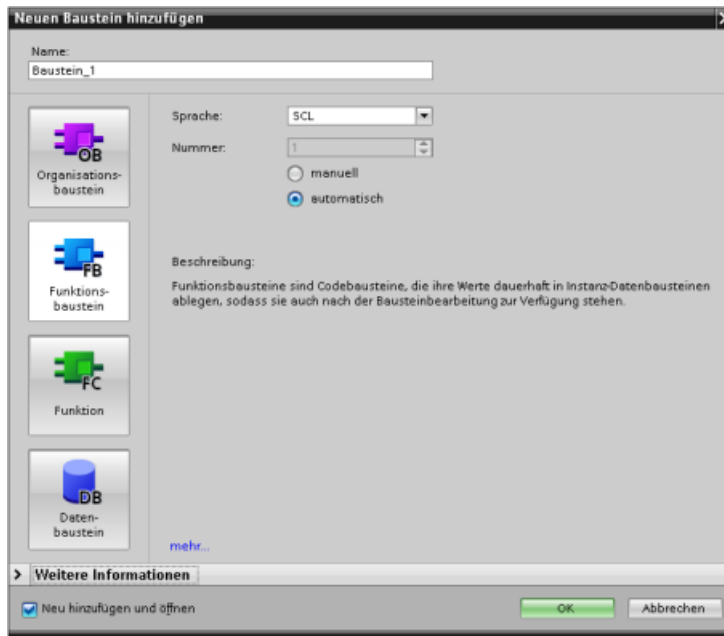
Hinweis

Bausteinzugriffstyp für einen FB und seinen Instanz-DB

Wenn für Ihren FB die Option "Optimierter Bausteinzugriff" eingestellt ist, muss für den Instanz-DB dieses FBs auch die Option "Optimierter Bausteinzugriff" eingestellt sein. Ebenso gilt, dass, wenn für den FB die Option "Optimierter Bausteinzugriff" nicht eingestellt ist und es sich somit um einen FB mit Standardzugriff handelt, auch für den Instanz-DB der Standardzugriff und nicht die Option "Optimierter Bausteinzugriff" eingestellt sein muss.

Wenn die Zugriffstypen nicht kompatibel sind, können bei der Ausführung des FBs über ein HMI-Gerät vorgenommene Änderungen an den IN/OUT-Parameterwerten des FBs verlorengehen.

7.3.5 Anlegen wiederverwendbarer Codebausteine



OBs, FBs, FCs und globale DBs können Sie in der Projektnavigation unter "Programmbausteine" im Dialog "Neuen Baustein hinzufügen" anlegen.

Beim Anlegen eines Codebausteins müssen Sie die Programmiersprache für den Baustein auswählen. Für einen DB wird keine Sprache ausgewählt, weil er nur Daten speichert.

Bei Auswahl des Kontrollkästchens "Neu hinzufügen und öffnen" (Standardeinstellung) wird der Codebaustein in der Projektsicht geöffnet.

Objekte, die Sie wiederverwenden möchten, können Sie in Bibliotheken speichern. Jedes Projekt besitzt eine Projektbibliothek, die mit dem Projekt verbunden ist. Zusätzlich zur Projektbibliothek können Sie beliebig viele globale Bibliotheken erstellen, die in mehreren Projekten verwendet werden können. Da die Bibliotheken untereinander kompatibel sind, können Bibliothekselemente von einer Bibliothek in die andere kopiert oder verschoben werden.

Bibliotheken dienen beispielsweise dazu, Bausteinvorlagen zu erstellen, die Sie zunächst in die Projektbibliothek kopieren und dann dort weiterentwickeln. Abschließend kopieren Sie die Bausteine von der Projektbibliothek in eine globale Bibliothek. Die globale Bibliothek stellen Sie anderen Kollegen im Projekt zur Verfügung. Ihre Kollegen können die Bausteine verwenden und bei Bedarf weiter an ihre jeweiligen Anforderungen anpassen.

Einzelheiten zu Bibliotheksoperationen finden Sie in den Themen zu Bibliotheken in der STEP 7 Online-Hilfe.

7.3.6 Übergabe von Parametern an Bausteine

Funktionsbausteine (FBs) und Funktionen (FCs) haben drei verschiedene Schnittstellentypen:

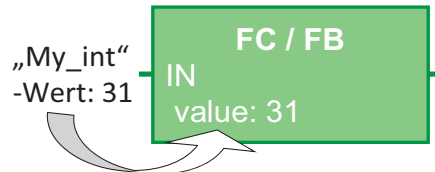
- IN
- IN/OUT
- OUT

FBs und FCs empfangen Parameter über die Schnittstellentypen IN und IN/OUT. Die Bausteine verarbeiten die Parameter und geben über die Schnittstellentypen IN/OUT und OUT Werte an den Aufrufer zurück.

Das Anwenderprogramm überträgt Parameter mittels einer von zwei Methoden:

Call-by-Value

Wenn das Anwenderprogramm einen Parameter als "Call-by-Value" an eine Funktion übergibt, kopiert das Anwenderprogramm den tatsächlichen Parameterwert in den Eingangsparameter des Bausteins für den Schnittstellentyp IN. Dieser Vorgang erfordert zusätzlichen Speicher für den kopierten Wert.



Wenn das Anwenderprogramm den Baustein aufruft, kopiert es die Werte.

Call-by-Reference

Wenn das Anwenderprogramm einen Parameter als "Call-by-Reference" an eine Funktion übergibt, verweist das Anwenderprogramm auf die Adresse des tatsächlichen Parameters für den Schnittstellentyp IN/OUT und kopiert den Wert nicht. Dieser Vorgang erfordert keinen zusätzlichen Speicher.



Wenn das Anwenderprogramm den Baustein aufruft, verweist es auf die Adresse der tatsächlichen Parameter.

Hinweis

Der Schnittstellentyp IN/OUT wird im Allgemeinen für strukturierte Variablen (zum Beispiel ARRAY, STRUCT und STRING) verwendet, um eine unnötige Vergrößerung des erforderlichen Datenspeichers zu vermeiden.

Bausteinoptimierung und Parameterübergabe

Bei einfachen Datentypen (zum Beispiel INT, DINT und REAL) übergibt das Anwenderprogramm FC-Parameter als "Call-by-Value". Komplexe Datentypen (zum Beispiel STRUCT, ARRAY und STRING) werden als "Call-by-Reference" übergeben.

Das Anwenderprogramm übergibt FB-Parameter normalerweise in dem zum FB zugehörigen Instanzdatenbaustein (DB):

- Das Anwenderprogramm übergibt einfache Datentypen (zum Beispiel INT, DINT und REAL) als "Call-by-Value", indem die Parameter in den und aus dem Instanz-DB kopiert werden.
- Das Anwenderprogramm kopiert komplexe Datentypen (zum Beispiel STRUCT, ARRAY und STRING) für die Parametertypen IN und OUT in den und aus dem Instanz-DB.
- Das Anwenderprogramm übergibt komplexe Datentypen als "Call-by-Reference" für den Schnittstellentyp IN/OUT.

DBs können mit den Optionen "Optimiert" oder "Standard" (nicht optimiert) angelegt werden. Die optimierten Datenbausteine sind kompakter als die nicht optimierten Datenbausteine. Auch unterscheidet sich die Reihenfolge der Datenelemente innerhalb der DBs bei optimierten und nicht optimierten DBs. Im Abschnitt "Optimierte Bausteine" in der S7-Programmerrichtlinie für S7-1200/1500, STEP 7 (TIA Portal), 03/2014 (<https://support.industry.siemens.com/cs/ww/de/view/81318674>) finden Sie eine Erläuterung zu optimierten Bausteinen.

Sie erstellen FBs und FCs, um entweder optimierte oder nicht optimierte Daten zu verarbeiten. Sie können das Kontrollkästchen "Optimierter Bausteinzugriff" als eines der Attribute für den Baustein aktivieren. Das Anwenderprogramm optimiert Programmbausteine standardmäßig und die Programmbausteine erwarten die an den Baustein übergebenen Daten im optimierten Format.

Wenn das Anwenderprogramm einen komplexen Parameter (zum Beispiel STRUCT) an eine Funktion übergibt, prüft das System die Optimierungseinstellung des Datenbausteins mit der Struktur und die Optimierungseinstellung des Programmbausteins. Wenn Sie sowohl den Datenbaustein als auch die Funktion optimieren, dann übergibt das Anwenderprogramm die Struktur als "Call-by-Reference". Gleiches gilt, wenn Sie sowohl den Datenbaustein als auch die Funktion nicht optimieren.

Wenn jedoch Funktion und Datenbaustein unterschiedliche Optimierungseinstellungen haben (d. h., ein Baustein ist optimiert und der andere nicht), dann muss die Struktur in das von der Funktion erwartete Format umgewandelt werden. Beispiel: Wenn der Datenbaustein nicht optimiert und die Funktion optimiert ist, dann muss eine Struktur im Datenbaustein in ein optimiertes Format umgewandelt werden, damit die Funktion die Struktur verarbeiten kann. Das System führt diese Umwandlung durch, indem es die Struktur "kopiert" und sie in das von der Funktion erwartete optimierte Format umwandelt.

Zusammenfassend lässt sich sagen, dass, wenn das Anwenderprogramm einen komplexen Datentyp (zum Beispiel STRUCT) als IN/OUT-Parameter an eine Funktion übergibt, die Funktion erwartet, dass das Anwenderprogramm die Struktur als "Call-by-Reference" übergibt:

- Wenn Sie sowohl den Datenbaustein mit der Struktur als auch die Funktion optimieren bzw. nicht optimieren, dann übergibt das Anwenderprogramm die Daten als "Call-by-Reference".
- Wenn Sie für den Datenbaustein und die Funktion nicht die gleichen Optimierungseinstellungen konfigurieren (ein Baustein ist optimiert, der andere ist nicht optimiert), dann muss das System vor der Übergabe an die Funktion eine Kopie der Struktur anlegen. Weil das System diese Kopie der Struktur anlegen muss, wird "Call-by-Reference" dadurch praktisch in "Call-by-Value" umgewandelt.

Auswirkung der Optimierungseinstellungen auf Anwenderprogramme

Das Kopieren der Parameter kann ein Problem in einem Anwenderprogramm verursachen, wenn durch ein HMI-Gerät oder einen Alarm-OB Elemente in der Struktur verändert werden. Beispiel: Es gibt einen IN/OUT-Parameter einer Funktion (der normalerweise als "Call-by-Reference" übergeben wird), doch die Optimierungseinstellungen des Datenbausteins und der Funktion unterscheiden sich:

1. Wenn das Anwenderprogramm die Funktion aufrufen will, muss das System eine "Kopie" der Struktur anlegen, um das Format der Daten an die Funktion anzupassen.
2. Das Anwenderprogramm ruft die Funktion mit einem Verweis auf die "Kopie" der Struktur auf.
3. Ein Alarm-OB tritt während der Ausführung der Funktion auf und der Alarm-OB ändert einen Wert in der ursprünglichen Struktur.
4. Die Funktion wird abgearbeitet und, weil die Struktur ein IN/OUT-Parameter ist, kopiert das System die Werte zurück in die ursprüngliche Struktur im ursprünglichen Format.

Weil eine Kopie der Struktur zum Ändern des Formats angelegt wurde, gehen die vom Alarm-OB geschriebenen Daten verloren. Das Gleiche kann beim Schreiben eines Werts mit einem HMI-Gerät passieren. Das HMI-Gerät kann das Anwenderprogramm unterbrechen und in derselben Weise wie ein Alarm-OB einen Wert schreiben.

Es gibt mehrere Möglichkeiten, dieses Problem zu vermeiden:

- Die beste Lösung ist, bei Verwendung komplexer Datentypen (zum Beispiel STRUCT) die Optimierungseinstellungen des Programmbausteins und des Datenbausteins aneinander anzupassen. Dadurch wird gewährleistet, dass das Anwenderprogramm die Parameter stets als "Call-by-Reference" übergibt.
- Eine andere Lösung ist die, dass ein Alarm-OB bzw. ein HMI-Gerät ein Element in der Struktur nicht direkt ändert. Der OB oder das HMI-Gerät kann eine andere Variable ändern und dann können Sie diese Variable an einem bestimmten Punkt im Anwenderprogramm in die Struktur kopieren.

7.4 Datenkonsistenz

Die CPU erhält die Datenkonsistenz für alle elementaren Datentypen (z.B. Word oder DWord) und alle systemdefinierten Strukturen (z.B. IEC_TIMERS oder DTL). Der Lese- bzw. Schreibvorgang des Werts kann nicht unterbrochen werden. (Die CPU z.B. schützt den Zugriff auf einen Wert vom Datentyp DWord, bis die vier Bytes des DWord gelesen oder geschrieben wurden.) Um sicherzustellen, dass die Programmzyklus-OBs und die Alarm-OBs nicht gleichzeitig in dieselbe Adresse im Speicher schreiben können, führt die CPU einen Alarm-OB erst aus, wenn der Lese- oder Schreibvorgang im Programmzyklus-OB beendet ist.

Wenn in Ihrem Anwenderprogramm mehrere Werte im Speicher von einem Programmzyklus-OB und einem Alarm-OB gemeinsam genutzt werden, muss Ihr Anwenderprogramm auch sicherstellen, dass diese Werte konsistent geändert oder gelesen werden. Mit den Anweisungen

DIS_AIRT (Alarmbearbeitung deaktivieren) und EN_AIRT (Alarmbearbeitung aktivieren) können Sie in Ihrem Programmzyklus-OB den Zugriff auf die gemeinsam genutzten Werte schützen.

- Fügen Sie eine Anweisung DIS_AIRT in den Codebaustein ein, um sicherzustellen, dass während eines Lese- oder Schreibvorgangs kein Alarm-OB ausgeführt werden kann.
- Fügen Sie die Anweisungen ein, die die Werte, die von einem Alarm-OB geändert werden könnten, lesen oder schreiben.
- Fügen Sie am Ende der Sequenz eine Anweisung EN_AIRT ein, um die Anweisung DIS_AIRT zu stornieren und die Ausführung eines Alarm-OBs zu gestatten.

Auch eine Kommunikationsanforderung eines HMI-Geräts oder einer anderen CPU kann die Ausführung des Programmzyklus-OBs unterbrechen. Die Kommunikationsanforderungen können auch zu Problemen mit der Datenkonsistenz führen. Die CPU stellt sicher, dass die elementaren Datentypen von den Anweisungen im Anwenderprogramm stets konsistent gelesen und geschrieben werden. Weil das Anwenderprogramm regelmäßig von Kommunikationsanforderungen unterbrochen wird, kann nicht gewährleistet werden, dass mehrere Werte der CPU alle gleichzeitig von der HMI aktualisiert werden. Die in einem HMI-Bild angezeigten Werte beispielsweise können aus unterschiedlichen Zyklen der CPU stammen.

Die PtP-Anweisungen (Punkt-zu-Punkt), PROFINET-Anweisungen (wie TSEND_C und TRCV_C), die PROFINET-Anweisungen für die dezentrale Peripherie (Seite 371) und die PROFIBUS-Anweisungen für die dezentrale Peripherie (Seite 371) übertragen Datenpuffer, die unterbrochen werden können. Stellen Sie die Datenkonsistenz für die Datenpuffer dadurch sicher, dass Sie Lese- und Schreibvorgänge in den Puffern sowohl im Programmzyklus-OB als auch im Alarm-OB verhindern. Sollte es erforderlich sein, die Pufferwerte für diese Anweisungen in einem Alarm-OB zu ändern, verzögern Sie mit der Anweisung DIS_AIRT die Unterbrechung (durch einen Alarm-OB oder eine Kommunikationsunterbrechung von einem HMI-Gerät oder einer anderen CPU), bis eine Anweisung EN_AIRT ausgeführt wird.

Hinweis

Die Verwendung der Anweisung DIS_AIRT verzögert die Verarbeitung von Alarm-OBs, bis die Anweisung EN_AIRT ausgeführt wird, und wirkt sich auf die Alarmlatenz (Zeitraum vom Auftreten eines Ereignisses bis zur Ausführung des Alarm-OBs) Ihres Anwenderprogramms aus.

7.5 Programmiersprache

STEP 7 bietet die folgenden Standardprogrammiersprachen für die S7-1200:

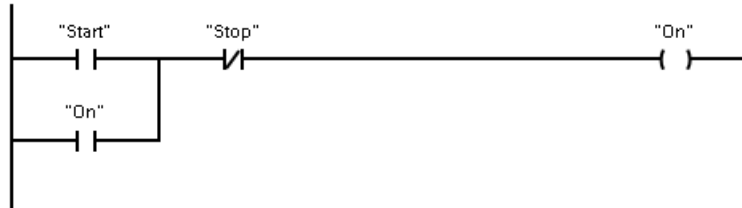
- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen (Seite 189).
- FUP (Funktionsplan) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra (Seite 190) basiert.
- SCL (Structured Control Language) ist eine textbasierte, höhere Programmiersprache (Seite 190).

Wenn Sie einen Codebaustein anlegen, müssen Sie die Programmiersprache für den Baustein auswählen.

Ihr Anwenderprogramm kann mit Codebausteinen arbeiten, die in einer dieser Programmiersprachen angelegt wurden.

7.5.1 Kontaktplan (KOP)

Die Elemente eines Schaltplans, wie Öffner- und Schließerkontakte, und Spulen werden zu Netzwerken verknüpft.



Um Verknüpfungen für komplexe Operationen anzulegen, können Sie Verzweigungen für parallele Kreise einfügen. Parallele Verzweigungen sind nach unten geöffnet oder direkt mit der Stromschiene verbunden. Sie beenden die Verzweigungen nach unten.

KOP bietet Box-Anweisungen für eine Vielzahl von Funktionen wie Arithmetik, Zeiten, Zähler und Übertragen.

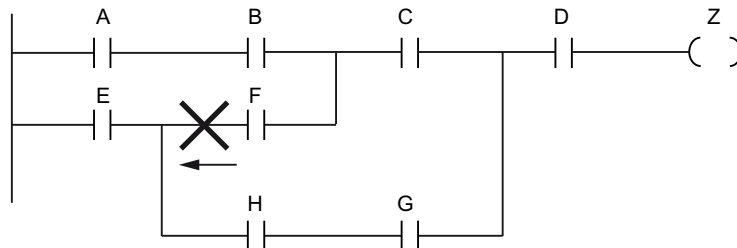
STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem KOP-Netzwerk nicht.

Hinweis

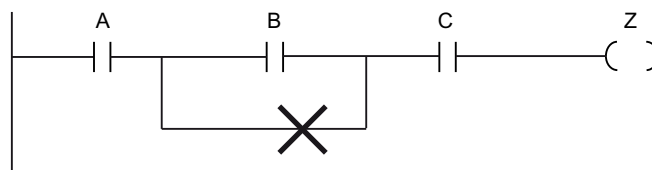
Jedes KOP-Netzwerk muss mit einer Spule oder einer Box abgeschlossen werden.

Beim Anlegen eines KOP-Netzwerks sind die folgenden Regeln zu beachten:

- Sie können keine Verzweigung anlegen, die zu einem Signalfluss in die Gegenrichtung führen könnte.

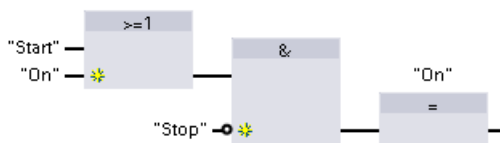


- Sie können keine Verzweigung anlegen, die einen Kurzschluss verursachen würde.



7.5.2 Funktionsplan (FUP)

Ebenso wie KOP ist auch FUP eine grafische Programmiersprache. Die Darstellung der Verknüpfungslogik beruht auf den grafischen Symbolen, die in der booleschen Algebra üblich sind.



Um Verknüpfungen für komplexe Operationen anzulegen, fügen Sie parallele Verzweigungen zwischen den Boxen ein.

Arithmetische Funktionen und andere komplexe Funktionen können direkt in Verbindung mit den Logikboxen dargestellt werden.

STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem FUP-Netzwerk nicht.

7.5.3 SCL

Structured Control Language (SCL) ist eine höhere, PASCAL-basierte Programmiersprache für die SIMATIC S7-CPU. SCL unterstützt die Bausteinstruktur von STEP 7 (Seite 178). Ihr Projekt kann Programmbausteine in jeder der drei Programmiersprachen: SCL, KOP und FUP enthalten.

In SCL-Anweisungen werden die Standardoperatoren der Programmierung verwendet, z. B. für Zuweisung ($:=$), mathematische Funktionen (+ für Addition, - für Subtraktion, * für Multiplikation und / für Division). SCL verwendet auch standardmäßige PASCAL-Operationen für die Programmsteuerung, z. B. IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO und RETURN. Für syntaktische Elemente der Programmiersprache SCL können Sie alle PASCAL-Referenzen verwenden. Viele der anderen Anweisungen für SCL wie Zeiten und Zähler entsprechen den Anweisungen in KOP und FUP. Weitere Informationen zu spezifischen Anweisungen finden Sie unter den jeweiligen Anweisungen in den Kapiteln Grundlegende Anweisungen (Seite 209) und Erweiterte Anweisungen (Seite 327).

7.5.3.1 SCL-Programmiereditor

Sie können für alle Bausteintypen (OB, FB oder FC) beim Erstellen des Bausteins angeben, dass er die Programmiersprache SCL verwenden soll. STEP 7 verfügt über einen SCL-Programmiereditor, der die folgenden Elemente enthält:

- Schnittstellenabschnitt zum Definieren der Parameter des Codebausteins
- Codeabschnitt für den Programmcode
- Anweisungsverzeichnis mit den SCL-Anweisungen, die von der CPU unterstützt werden

Sie geben den SCL-Code für Ihre Anweisung direkt in den Codeabschnitt ein. Der Editor enthält Schaltflächen für gängige Codekonstruktionen und Kommentare. Um komplexere Anweisungen anzulegen, ziehen Sie die SCL-Anweisungen einfach mit der Maus aus dem Anweisungsverzeichnis in Ihr Programm. Ferner können Sie in jedem Texteditor ein SCL-Programm anlegen und die Datei dann in STEP 7 importieren.

Function_1			
	Name	Datentyp	Kommentar
1	Input		
2	StartStopSwitch	Bool	
3	Output		
4	RunYesNo	Bool	
5	InOut		
6	<Add new>		
7	Temp		
8	<Add new>		
9	Constant		
10	<Add new>		
11	Return		
12	Function_1	Void	


```

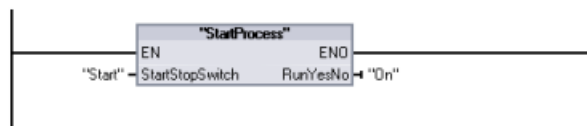
IF... CASE... FOR... WHILE... (*...*)
  OF... TO DO... DO...
1 IF condition THEN
2   // Statement section IF
3   ;
4 END_IF;

```

Im Schnittstellenabschnitt des SCL-Codebausteins können Sie die folgenden Arten von Parametern deklarieren:

- Input, Output, InOut und Ret_Val: Diese Parameter definieren die Eingangs- und Ausgangsvariablen sowie den Rückgabewert für den Codebaustein. Der Variablenname, den Sie hier eingeben, wird lokal während der Ausführung des Codebausteins verwendet. Üblicherweise wird der globale Variablenname nicht in der Variablen-tabelle verwendet.
- Static (nur FBs; die obige Abbildung zeigt einen FC): Der Codebaustein nutzt statische Variablen zum Speichern von statischen Zwischenergebnissen im Instanzdatenbaustein. Der Baustein speichert statische Daten, bis sie überschrieben werden, d. h. zum Teil über mehrere Zyklen. Zusammen mit den statischen Daten werden auch die Namen der Bausteine gespeichert, die in diesem Codebaustein als Multiinstanz-DBs aufgerufen werden.
- Temp: Bei diesen Parametern handelt es sich um temporäre Variablen, die während der Ausführung des Codebausteins verwendet werden.
- Constant: Hierbei handelt es sich um benannte konstante Werte für Ihren Codebaustein.

Wenn Sie den SCL-Codebaustein aus einem anderen Codebaustein aufrufen, treten die Parameter des SCL-Codebausteins als Eingänge oder Ausgänge auf.



In diesem Beispiel entsprechen die Variablen für "Start" und "On" (aus der Variablen-tabelle des Systems) den Variablen "StartStopSwitch" und "RunYesNo" in der Deklarations-tabelle des SCL-Programms.

7.5.3.2 SCL-Ausdrücke und -Operationen

SCL-Ausdruck konstruieren

Ein SCL-Ausdruck ist eine Formel zum Berechnen eines Werts. Der Ausdruck besteht aus Operanden und Operatoren (wie *, /, + oder -). Bei den Operanden kann es sich um Variablen, Konstanten oder Ausdrücke handeln.

Die Auswertung des Ausdrucks erfolgt in einer bestimmten Reihenfolge, die von den folgenden Faktoren festgelegt wird:

- Jeder Operator hat eine vordefinierte Priorität, wobei die Operation mit der höchsten Priorität zuerst ausgeführt wird.
- Bei Operatoren mit gleicher Priorität werden die Operatoren von links nach rechts verarbeitet.
- Mit Hilfe von Klammern kennzeichnen Sie eine Reihe von Operatoren, die gemeinsam ausgewertet werden sollen.

Mit dem Ergebnis eines Ausdrucks kann ein Wert einer von Ihrem Programm verwendeten Variablen zugewiesen werden, das Ergebnis kann als Bedingung für eine Steuerungsanweisung oder als Parameter für eine andere SCL-Anweisung oder zum Aufrufen eines Codebausteins verwendet werden.

Tabelle 7-2 Operatoren in SCL

Typ	Operation	Operator	Priorität
Klammern	<i>Ausdruck</i>)	(,)	1
Arithmetik	Potenz	**	2
	Vorzeichen (unäres Plus)	+	3
	Vorzeichen (unäres Minus)	-	3
	Multiplikation	*	4
	Division	/	4
	Modulo-Funktion	MOD	4
	Addition	+	5
	Subtraktion	-	5
Vergleich	Kleiner als	<	6
	Kleiner oder gleich	<=	6
	Größer als	>	6
	Größer oder gleich	>=	6
	Gleich	=	7
	Ungleich	<>	7
Bitverknüpfung	Negation (unär)	NOT	3
	Logische UND-Verknüpfung	AND oder &	8
	Logische Exklusiv-ODER-Verknüpfung	XOR	9
	Logische ODER-Verknüpfung	OR	10
Zuweisung	Zuweisung	:=	11

Als höhere Programmiersprache nutzt SCL Standardanweisungen für grundlegende Aufgaben:

- Zuweisungsanweisung: :=
- Arithmetische Funktionen: +, -, * und /
- Adressierung von globalen Variablen: "<Variablenname>" (Variablenname oder Datenbausteinname in doppelten Anführungszeichen)
- Adressierung von lokalen Variablen: #<Variablenname> (Variablenname mit vorangestelltem Symbol "#")

Die folgenden Beispiele zeigen verschiedene Ausdrücke für verschiedene Einsatzzwecke:

<code>"C" := #A+#B;</code>	Weist einer Variablen die Summe zweier lokaler Variablen zu
<code>"Data_block_1".Tag := #A;</code>	Zuweisung zu einer Datenbausteinvariablen
<code>IF #A > #B THEN "C" := #A;</code>	Bedingung für die IF-THEN-Anweisung
<code>"C" := SQRT (SQR (#A) + SQR (#B));</code>	Parameter für die SQRT-Anweisung

Arithmetische Operatoren können verschiedene numerische Datentypen verarbeiten. Der Datentyp des Ergebnisses wird vom Datentyp des höchstwertigen Operanden festgelegt. Beispiel: Eine Multiplikationsoperation mit einem Operanden vom Typ INT und einem Operanden vom Typ REAL ergibt einen Wert vom Typ REAL als Ergebnis.

Steuerungsanweisungen

Eine Steuerungsanweisung ist eine besondere Art von SCL-Ausdruck, der die folgenden Aufgaben durchführt:

- Programmverzweigung
- Wiederholung von Abschnitten des SCL-Programmcodes
- Sprung zu anderen Teilen des SCL-Programms
- Bedingte Ausführung

Die SCL-Steuerungsanweisungen umfassen IF-THEN, CASE-OF, FOR-TO-DO, WHILE-DO, REPEAT-UNTIL, CONTINUE, GOTO und RETURN.

Eine einzelne Anweisung belegt üblicherweise eine Codezeile. Sie können mehrere Anweisungen in einer Zeile eingeben oder Sie können die Anweisung auf mehrere Codezeilen verteilen, damit der Code besser lesbar ist. Trennzeichen (wie Tabulatoren, Zeilenumbrüche und zusätzliche Leerzeichen) werden während der Syntaxprüfung ignoriert. Die END-Anweisung beendet die Steuerungsanweisung.

Die folgenden Beispiele zeigen eine FOR-TO-DO-Steuerungsanweisung. (Beide Formen der Codierung sind syntaktisch gültig.)

```
FOR x := 0 TO max DO sum := sum + value(x); END_FOR;
FOR x := 0 TO max DO
    sum := sum + value(x);
END_FOR;
```

Eine Steuerungsanweisung kann auch eine Sprungmarke umfassen. Eine Sprungmarke wird gefolgt von einem Doppelpunkt an den Anfang der Anweisung gesetzt:

Sprungmarke: <Anweisung>;

Die Online-Hilfe von STEP 7 enthält vollständige Referenzinformationen für die Programmiersprache SCL.

Bedingungen

Eine Bedingung ist ein Vergleichsausdruck oder ein logischer Ausdruck, dessen Ergebnis vom Typ BOOL ist (Wert WAHR oder FALSCH). Die folgenden Beispiele zeigen Bedingungen verschiedener Arten:

<code>#Temperatur > 50</code>	Relationaler Ausdruck
<code>#Zähler <= 100</code>	
<code>#CHAR1 < 'S'</code>	
<code>(&#Alpha <> 12) AND NOT #Beta</code>	Vergleichsausdruck und logischer Ausdruck
<code>5 + #Alpha</code>	Arithmetischer Ausdruck

Eine Bedingung kann arithmetische Ausdrücke verwenden:

- Die Bedingung des Ausdrucks ist WAHR, wenn das Ergebnis ein beliebiger Wert außer Null ist.
- Die Bedingung des Ausdrucks ist FALSCH; wenn das Ergebnis null ist.

Andere Codebausteine aus Ihrem SCL-Programm aufrufen

Um einen anderen Codebaustein in Ihrem Anwenderprogramm aufzurufen, geben Sie einfach den Namen (oder die absolute Adresse) von FB oder FC mit den Parametern ein. Für einen FB müssen Sie den Instanz-DB angeben, der mit dem FB aufgerufen werden soll.

<code><DB-Name> (Parameterliste)</code>	Aufruf als eine Instanz
<code><#Instanzname> (Parameterliste)</code>	Aufruf als Multiinstanz
<code>"MyDB" (MyInput:=10, MyInOut:="Tag1") ;</code>	
<code><FC-Name> (Parameterliste)</code>	Standardaufruf
<code><Operand>:=<FC-Name> (Parameterliste)</code>	Aufruf in einem Ausdruck
<code>"MyFC" (MyInput:=10, MyInOut:="Tag1") ;</code>	

Sie können Bausteine auch mit der Maus aus der Projektnavigation in den SCL-Programmieditor ziehen und die Parametrierung vervollständigen.

Bausteinkommentare in SCL-Code hinzufügen

Sie können in Ihrem SCL-Code einen Bausteinkommentar hinzufügen, indem Sie Kommentartext zwischen (* und *) einfügen. Sie können zwischen (* und *) eine beliebige Anzahl Kommentarzeilen einfügen. Ihr SCL-Programmbaustein kann viele Bausteinkommentare enthalten. Zur leichteren Programmierung enthält der SCL-Editor neben den üblichen Steuerungsanweisungen eine Schaltfläche für Bausteinkommentare:



Adressierung

Wie bei KOP und FUP können Sie in SCL entweder Variablen (symbolische Adressierung) oder absolute Adressen in Ihrem Anwenderprogramm verwenden. In SCL können Sie eine Variable auch als Array-Index verwenden.

Absolute Adressierung

`%I0.0`

`%MB100`

Der absoluten Adresse ist das Symbol "%" voranzustellen. Fehlt das "%", erzeugt STEP 7 beim Übersetzen einen undefinierten Variablenfehler.

Symbolische Adressierung

`"PLC_Tag_1"`

`"Data_block_1".Tag_1`

`"Data_block_1".MyArray[#i]`

Variable in PLC-Variablen-Tabelle

Variable in einem Datenbaustein

Array-Element in einem Datenbaustein-Array

7.5.3.3 Indexierte Adressierung mit den Anweisungen PEEK und POKE

SCL bietet die Anweisungen PEEK und POKE, mit denen Sie aus Datenblöcken, E/A oder dem Speicher lesen oder in diese schreiben können. Sie geben für die Operation Parameter mit einem spezifischen Byte- oder Bit-Versatz an.

Hinweis

Um die Anweisungen PEEK und POKE mit Datenbausteinen zu verwenden, müssen Sie Standarddatenbausteine (keine optimierten Datenbausteine) verwenden. Beachten Sie zudem, dass die Anweisungen PEEK und POKE lediglich Daten übertragen. Die Datentypen an den Adressen werden nicht beachtet.

```
PEEK(area:=_in_,  
      dbNumber:=_in_,  
      byteOffset:=_in_);
```

Liest das von byteOffset angegebene Byte aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel für den Verweis auf einen Datenbaustein:

```
%MB100 := PEEK(area:=16#84,  
              dbNumber:=1, byteOffset:=#i);
```

Beispiel für den Verweis auf Eingang EB3:

```
%MB100 := PEEK(area:=16#81,  
              dbNumber:=0, byteOffset:=#i); // when  
#i = 3
```

```
PEEK_WORD(area:=_in_,  
          dbNumber:=_in_,  
          byteOffset:=_in_);
```

Liest das von byteOffset angegebene Wort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MW200 := PEEK_WORD(area:=16#84,  
                   dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD (area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

```
PEEK_BOOL (area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

```
POKE (area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

```
POKE_BOOL (area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_,
           value:=_in_);
```

```
POKE_BLK (area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

Liest das von byteOffset angegebene Doppelwort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MD300 := PEEK_DWORD (area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

Liest einen von bitOffset und byteOffset angegebenen Booleschen Wert aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MB100.0 := PEEK_BOOL (area:=16#84,
                      dbNumber:=1, byteOffset:=#i,
                      bitOffset:=#j);
```

Schreibt den Wert (Byte, Wort oder Doppelwort) in den angegebenen byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel für den Verweis auf einen Datenbaustein:

```
POKE (area:=16#84, dbNumber:=2,
     byteOffset:=3, value:="Tag_1");
```

Beispiel für den Verweis auf Ausgang AB3:

```
POKE (area:=16#82, dbNumber:=0,
     byteOffset:=3, value:="Tag_1");
```

Schreibt den Booleschen Wert in den angegebenen bitOffset und byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel:

```
POKE_BOOL (area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5,
           value:=0);
```

Schreibt die unter "count" angegebene Anzahl von Bytes beginnend mit dem angegebenen byteOffset des angegebenen Quelldatenbausteins, der Quell-E/A oder des Quellspeicherbereichs in den angegebenen byteOffset des angegebenen Zieldatenbausteins, der Ziel-E/A oder des Zielspeicherbereichs.

Beispiel:

```
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);
```

Bei den Anweisungen PEEK und POKE gelten die folgenden Werte für die Parameter "area", "area_src" und "area_dest". Für andere Bereiche als Datenbausteine muss der Parameter dbNumber 0 sein.

16#81	I
16#82	Q
16#83	M
16#84	DB

7.5.4 EN und ENO in KOP, FUP und SCL

"Signalfluss" (EN und ENO) für eine Anweisung ermitteln

Einige Anweisungen (z. B. mathematische Anweisungen und Übertragungsanweisungen) zeigen Parameter für EN und ENO an. Diese Parameter beziehen sich auf den Signalfluss in KOP oder FUP und legen fest, ob die Anweisung in diesem Zyklus ausgeführt wird. In SCL können Sie den Parameter ENO auch für einen Codebaustein angeben.

- EN (Freigabeeingang) ist ein Boolescher Eingang. An diesem Eingang muss Signalfluss (EN = 1) vorhanden sein, damit die Box ausgeführt werden kann. Wenn der Eingang EN einer KOP-Box direkt an die linke Stromschiene angeschlossen ist, wird die Anweisung immer ausgeführt.
- ENO (Freigabeausgang) ist ein Boolescher Ausgang. Liegt am Eingang EN einer Box ein Signalfluss an und die Box wird fehlerfrei ausgeführt, dann leitet der Ausgang ENO den Signalfluss (ENO = 1) zum nächsten Element weiter. Tritt während der Ausführung der Box ein Fehler auf, dann wird der Signalfluss an der Box-Anweisung, die den Fehler verursacht hat, beendet (ENO = 0).

Tabelle 7-3 Operanden für EN und ENO

Programm-Editor	Eingänge/Ausgänge	Operanden	Datentyp
KOP	EN, ENO	Signalfluss	Bool
FUP	EN	E, E:P, A, M, DB, Temp, Signalfluss	Bool
	ENO	Signalfluss	Bool
SCL	EN ¹	WAHR, FALSCH	Bool
	ENO ²	WAHR, FALSCH	Bool

¹ Die Verwendung von EN steht nur bei FBs zur Verfügung.

² Die Verwendung von ENO mit dem SCL-Codebaustein ist optional. Sie müssen die SCL-Übersetzung so einrichten, dass ENO bei Fertigstellung der Codebausteinbearbeitung gesetzt wird.

Setzen von ENO in SCL konfigurieren

Um die SCL-Übersetzung so einzurichten, dass ENO gesetzt wird, gehen Sie folgendermaßen vor:

1. Wählen Sie im Menü "Extras" den Befehl "Einstellungen".

2. Erweitern Sie die Eigenschaften "PLC-Programmierung" und wählen Sie "SCL (Structured Control Language)".
3. Wählen Sie die Option "ENO automatisch setzen".

Verwendung von ENO im Programmcode

Sie können ENO auch in Ihrem Programmcode verwenden, zum Beispiel durch Zuweisen von ENO zu einer PLC-Variablen oder durch Auswerten von ENO in einem lokalen Baustein.

Beispiele:

```

"MyFunction"
  ( IN1 := ... ,
    IN2 := ... ,
    OUT1 => #myOut,
    ENO => #statusFlag ); // PLC tag statusFlag holds the value of ENO

"MyFunction"
  ( IN1 := ...
    IN2 := ... ,
    OUT1 => #myOut,
    ENO => ENO ); // block status flag of "MyFunction"
                  // is stored in the local block

IF ENO = TRUE THEN
  // execute code only if MyFunction returns true ENO
    
```

Auswirkung der Parameter Ret_Val oder Status auf ENO

Einige Anweisungen, z. B. die Kommunikationsanweisungen oder die Anweisungen für die Zeichenkettenkonvertierung, verfügen über einen Ausgangsparameter, der Informationen über die Verarbeitung der Anweisung enthält. Einige Anweisungen haben beispielsweise einen Parameter Ret_Val (Rückgabewert), üblicherweise vom Datentyp Int, der Statusinformationen im Bereich von -32768 bis +32767 enthält. Andere Anweisungen haben den Parameter Status, typischerweise vom Datentyp Word, der Statusinformationen im Hexadezimalbereich von 16#0000 bis 16#FFFF speichert. Der numerische Wert in einem der Parameter Ret_Val oder Status gibt den Zustand von ENO der jeweiligen Anweisung an.

- Ret_Val: Ein Wert zwischen 0 und 32767 setzt ENO typischerweise = 1 (bzw. WAHR). Ein Wert zwischen -32768 und -1 setzt ENO typischerweise = 0 (bzw. FALSCH). Um Ret_Val auszuwerten, ändern Sie die Darstellung in hexadezimal.
- Status: Ein Wert zwischen 16#0000 und 16#7FFF setzt ENO typischerweise = 1 (bzw. WAHR). Ein Wert zwischen 16#8000 und 16#FFFF setzt ENO typischerweise = 0 (bzw. FALSCH).

Anweisungen, die über mehrere Zyklen ausgeführt werden, haben häufig den Parameter Busy (Bool), um kenntlich zu machen, dass die Anweisung aktiv ist, die Ausführung jedoch noch nicht beendet ist. Solche Anweisungen haben auch häufig auch den Parameter Done (Bool) und den

Parameter Error (Bool). Done zeigt an, dass die Anweisung fehlerfrei ausgeführt wurde, und Error weist darauf hin, dass die Anweisung mit Fehler beendet wurde.

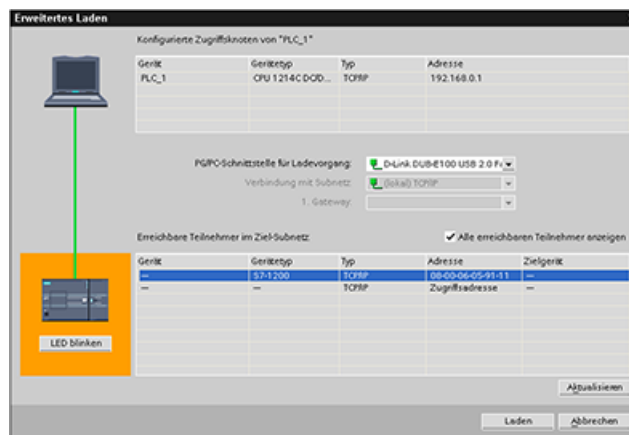
- Wenn Busy = 1 (bzw. WAHR), ist ENO = 1 (bzw. WAHR).
- Wenn Done = 1 (bzw. WAHR), ist ENO = 1 (bzw. WAHR).
- Wenn Error = 1 (bzw. WAHR), ist ENO = 0 (bzw. FALSCH).

Siehe auch

OK (Gültigkeit prüfen) und NOT_OK (Ungültigkeit prüfen) (Seite 232)

7.6 Laden der Programmelemente

Sie können die Elemente Ihres Projekts aus dem Programmiergerät in die CPU laden. Wenn Sie ein Projekt laden, speichert die CPU das Anwenderprogramm (OBs, FCs, FBs und DBs) im internen Ladespeicher oder, sofern eine SIMATIC Memory Card vorhanden ist, im externen Ladespeicher (Karte).



Sie können Ihr Projekt aus dem Programmiergerät in die CPU laden, und zwar von einer der folgenden Stellen:

- Projektnavigation: Klicken Sie mit der rechten Maustaste auf das Programmelement und wählen Sie dann "Laden in CPU".
- Onlinemenü: Klicken Sie auf die Option "Laden in Gerät".
- Symbolleiste: Klicken Sie auf das Symbol "Laden in Gerät".
- Gerätekonfiguration: Klicken Sie mit der rechten Maustaste auf die CPU und wählen Sie die zu ladenden Elemente aus.

Ist für einen Programmbaustein eine dynamische Verknüpfung mit obligatorischem Passwort (Seite 168) festgelegt, muss zum Laden der geschützten Bausteine das Passwort eingegeben

werden. Ist diese Art des Kopierschutzes für mehrere Bausteine eingerichtet, muss für jeden der geschützten Bausteine das Passwort für das Laden des Bausteins eingegeben werden.

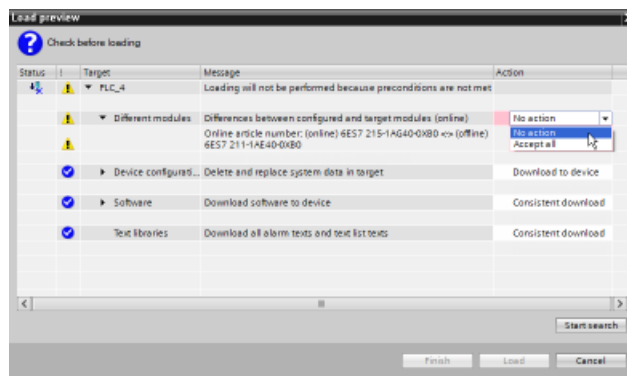
Hinweis

Durch das Laden eines Programms werden die Werte im remanenten Speicher weder gelöscht noch verändert. Soll der remanente Speicher vor dem Laden gelöscht werden, ist vor dem Laden des Programms die CPU auf die Werkseinstellungen zurückzusetzen.

Sie können auch einPanel-Projekt für die Basic HMI-Panels (Seite 32) vom TIA Portal auf eine Memory Card in der S7-1200 CPU laden.

Laden ins Zielsystem, wenn sich die konfigurierte CPU von der angeschlossenen CPU unterscheidet

STEP 7 und die S7-1200 gestatten das Laden ins Zielsystem, wenn die angeschlossene CPU die Kapazität hat, die geladenen Daten von der konfigurierten CPU basierend auf den Speichieranforderungen des Projekts und der Kompatibilität der Peripherie zu speichern. Sie können die Konfiguration und das Programm von einer CPU in eine größere CPU laden, beispielsweise von einer CPU 1211C DC/DC/DC in eine CPU 1215C DC/DC/DC, weil die Peripherie kompatibel ist und der Speicher ausreicht. In diesem Fall werden während des Ladevorgangs die Warnung "Unterschiede zwischen den konfigurierten Baugruppen und den Zielbaugruppen (online)" sowie im Dialog "Vorschau Laden" die Artikelnummern und die Firmware-Versionen angezeigt. Sie müssen entweder "Keine Aktion" wählen, wenn der Ladevorgang nicht fortgesetzt werden soll, oder "Alle übernehmen", wenn Sie mit dem Laden fortfahren möchten:



Hinweis

Wenn Sie nach dem Laden der konfigurierten CPU in eine andere angeschlossene CPU online gehen (Seite 1209), wird das Projekt für die konfigurierte CPU im Projektbaum mit Online-Statusanzeigen angezeigt. In der Ansicht "Online und Diagnose" jedoch wird der tatsächlich angeschlossene CPU-Modultyp angezeigt.

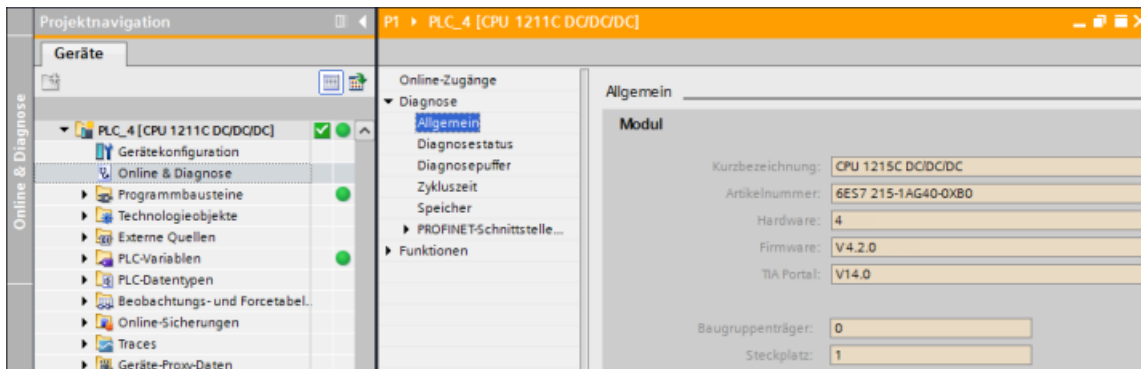


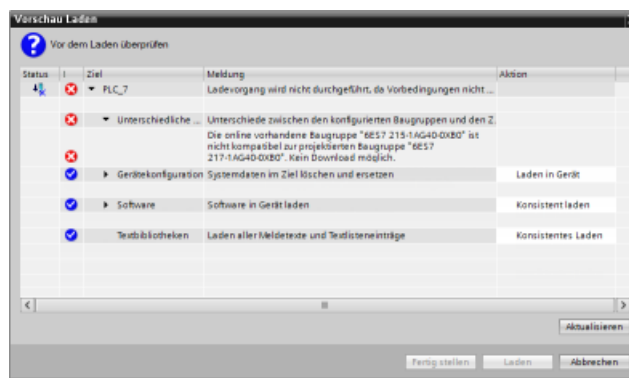
Bild 7-1 Online-Ansicht, wenn sich die konfigurierte CPU von der angeschlossenen CPU unterscheidet

Sie können natürlich in der Gerätekonfiguration Ihr Gerät ändern (Seite 152), so dass die konfigurierte CPU den gleichen Modultyp hat wie die angeschlossene CPU. Im Dialog "Gerät ändern" werden vollständige Informationen zur Kompatibilität angezeigt, wenn Sie versuchen, ein Gerät zu ändern.

STEP 7 und die S7-1200 verbieten das Laden ins Zielsystem, wenn die angeschlossene CPU nicht die Kapazität hat, die aus der konfigurierten CPU geladenen Daten zu speichern. Beispielsweise können Sie die Hardwarekonfiguration und das Programm in den folgenden Fällen nicht laden:

- Von einer CPU 1215C DC/DC/DC in eine CPU 1212C DC/DC/DC, weil der Arbeitsspeicher nicht ausreicht
- Von einer CPU 1211C DC/DC/Relais in eine CPU 1211C DC/DC/DC, weil es Unterschiede in der Peripherie gibt
- Von einer CPU 1217C DC/DC/DC in eine beliebige CPU 1211C, CPU 1212C, CPU 1214C oder CPU 1215C, weil die CPU 1217C über 1,5-V-Ausgänge verfügt
- Von einer CPU 1214C V4.2.x in eine CPU 1214C V4.0, weil die Firmware-Version nicht abwärtskompatibel ist

In solchen Fällen zeigt der Dialog "Vorschau Laden" einen Fehler an:



CPUs mit Schutz vertraulicher PLC-Konfigurationsdaten

Wenn Sie Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161) konfiguriert haben, beachten Sie Folgendes:

- Wenn das Gerät das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten nicht hat, werden Sie vom TIA Portal beim ersten Laden ins Zielsystem aufgefordert, das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten einzugeben.
- Wenn das Gerät das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten bereits hat, müssen das Passwort im STEP 7-Projekt und im Gerät übereinstimmen. Stimmen die Passwörter nicht überein, können Sie das Projekt nicht in die CPU laden. Sie müssen das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten löschen oder dafür das gleiche Passwort wie das im Gerät festlegen. Sie können das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten auch unter "Online & Diagnose (Seite 1213)" festlegen oder löschen.

Wiederherstellen nach fehlgeschlagenem Laden ins Zielsystem

Wenn der Ladevorgang fehlschlägt, wird im Register "Info" im Inspektorfenster der Grund angezeigt. Auch der Diagnosepuffer liefert Informationen. Nach einem fehlgeschlagenen Ladevorgang gehen Sie wie folgt vor, um das Laden ins Zielsystem erfolgreich durchzuführen:

1. Beheben Sie das Problem wie in der Fehlermeldung beschrieben.
2. Versuchen Sie erneut, die Daten ins Zielsystem zu laden.

In seltenen Fällen wird der Ladevorgang erfolgreich durchgeführt, doch ein anschließendes Aus- und Wiedereinschalten der CPU schlägt fehl. In diesem Fall wird im Diagnosepuffer möglicherweise ein Fehler wie der folgende angezeigt:

- 16# 02:4175 -- CPU-Fehler: Auswertungsfehler Memory Card: Unbekannte oder inkompatible Version des aktuellen Kartentyps in der CPU-Konfigurationsbeschreibung: Keine Memory-Card-Funktion fertiggestellt/abgebrochen, neue Anlaufsperr festgelegt: ...- Memory Card fehlt, falscher Typ, falscher Inhalt oder geschützt

Tritt diese Situation auf und weitere Ladeversuche schlagen fehl, müssen Sie den internen Ladespeicher oder den externen Ladespeicher löschen:

1. Wenn Sie mit dem internen Ladespeicher arbeiten, setzen Sie die CPU auf die Werkseinstellungen zurück.
2. Wenn Sie mit einer SIMATIC Memory Card arbeiten, entnehmen Sie sie und löschen den Inhalt der Memory Card (Seite 124). Stecken Sie die Karte dann erneut.
3. Laden Sie die Hardwarekonfiguration und die Software ins Zielsystem.

Siehe auch

Synchronisieren der Online-CPU und des Offline-Projekts (Seite 203)

7.7 Synchronisieren der Online-CPU und des Offline-Projekts

Wenn Sie Projektbausteine in die CPU laden, kann die CPU erkennen, ob sich seit dem letzten Ladevorgang Bausteine oder Variablen in der Online-CPU geändert haben. Dann bietet Ihnen die CPU an, die Änderungen zu synchronisieren. Das bedeutet, dass Sie die Änderungen aus der Online-CPU ins Projekt laden können, bevor Sie das Projekt in die CPU laden. Änderungen in der Online-CPU können auf eine Vielzahl von Faktoren zurückzuführen sein:

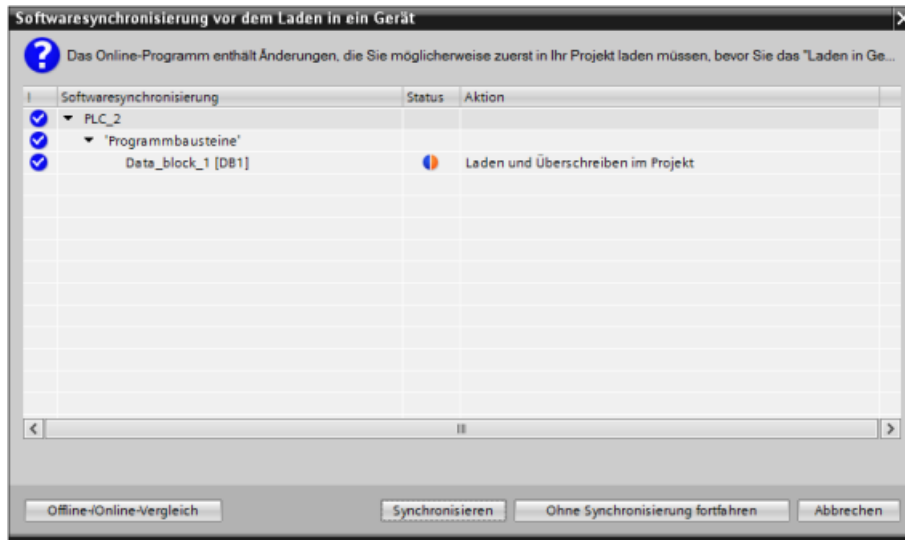
- Änderungen an den Startwerten von Datenbausteinvariablen während der Laufzeit, zum Beispiel durch die Anweisung WRIT_DBL (Seite 522) oder durch Laden eines Rezepts
- Ein Ladevorgang aus einem "sekundären" Projekt (ein anderes Projekt als das Projekt des letzten Ladevorgangs), bei dem eine oder mehrere der folgenden Bedingungen bestehen:
 - Die Online-CPU enthält Programmbausteine, die nicht im Projekt vorhanden sind.
 - Datenbausteinvariablen oder Bausteinattribute sind im Offline-Projekt und in der Online-CPU unterschiedlich.
 - In der Online-CPU sind PLC-Variablen vorhanden, die im Offline-Projekt nicht vorhanden sind.

Hinweis

Wenn Sie in dem Projekt, das Sie für den letzten Ladevorgang verwendet haben, Bausteine oder Variablen bearbeiten, brauchen Sie hinsichtlich der Synchronisierung keine Wahl zu treffen. STEP 7 und die CPU erkennen, dass die Änderungen im Offline-Projekt neuer sind als die Online-CPU und fährt mit dem normalen Ladevorgang fort.

Synchronisierungsoptionen

Wenn Sie ein Projekt in die CPU laden, wird der Synchronisierungsdialog angezeigt, wenn STEP 7 erkennt, dass Datenbausteine oder Variablen in der Online-CPU neuer sind als die Werte im Projekt. Beispiel: Wenn das STEP 7-Programm WRIT_DBL ausgeführt und einen Startwert für eine Variable in Data_block_1 geändert hat, zeigt STEP 7 den folgenden Synchronisierungsdialog an, sobald Sie das Laden ins Zielsystem anstoßen:



Dieser Dialog führt die Programmbausteine auf, in denen Unterschiede vorhanden sind. In diesem Dialog haben Sie folgende Möglichkeiten:

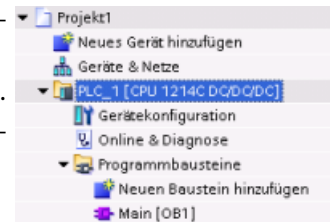
- **Online/Offline-Vergleich:** Wenn Sie auf diese Schaltfläche klicken, zeigt STEP 7 die Programmbausteine, Systembausteine, Technologieobjekte, PLC-Variablen und PLC-Datentypen für das Projekt als mit der Online-CPU verglichen (Seite 1221) an. Für jedes Objekt können Sie mittels Klick eine ausführliche Analyse der Unterschiede einschließlich Zeitstempel anzeigen. Anhand dieser Informationen können Sie entscheiden, wie Sie mit den Unterschieden zwischen der Online-CPU und dem Projekt umgehen möchten.
- **Synchronisieren:** Wenn Sie auf diese Schaltfläche klicken, lädt STEP 7 die Datenbausteine, Variablen und anderen Objekte aus der Online-CPU ins Projekt. Sie können anschließend damit fortfahren, das Programm ins Zielsystem zu laden, sofern nicht die Programmausführung erneut dazu geführt hat, dass das Projekt mit der CPU nicht synchron ist.
- **Ohne Synchronisierung fortfahren:** Wenn Sie auf diese Schaltfläche klicken, lädt STEP 7 das Projekt in die CPU.
- **Abbrechen:** Wenn Sie auf diese Schaltfläche klicken, brechen Sie den Ladevorgang ab.

7.8 Laden von der Online-CPU

Sie können die Programmbausteine einer Online-CPU oder einer an Ihr Programmiergerät angeschlossenen Memory Card auch kopieren.

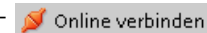
Bereiten Sie das Offline-Projekt für die kopierten Programmbausteine vor:

1. Fügen Sie eine CPU hinzu, die der Online-CPU entspricht.
2. Erweitern Sie den CPU-Knoten, so dass der Ordner "Programmbausteine" angezeigt wird.

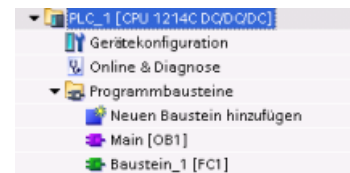


Um die Programmbausteine aus der Online-CPU in das Offline-Projekt zu laden, gehen Sie folgendermaßen vor:

1. Klicken Sie im Offline-Projekt auf den Ordner "Programmbausteine".
2. Klicken Sie auf die Schaltfläche "Online verbinden".
3. Wählen Sie die Schaltfläche "Laden aus CPU".
4. Bestätigen Sie den Vorgang im Dialog Laden aus CPU (Seite 1209).



Nach dem Laden zeigt STEP 7 alle geladenen Programmbausteine im Projekt.



7.8.1 Vergleich der Online-CPU mit der Offline-CPU

Mit dem Editor "Vergleichen" (Seite 1221) in STEP 7 ermitteln Sie Unterschiede zwischen dem Online- und dem Offline-Projekt. Dies kann vor dem Laden des Programms aus der CPU nützlich sein.

7.9 Debugging und Testen des Programms

7.9.1 Daten in der CPU beobachten und steuern

Sie können Werte in der Online-CPU beobachten und steuern (siehe folgende Tabelle).

Tabelle 7-4 Daten mit STEP 7 beobachten und steuern

Editor	Beobachten	Steuern	Forcen
Beobachtungstabelle	Ja	Ja	Nein
Forcetabelle	Ja	Nein	Ja

Editor	Beobachten	Steuern	Forcen
Programmieditor	Ja	Ja	Nein
Variablentabelle	Ja	Nein	Nein
DB-Editor	Ja	Nein	Nein



In einer Beobachtungstabelle beobachten



Im KOP-Editor beobachten

Im Kapitel "Online und Diagnose" finden Sie weitere Informationen zum Thema Daten in der CPU beobachten und steuern (Seite 1223).

7.9.2 Beobachtungstabellen und Forcetabellen

Mit Hilfe von "Beobachtungstabellen" können Sie die Werte eines Anwenderprogramms, das von der Online-CPU ausgeführt wird, überwachen und ändern. Sie können in Ihrem Projekt unterschiedliche Beobachtungstabellen erstellen und speichern, um eine Vielzahl von Testumgebungen abzudecken. So können Sie Tests zum Beispiel bei der Inbetriebnahme oder für Service- und Wartungszwecke durchführen.

Mit einer Beobachtungstabelle können Sie die Ausführung des Anwenderprogramms durch die CPU überwachen und in die Ausführung eingreifen. Sie können nicht nur für die Variablen der Codebausteine und Datenbausteine, sondern auch für die Speicherbereiche der CPU, einschließlich Eingänge und Ausgänge (E und A), periphere Eingänge (E:P), Merker (M) und Datenbausteine (DB) Werte aufrufen und ändern.

Mit der Beobachtungstabelle können Sie die physischen Ausgänge (A:P) einer CPU, die sich im Betriebszustand STOP befindet, freigeben. Beispielsweise können Sie den Ausgängen bestimmte Werte zuweisen, während Sie die Verdrahtung der CPU testen.

STEP 7 bietet zudem eine Forcetabelle zum Forcen einer Variablen auf einen bestimmten Wert. Weitere Informationen zum Forcen finden Sie im Abschnitt zum Forcen von Werten in der CPU (Seite 1230) im Kapitel "Online und Diagnose".

Hinweis

Die Forcewerte werden in der CPU und nicht in der Beobachtungstabelle gespeichert.

Sie können keinen Eingang forcen (Adresse "E"). Sie können jedoch einen Peripherieingang forcen. Um einen Peripherieingang zu forcen, hängen Sie ein ":P" an die Adresse an (Beispiel: "On:P").

Mit STEP 7 ist es ferner möglich, Programmvariablen anhand von Auslösebedingungen zu verfolgen und aufzuzeichnen (Seite 1242).

7.9.3 Querverweis zum Anzeigen der Verwendung

Das Inspektorfenster zeigt Querverweise dazu an, wie ein Objekt innerhalb des gesamten Projekts verwendet wird, z. B. im Anwenderprogramm, in der CPU oder den HMI-Geräten. Im Register "Querverweis" werden die Instanzen angezeigt, wo und von welchen anderen Objekten ein ausgewähltes Objekt verwendet wird. Das Inspektorfenster enthält außerdem Bausteine, die nur online innerhalb der Querverweise verfügbar sind. Um die Querverweise anzuzeigen, wählen Sie den Befehl "Querverweise anzeigen". (In der Projektansicht befindet sich dieser Befehl im Menü "Werkzeuge".)

Hinweis

Zum Anzeigen der Querverweisinformationen muss der Editor nicht geschlossen werden.

Die Einträge der Querverweisliste können verschieden sortiert werden. Die Liste der Querverweise bietet einen Überblick über die Verwendung von Speicheradressen und Variablen im Anwenderprogramm.

- Wenn Sie ein Programm anlegen oder ändern, behalten Sie einen Überblick über die verwendeten Operanden, Variablen und Bausteinaufrufe.
- Aus den Querverweisen können Sie direkt an die Stelle springen, an der die Operanden und Variablen verwendet werden.
- Während eines Programmtests oder einer Fehlerbehebung erhalten Sie Informationen dazu, welche Speicheradresse von welchem Befehl in welchem Baustein verarbeitet wird, welche Variable in welchem Bild verwendet wird und welcher Baustein von welchem anderen Baustein aufgerufen wird.

Tabelle 7-5 Querverweiselemente

Spalte	Beschreibung
Objekt	Name des Objekts, das die angegebenen unterlagerten Objekte verwendet oder das von den unterlagerten Objekten verwendet wird
Anzahl	Anzahl Verwendungen
Verwendung	Der Ort der Verwendung, z. B. ein Netzwerk
Eigenschaft	Besondere Eigenschaften der referenzierten Objekte, z. B. die Variablennamen in Multiinstanz-Deklarationen
als	Zeigt zusätzliche Informationen zum Objekt an, z. B. ob ein Instanz-DB als Vorlage oder als Multiinstanz verwendet wird
Zugriff	Art des Zugriffs, d. h. ob auf den Operanden Lesezugriff (R) und/oder Schreibzugriff (W) besteht
Adresse	Adresse des Operanden
Typ	Angabe, mit welchem Typ und welcher Sprache das Objekt angelegt wurde
Pfad	Pfad des Objekts in der Projektnavigation

Abhängig von den installierten Produkten werden in der Querverweistabelle zusätzliche oder unterschiedliche Spalten angezeigt.

7.9.4 Aufrufstruktur zur Prüfung der Aufrufhierarchie

Die Aufrufstruktur zeigt die Aufrufhierarchie des Bausteins innerhalb Ihres Anwenderprogramms. Sie bietet einen Überblick über die verwendeten Bausteine, die Aufrufe anderer Bausteine, die Beziehungen zwischen Bausteinen, die Datenanforderungen an jeden Baustein sowie den Status der einzelnen Bausteine. Die Bausteine in der Aufrufstruktur können mit dem Programmiereditor geöffnet und bearbeitet werden.

Durch Anzeigen der Aufrufstruktur erhalten Sie eine Liste der im Anwenderprogramm verwendeten Bausteine. STEP 7 zeigt die erste Ebene der Aufrufstruktur hervorgehoben an und zeigt auch die Bausteine an, die durch keinen anderen Baustein im Programm aufgerufen werden. Die erste Ebene der Aufrufstruktur enthält die OBs sowie diejenigen FCs, FBs und DBs, die nicht durch einen OB aufgerufen werden. Von anderen Bausteinen aufgerufene Codebausteine erscheinen eingerückt unter dem aufrufenden Baustein. In der Aufrufstruktur werden nur die Bausteine angezeigt, die von einem Codebaustein aufgerufen werden.

Sie können selektiv nur jene Bausteine anzeigen, die innerhalb der Aufrufstruktur Konflikte verursachen. Folgende Bedingungen führen zu Konflikten:

- Bausteine, die Aufrufe mit älteren oder neueren Zeitstempeln im Code ausführen
- Bausteine, die einen Baustein mit geänderter Schnittstelle aufrufen
- Bausteine, die eine Variable mit geänderter Adresse und/oder geändertem Datentyp verwenden
- Bausteine, die weder direkt noch indirekt durch einen OB aufgerufen werden
- Bausteine, die einen nicht vorhandenen oder fehlenden Baustein aufrufen

Sie können mehrere Bausteinaufrufe und Datenbausteine zu einer Gruppe zusammenfassen. Über eine Klappliste können Sie die Verknüpfungen mit den verschiedenen Aufrufstellen anzeigen.

Sie können außerdem eine Konsistenzprüfung durchführen, um Zeitstempelkonflikte aufzuzeigen. Zeitstempelkonflikte können durch die Änderung des Zeitstempels eines Bausteins während oder nach der Programmgenerierung verursacht werden. Diese Konflikte führen zu Inkonsistenzen zwischen den aufrufenden und den aufgerufenen Bausteinen.

- Die meisten Zeitstempel- und Schnittstellenkonflikte lassen sich durch erneutes Übersetzen der Codebausteine beheben.
- Wenn durch Übersetzen die Inkonsistenzen nicht beseitigt werden, navigieren Sie mit dem Programmiereditor über die Verknüpfung in der Spalte "Details" zur Quelle des Problems. Hier können Sie die Inkonsistenzen manuell beseitigen.
- Sind Bausteine rot markiert, müssen sie erneut übersetzt werden.

Anweisungen

8.1 Bitverknüpfungen

8.1.1 Bitverknüpfungsanweisungen

KOP und FUP verarbeiten Boolesche Logik sehr effektiv. SCL ist zwar besonders effektiv bei komplexen mathematischen Berechnungen und bei Projektsteuerstrukturen, doch Sie können SCL auch für Boolesche Logik verwenden.

KOP-Kontakte

Tabelle 8-1 Schließer- und Öffnerkontakte



KOP	SCL	Beschreibung
"IN" 	<pre>IF in THEN Statement; ELSE Statement; END_IF;</pre>	Schließer- und Öffnerkontakte: Sie können Kontakte untereinander verschalten und so Ihre eigene Verschaltungslogik erstellen. Nutzt das von Ihnen angegebene Eingangsbit die Speicherkennung E (Eingang) oder A (Ausgang), so wird der Bitwert aus dem Prozessabbildregister gelesen. Die physischen Kontaktsignale in Ihrem Steuerungsprozess werden mit Eingangsanschlüssen der PLC-Geräts verschaltet. Die CPU fragt die verschalteten Eingangssignale ab und aktualisiert fortlaufend die entsprechenden Zustandswerte im Prozessabbild der Eingänge.
"IN" 	<pre>IF NOT (in) THEN Statement; ELSE Statement; END_IF;</pre>	Das direkte Lesen eines physischen Eingangs führen Sie mit ":P" nach der E-Adresse durch (Beispiel: "%E3.4:P"). Beim direkten Lesen werden die Bitdatenwerte direkt aus dem physischen Eingang und nicht aus dem Prozessabbild gelesen. Beim direkten Lesen wird das Prozessabbild nicht aktualisiert.

Tabelle 8-2 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Bool	Zugewiesenes Bit

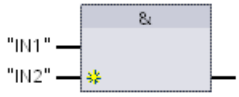
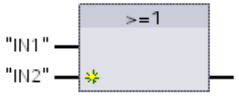

- Der Schließer ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 1 ist.
- Der Öffner ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 0 ist.
- In Reihe geschaltete Kontakte bilden logische UND-Verknüpfungen.
- Parallel geschaltete Kontakte bilden logische ODER-Verknüpfungen.

UND-, ODER- und XOR-Boxen in FUP

Bei der FUP-Programmierung werden Netzwerke mit KOP-Kontakten in die Box-Netzwerke UND (&), ODER (≥ 1) und EXKLUSIV ODER (x) umgewandelt, in denen Sie Bitwerte für die Ein- und Ausgänge der Box angeben können. Sie können ferner Verschaltungen mit anderen Logik-Boxen herstellen und so Ihre eigene Verschaltungslogik erstellen. Nachdem die Box in Ihrem Netzwerk platziert ist, können Sie die Funktion "Eingang einfügen" aus der Funktionsleiste "Favoriten" oder dem Anweisungsverzeichnis zur Eingangsseite der Box ziehen, um weitere Eingänge hinzuzufügen. Sie können auch mit der rechten Maustaste auf den Eingangsanschluss der Box klicken und "Eingang einfügen" auswählen.

Die Ein- und Ausgänge einer Box können mit anderen Boxen verschaltet werden oder Sie können eine Bitadresse oder einen Bitsymbolnamen für einen unverschalteten Eingang eingeben. Bei der Ausführung der Box-Anweisung werden die Eingangszustände auf die binäre Box-Verknüpfung geschaltet und dann der Box-Ausgang, sofern zutreffend, auf Wahr gesetzt.

Tabelle 8-3 Boxen UND, ODER und EXKLUSIV ODER

FUP	SCL ¹	Beschreibung
	<code>out := in1 AND in2;</code>	Damit der Ausgang WAHR ist, müssen alle Eingänge einer UND-Box WAHR sein
	<code>out := in1 OR in2;</code>	Damit der Ausgang WAHR ist, muss ein beliebiger Eingang einer ODER-Box WAHR sein
	<code>out := in1 XOR in2;</code>	Damit der Ausgang WAHR ist, muss eine ungerade Anzahl der Eingänge einer XOR-Box WAHR sein

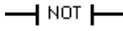
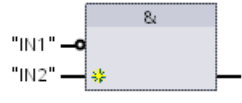
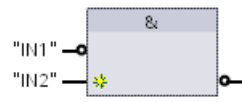
¹ In SCL: Sie müssen das Ergebnis der Operation einer Variable zuweisen, damit es in einer anderen Anweisung verwendet werden kann.

Tabelle 8-4 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN1, IN2	Bool	Eingangsbit

Logikinvertierer NOT

Tabelle 8-5 VKE (Verknüpfungsergebnis) invertieren

KOP	FUP	SCL	Beschreibung
	 	NOT	<p>Bei der FUP-Programmierung können Sie die Funktion "VKE invertieren" aus der Funktionsleiste "Favoriten" oder dem Anweisungsverzeichnis auf einen Eingang oder einen Ausgang ziehen, um einen Logikinvertierer für diesen Box-Anschluss zu erstellen.</p> <p>Der KOP-Kontakt NOT invertiert den logischen Zustand des Signalflusseingangs.</p> <ul style="list-style-type: none"> Ist kein Signalfluss zum NOT-Kontakt vorhanden, so steht ein Signalfluss am Ausgang an. Ist ein Signalfluss zum NOT-Kontakt vorhanden, so steht kein Signalfluss am Ausgang an.

Ausgangsspule und Zuweisungsbox

Die Anweisung für den Spulenausgang schreibt einen Wert in ein Ausgangsbit. Nutzt das angegebene Ausgangsbit die Speicherkennung A, so schaltet die CPU das Ausgangsbit im Prozessabbildregister ein oder aus und setzt das angegebene Bit jeweils entsprechend dem Signalfluss. Die Ausgangssignale für Ihre Steuerstellglieder werden mit den Ausgangsklemmen der CPU verschaltet. Im Betriebszustand RUN fragt die CPU fortlaufend die Eingangssignale ab, verarbeitet die Eingangszustände gemäß der Programmlogik und reagiert dann, indem sie die neuen Ausgangswerte im Prozessabbild der Ausgänge setzt. Die CPU überträgt die im Prozessabbildregister gespeicherte Reaktion auf den neuen Ausgangszustand zu den verschalteten Ausgangsklemmen.

Tabelle 8-6 Zuweisung vornehmen und Zuweisung negieren

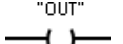
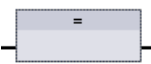
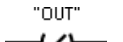
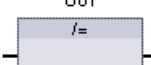
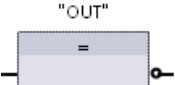
KOP	FUP	SCL	Beschreibung
		<code>out := <Boolescher Ausdruck>;</code>	<p>Bei der FUP-Programmierung werden die KOP-Spulen in Zuweisungsboxen (= und /=) umgewandelt, wobei eine Bitadresse für den Boxausgang anzugeben ist. Die Ein- und Ausgänge einer Box können mit anderen Boxverknüpfungen verschaltet werden oder Sie können eine Bitadresse eingeben.</p> <p>Das direkte Schreiben eines physischen Ausgangs geben Sie mit ":P" nach der A-Adresse an (Beispiel: "%A3.4:P"). Beim direkten Schreiben werden die Bitdatenwerte in den Ausgang im Prozessabbild und direkt in den physischen Ausgang geschrieben.</p>
		<code>out := NOT <Boolescher Ausdruck>;</code>	
			

Tabelle 8-7 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
OUT	Bool	Zugewiesenes Bit

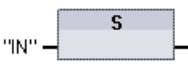
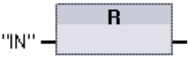
8.1 Bitverknüpfungen

- Ist ein Signalfluss durch eine Ausgangsspule vorhanden oder eine FUP-Box "=" aktiviert, so wird das Ausgangsbit auf 1 gesetzt.
- Ist kein Signalfluss durch eine Ausgangsspule vorhanden oder keine FUP-Box "=" aktiviert, so wird das Ausgangsbit auf 0 gesetzt.
- Ist ein Signalfluss durch eine invertierte Ausgangsspule vorhanden oder eine FUP-Box "/=" aktiviert, so wird das Ausgangsbit auf 0 gesetzt.
- Ist kein Signalfluss durch eine invertierte Ausgangsspule vorhanden oder keine FUP-Box "/=" aktiviert, so wird das Ausgangsbit auf 1 gesetzt.

8.1.2 Setz- und Rücksetzoperationen

1 Bit setzen und rücksetzen

Tabelle 8-8 Anweisungen S und R

KOP	FUP	SCL	Beschreibung
"OUT" —(S)—	"OUT" 	Nicht verfügbar	Ausgang setzen: Wird S (Setzen) aktiviert, wird der Datenwert an Adresse OUT auf 1 gesetzt. Wird S nicht aktiviert, verändert sich OUT nicht.
"OUT" —(R)—	"OUT" 	Nicht verfügbar	Ausgang rücksetzen: Wird R (Rücksetzen) aktiviert, wird der Datenwert an Adresse OUT auf 0 gesetzt. Wird R nicht aktiviert, verändert sich OUT nicht.

¹ In KOP und FUP: Diese Anweisungen können an jeder beliebigen Stelle im Netzwerk eingefügt werden.





² In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 8-9 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN (oder Verschalten mit Kontakt-/Steuerungslogik)	Bool	Bitvariable der zu beobachtenden Adresse
OUT	Bool	Bitvariable der zu setzenden oder rückzusetzenden Adresse

Bitfeld setzen und rücksetzen

Tabelle 8-10 Anweisungen SET_BF und RESET_BF

KOP ¹	FUP	SCL	Beschreibung
		Nicht verfügbar	Bitfeld setzen: Wird SET_BF aktiviert, so wird der Datenwert 1 in "n" Bits geschrieben, beginnend an Adresse OUT. Wird SET_BF nicht aktiviert, verändert sich OUT nicht.
		Nicht verfügbar	Bitfeld rücksetzen: RESET_BF schreibt den Datenwert 0 in "n" Bits, beginnend an Adresse OUT. Wird RESET_BF nicht aktiviert, verändert sich OUT nicht.

¹ In KOP und FUP: Diese Anweisungen müssen ganz rechts in einer Verzweigung angeordnet werden.

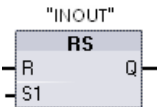
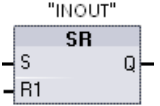
² In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 8-11 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
OUT	Bool	Startelement des Bitfelds, das gesetzt oder zurückgesetzt werden soll (Beispiel: #MyArray[3])
n	Konstante (UInt)	Anzahl der zu schreibenden Bits

Flipflop-Schaltungen vorrangig setzen und vorrangig rücksetzen

Tabelle 8-12 Anweisungen RS und SR

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	Flipflop rücksetzen/setzen: RS ist ein Flipflop, bei dem das Setzen Vorrang hat. Sind die Signale Setzen (S1) und Rücksetzen (R) beide wahr, so lautet die Ausgangsadresse INOUT 1.
	Nicht verfügbar	Flipflop rücksetzen/setzen: SR ist ein Flipflop, bei dem das Rücksetzen Vorrang hat. Sind die Signale Setzen (S) und Rücksetzen (R1) beide wahr, so lautet die Ausgangsadresse INOUT 0.

¹ In KOP und FUP: Diese Anweisungen müssen ganz rechts in einer Verzweigung angeordnet werden.

² In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 8-13 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
S, S1	Bool	Eingang setzen; 1 weist auf Dominanz hin
R, R1	Bool	Eingang zurücksetzen; 1 weist auf Dominanz hin

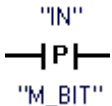
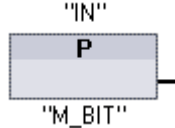
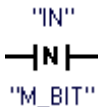
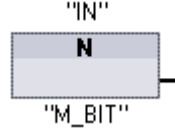
Parameter	Datentyp	Beschreibung
INOUT	Bool	Zugewiesene Bitvariable "INOUT"
Q	Bool	Folgt dem Zustand von Bit "INOUT"

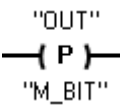
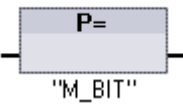
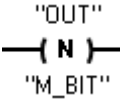
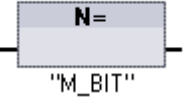
Variable "INOUT" weist die Bitadresse zu, die gesetzt bzw. zurückgesetzt wird. Der optionale Ausgang Q gibt den Signalzustand der Adresse "INOUT" an.

Anweisung	S1	R	Bit "INOUT"
RS	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	1
SR	S	R1	
	0	0	Vorheriger Zustand
	0	1	0
	1	0	1
	1	1	0

8.1.3 Operationen Steigende Flanke und Fallende Flanke



Tabelle 8-14 Anweisungen "Steigende Flanke" und "Fallende Flanke"

KOP	FUP	SCL	Beschreibung
		Nicht verfügbar ¹	<p>Operand auf positive Signalflanke abfragen.</p> <p>KOP: Der Zustand dieses Kontakts ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am zugewiesenen Bit IN erkannt wird. Der logische Zustand des Kontakts wird dann mit dem Eingangszustand des Signalflusses verknüpft, um den Ausgangszustand des Signalflusses zu setzen. Der Kontakt P kann an einer beliebigen Stelle im Netzwerk angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p> <p>FUP: Der logische Zustand des Ausgangs ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am zugewiesenen Eingangsbit erkannt wird. Die Box P darf nur am Anfang einer Verzweigung angeordnet werden.</p>
		Nicht verfügbar ¹	<p>Operand auf negative Signalflanke abfragen.</p> <p>KOP: Der Zustand dieses Kontakts ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am zugewiesenen Eingangsbit erkannt wird. Der logische Zustand des Kontakts wird dann mit dem Eingangszustand des Signalflusses verknüpft, um den Ausgangszustand des Signalflusses zu setzen. Der Kontakt N kann an einer beliebigen Stelle im Netzwerk angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p> <p>FUP: Der logische Zustand des Ausgangs ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am zugewiesenen Eingangsbit erkannt wird. Die Box N darf nur am Anfang einer Verzweigung angeordnet werden.</p>

KOP	FUP	SCL	Beschreibung
 <p>"OUT" "M_BIT"</p>	 <p>"OUT" "M_BIT"</p>	Nicht verfügbar ¹	<p>Operand bei positiver Signalflanke setzen.</p> <p>KOP: Das zugewiesene Bit OUT ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am Signalfluss, der in die Spule eintritt, erkannt wird. Der Eingangszustand des Signalflusses durchläuft die Spule immer als Ausgangszustand des Signalflusses. Die Spule P kann an jeder beliebigen Stelle im Netzwerk eingefügt werden.</p> <p>FUP: Das zugewiesene Bit OUT ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am logischen Zustand des Eingangsanschlusses der Box oder am zugewiesenen Eingangsbit erkannt wird, sofern sich die Box am Anfang einer Verzweigung befindet. Der logische Zustand des Eingangs durchläuft die Box immer als logischer Zustand des Ausgangs. Die Box P= kann an jeder beliebigen Stelle in der Verzweigung eingefügt werden.</p>
 <p>"OUT" "M_BIT"</p>	 <p>"OUT" "M_BIT"</p>	Nicht verfügbar ¹	<p>Operand bei negativer Signalflanke setzen.</p> <p>KOP: Das zugewiesene Bit OUT ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am Signalfluss, der in die Spule eintritt, erkannt wird. Der Eingangszustand des Signalflusses durchläuft die Spule immer als Ausgangszustand des Signalflusses. Die Spule N kann an jeder beliebigen Stelle im Netzwerk eingefügt werden.</p> <p>FUP: Das zugewiesene Bit OUT ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am logischen Zustand des Eingangsanschlusses der Box oder am zugewiesenen Eingangsbit erkannt wird, sofern sich die Box am Anfang einer Verzweigung befindet. Der logische Zustand des Eingangs durchläuft die Box immer als logischer Zustand des Ausgangs. Die Box N= kann an jeder beliebigen Stelle in der Verzweigung eingefügt werden.</p>

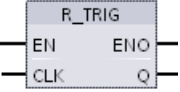
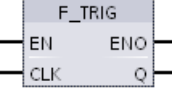
¹ In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 8-15 P_TRIG und N_TRIG

KOP/FUP	SCL	Beschreibung
 <p>"M_BIT"</p>	Nicht verfügbar ¹	<p>VKE (Verknüpfungsergebnis) auf positive Signalflanke abfragen.</p> <p>Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung P_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung P_TRIG an einer beliebigen Stelle angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p>
 <p>"M_BIT"</p>	Nicht verfügbar ¹	<p>VKE auf negative Signalflanke abfragen.</p> <p>Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung N_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung N_TRIG an einer beliebigen Stelle angeordnet werden, Ausnahme: am Ende einer Verzweigung.</p>

¹ In SCL: Sie müssen Code schreiben, um diese Funktion innerhalb Ihrer Anwendung zu replizieren.

Tabelle 8-16 Anweisungen R_TRIG und F_TRIG

KOP/FUP	SCL	Beschreibung
 <p>"R_TRIG_DB"</p>	<pre>"R_TRIG_DB" (CLK:= _in_, Q=> _bool_out_);</pre>	<p>Variable bei positiver Signalflanke setzen.</p> <p>Der vorhergehende Zustand des Eingangs CLK wird im zugewiesenen Instanz-DB gespeichert. Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine steigende Flanke (AUS-nach-EIN) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung R_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung R_TRIG an einer beliebigen Stelle angeordnet werden, ausgenommen am Ende einer Verzweigung.</p>
 <p>"F_TRIG_DB_1"</p>	<pre>"F_TRIG_DB" (CLK:= _in_, Q=> _bool_out_);</pre>	<p>Variable bei negativer Signalflanke setzen.</p> <p>Der vorhergehende Zustand des Eingangs CLK wird im zugewiesenen Instanz-DB gespeichert. Der Signalfluss oder der logische Zustand des Ausgangs Q ist WAHR, wenn eine fallende Flanke (EIN-nach-AUS) am Eingangszustand CLK (FUP) oder am Signalfluss CLK (KOP) erkannt wird.</p> <p>In KOP darf die Anweisung F_TRIG nicht am Anfang oder Ende eines Netzwerks angeordnet werden. In FUP darf die Anweisung F_TRIG an einer beliebigen Stelle angeordnet werden, ausgenommen am Ende einer Verzweigung.</p>

Für R_TRIG und F_TRIG wird automatisch der Dialog "Aufrufoptionen" geöffnet, wenn die Anweisung im Programm eingefügt wird. In diesem Dialog können Sie festlegen, ob der Flanken-Merker in seinem eigenen Datenbaustein gespeichert wird (Einzelinstantz) oder als lokale Variable (Multiinstanz) in der Bausteinschnittstelle gespeichert wird. Wird ein separater Datenbaustein angelegt, so ist dieser in der Projektnavigation im Ordner "Programmressourcen" unter "Programmbausteine > Systembausteine" zu finden.

Tabelle 8-17 Datentypen für die Parameter (Kontakte/Spulen P und N, P=, N=, P_TRIG and N_TRIG)

Parameter	Datentyp	Beschreibung
M_BIT	Bool	Merker, in dem der vorherige Zustand des Eingangs gespeichert wird
IN	Bool	Eingangsbit, dessen Flanke erkannt wird
OUT	Bool	Ausgangsbit, das angibt, dass eine Flanke erkannt wurde
CLK	Bool	Signalfluss oder Eingangsbit, dessen Flanke erkannt wird
Q	Bool	Ausgang, der angibt, dass eine Flanke erkannt wurde

Alle Flankenweisungen verwenden einen Merker (M_BIT: P/N Kontakte/Spulen, P_TRIG/ N_TRIG) oder (Instanz-DB-Bit: R_TRIG, F_TRIG) zur Speicherung des vorhergehenden Zustands des beobachteten Eingangssignals. Eine Flanke wird durch Vergleichen des Zustands des Eingangs mit dem vorhergehenden Zustand erkannt. Wenn die Zustände am Eingang auf einen

Signalwechsel in der gewünschten Richtung hinweisen, wird eine Flanke gemeldet, indem der Ausgang auf WAHR gesetzt wird. Andernfalls wird der Ausgang auf FALSCH gesetzt.

Hinweis

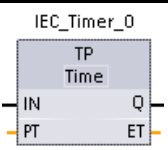
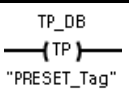
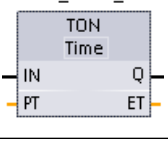
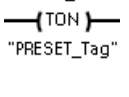
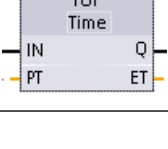
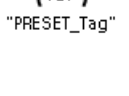
Flankenoperationen werten den Eingang und die Merkerwerte bei jeder Ausführung aus, auch bei der ersten Ausführung. Sie müssen die Ausgangszustände des Eingangs und des Merkers in Ihrem Programm berücksichtigen, um die Flankenerkennung im ersten Zyklus zuzulassen oder nicht.

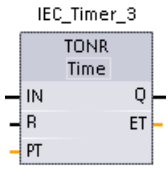
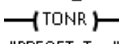
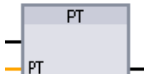
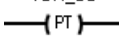
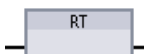
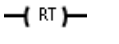
Weil der Merker von einer Ausführung zur nächsten gespeichert werden muss, müssen Sie für jede Flankenoperation ein eindeutiges Bit verwenden. Dieses Bit dürfen Sie nicht an anderen Stellen in Ihrem Programm nutzen. Vermeiden Sie außerdem temporären Speicher und Speicher, der von anderen Systemfunktionen geändert werden kann, z. B. E/A-Aktualisierungen. Verwenden Sie nur Merker (M), globale DBs oder statischen Speicher (in einem Instanz-DB) für M_BIT-Speicherzuweisungen.

8.2 Funktionsweise der Zeiten

Mit den Zeitanweisungen können Sie programmierte Zeitverzögerungen einrichten. Die Anzahl der Zeiten, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Jede Zeit nutzt eine 16 Byte große DB-Struktur vom Datentyp IEC_Timer, um Zeitdaten zu speichern, die im oberen Bereich der Box- oder Spulenanweisung angegeben werden. STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 8-18 Zeitoperationen

KOP/FUP-Boxen	KOP-Spu- len	SCL	Beschreibung
 <p>IEC_Timer_0</p>	 <p>TP_DB "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TP (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Die Zeit TP erzeugt einen Impuls mit einer voreingestellten Dauer.
 <p>IEC_Timer_1</p>	 <p>TON_DB "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TON (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN.
 <p>IEC_Timer_2</p>	 <p>TOF_DB "PRESET_Tag"</p>	<pre>"IEC_Timer_0_DB".TOF (IN:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Die Zeit TOF setzt den Ausgang Q nach einer voreingestellten Zeit auf AUS zurück.

KOP/FUP-Boxen	KOP-Spulen	SCL	Beschreibung
	TONR_DB  "PRESET_Tag"	<pre>"IEC_Timer_0_DB".TONR (IN:=_bool_in_, R:=_bool_in_, PT:=_time_in_, Q=>_bool_out_, ET=>_time_out_);</pre>	Die Zeit TONR setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis Eingang R zum Zurücksetzen der abgelaufenen Zeit angestoßen wird.
Nur FUP: 	TON_DB  "PRESET_Tag"	<pre>PRESET_TIMER (PT:=_time_in_, TIMER:=_iec_timer_in_);</pre>	Die Spule PT (Voreingestellte Zeit) lädt einen neuen PRESET-Zeitwert in den angegebenen IEC_Timer.
Nur FUP: 	TON_DB 	<pre>RESET_TIMER (_iec_timer_in_);</pre>	Die Spule RT (Zeit rücksetzen) setzt den angegebenen IEC_Timer zurück.

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- In den SCL-Beispielen ist "IEC_Timer_0_DB" der Name des Instanz-DBs.

Tabelle 8-19 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Box: IN Spule: Signalfluss	Bool	TP, TON und TONR: Box: 0 = Zeit deaktivieren, 1 = Zeit aktivieren Spule: Kein Signalfluss = Zeit deaktivieren, Signalfluss = Zeit aktivieren TOF: Box: 0 = Zeit aktivieren, 1 = Zeit deaktivieren Spule: Kein Signalfluss = Zeit aktivieren, Signalfluss = Zeit deaktivieren
R	Bool	Nur TONR-Box: 0 = Nicht zurücksetzen 1 = Abgelaufene Zeit und Q-Bit auf 0 zurücksetzen
Box: PT Spule: "PRESET_Tag"	Time	Zeitbox oder -spule: Eingang voreingestellte Zeit
Box: Q Spule: DBdata.Q	Bool	Zeitbox: Q-Boxausgang oder Q-Bit in den DB-Daten der Zeit Zeitspule: Sie können nur das Q-Bit in den DB-Daten der Zeit adressieren
Box: ET Spule: DBdata.ET	Time	Zeitbox: ET-Boxausgang (abgelaufene Zeit) oder ET-Zeitwert in den DB-Daten der Zeit Zeitspule: Sie können nur den ET-Zeitwert in den DB-Daten der Zeit adressieren

Tabelle 8-20 Auswirkung von Wertänderungen in den Parametern PT und IN

Zeitschaltuhr	Änderungen der Box-Parameter PT und IN und der entsprechenden Spulenparameter
TP	<ul style="list-style-type: none"> Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen. Ändert sich IN, während die Zeit läuft, hat dies keine Auswirkungen.
TON	<ul style="list-style-type: none"> Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen. Wenn IN nach FALSCH wechselt, während die Zeit läuft, wird die Zeit angehalten und zurückgesetzt.

Zeitschaltuhr	Änderungen der Box-Parameter PT und IN und der entsprechenden Spulenparameter
TOF	<ul style="list-style-type: none"> • Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen. • Wenn IN nach WAHR wechselt, während die Zeit läuft, wird die Zeit angehalten und zurückgesetzt.
TONR	<ul style="list-style-type: none"> • Ändert sich PT, während die Zeit läuft, hat dies keine Auswirkungen. Es hat dann Auswirkungen, wenn die Zeit fortgesetzt wird. • Wenn IN nach FALSCH wechselt, während die Zeit läuft, wird die Zeit angehalten, jedoch nicht zurückgesetzt. Wenn IN wieder nach WAHR wechselt, beginnt die Zeit ab dem kumulierten Zeitwert zu laufen.

Die Werte von PT (voreingestellte Zeit) und ET (abgelaufene Zeit) werden in den Daten des angegebenen DBs IEC_TIMER als vorzeichenbehaftete doppelte Ganzzahlen gespeichert, die einen Zeitwert in Millisekunden darstellen. Der Datentyp TIME verwendet die Kennung T# und kann als einfache Zeiteinheit (T#200ms oder 200) oder als zusammengesetzte Zeiteinheiten wie T#2s_200ms eingegeben werden.

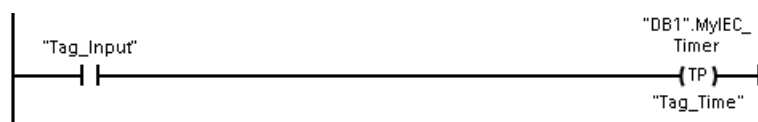
Tabelle 8-21 Größe und Bereich des Datentyps TIME

Datentyp	Größe	Gültige Zahlenbereiche ¹
TIME	32 Bit, gespeichert als DInt-Daten	T#-24d_20h_31m_23s_648ms bis T#24d_20h_31m_23s_647ms Gespeichert als -2.147.483.648 ms bis +2.147.483.647 ms

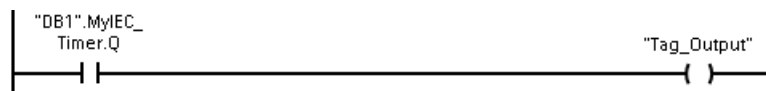
¹ Der negative Bereich des oben dargestellten Datentyps TIME kann für die Zeiten nicht verwendet werden. Negative Werte für PT (voreingestellte Zeit) werden bei Ausführung der Zeitoperation auf Null gesetzt. ET (abgelaufene Zeit) ist immer ein positiver Wert.

Beispiel für eine Zeitspule

Die Zeitspulen -(TP)-, -(TON)-, -(TOF)- und -(TONR)- müssen als letzte Anweisung in einem KOP-Netzwerk angeordnet werden. Wie im Beispiel für eine Zeit gezeigt, wertet eine Kontaktanweisung in einem nachfolgenden Netzwerk das Q-Bit in den Daten des DBs IEC_Timer einer Zeitspule aus. Ebenso müssen Sie das Element ELAPSED in den Daten des DBs IEC_Timer adressieren, wenn Sie den Wert der abgelaufenen Zeit in Ihrem Programm verwenden möchten.



Der Impulszeitgeber wird bei einem Wechsel von 0 nach 1 des Bitwerts von Tag_Input gestartet. Die Zeit läuft für die vom Zeitwert Tag_Time angegebene Zeitdauer.



Solange die Zeit läuft, sind der Zustand von DB1.MyIEC_Timer.Q = 1 und von Tag_Output = 1. Wenn der Wert von Tag_Time abgelaufen ist, sind DB1.MyIEC_Timer.Q = 0 und Tag_Output = 0.

Spulen Zeit rücksetzen -(RT)- und Zeitdauer laden -(PT)-

Diese Spulenanweisungen können mit Box- oder Spulenzeiten verwendet werden und in der Mitte eines Strompfads angeordnet werden. Der Signalzustand des Spulenausgangs ist immer der gleiche wie am Spuleneingang. Wenn die Spule -(RT)- aktiviert wird, wird das Zeitelement ELAPSED in den Daten des angegebenen DB IEC_Timer auf 0 zurückgesetzt. Wenn die Spule -(PT)- aktiviert wird, wird das Zeitelement PRESET in den Daten des angegebenen DB IEC_Timer mit dem zugeordneten Zeitdauerwert geladen.

Hinweis

Wenn Sie Zeiten in einem FB platzieren, können Sie die Option "Multiinstanz-Datenbaustein" auswählen. Die Namen der Zeitstrukturen können bei verschiedenen Datenstrukturen unterschiedlich sein, doch die Zeitdaten befinden sich in einem einzigen Datenbaustein, und es ist nicht für jede Zeit ein eigener Datenbaustein erforderlich. Dadurch verringert sich die Verarbeitungszeit und der für die Verwaltung der Zeiten erforderliche Datenspeicher. Zwischen den Datenstrukturen der Zeiten im gemeinsam genutzten Multiinstanz-DB gibt es keine Wechselwirkungen.

Funktionsweise der Zeiten

Tabelle 8-22 Typen von IEC-Zeiten

Zeit	Zeitdiagramm
<p>TP: Impuls generieren</p> <p>Die Zeit TP erzeugt einen Impuls mit einer voreingestellten Dauer.</p>	
<p>TON: Einschaltverzögerung generieren</p> <p>Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN.</p>	

Zeit	Zeitdiagramm
<p>TOF: Ausschaltverzögerung generieren</p> <p>Die Zeit TON setzt den Ausgang Q nach einer voreingestellten Zeit auf AUS zurück.</p>	
<p>TONR: Zeitakkumulator</p> <p>Die Zeit TONR setzt den Ausgang Q nach einer voreingestellten Zeit auf EIN. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis Eingang R zum Zurücksetzen der abgelaufenen Zeit angestoßen wird.</p>	

Hinweis

In der CPU ist keiner spezifischen Zeit eine bestimmte Ressource zugeordnet. Stattdessen nutzt jede Zeit eine eigene Zeitstruktur im DB-Speicher und eine fortlaufend in Betrieb befindliche interne CPU-Zeit, um die Zeitsteuerung durchzuführen.

Wenn eine Zeit aufgrund eines Flankenwechsels am Eingang einer Anweisung TP, TON, TOF oder TONR gestartet wird, wird der Wert der fortlaufend in Betrieb befindlichen internen CPU-Zeit in das START-Element der DB-Struktur kopiert, die dieser Zeit zugeordnet ist. Dieser Startwert bleibt solange unverändert, wie die Zeit läuft, und wird später bei jeder Aktualisierung der Zeit verwendet. Jedesmal, wenn die Zeit gestartet wird, wird ein neuer Startwert aus der internen CPU-Zeit in die Zeitstruktur geladen.

Wenn eine Zeit aktualisiert wird, wird der oben beschriebene Startwert vom aktuellen Wert der internen CPU-Zeit subtrahiert, um die abgelaufene Zeit zu ermitteln. Die abgelaufene Zeit wird dann mit der Voreinstellung verglichen, um den Zustand des Zeitbits Q zu ermitteln. Die Elemente ELAPSED und Q werden dann in der DB-Struktur, die dieser Zeit zugeordnet ist, aktualisiert. Beachten Sie, dass die abgelaufene Zeit an den voreingestellten Wert geknüpft ist (die Zeit akkumuliert die abgelaufene Zeit nicht weiter, wenn die Voreinstellung erreicht ist).

Die Zeit wird nur dann aktualisiert, wenn:

- Eine Zeitanweisung (TP, TON, TOF oder TONR) ausgeführt wird
- Das Element "ELAPSED" der Zeitstruktur im DB direkt von einer Anweisung referenziert wird
- Das Element "Q" der Zeitstruktur im DB direkt von einer Anweisung referenziert wird

Programmierung von Zeiten

Die folgenden Konsequenzen von Zeiten sind bei der Planung und Erstellung Ihres Anwenderprogramms zu berücksichtigen:

- Sie können mehrere Aktualisierungen einer Zeit im gleichen Zyklus haben. Die Zeit wird bei jeder Ausführung der Zeitanweisung (TP, TON, TOF, TONR) aktualisiert sowie jedesmal, wenn das Element ELAPSED oder Q der Zeitstruktur als Parameter einer anderen ausgeführten Anweisung verwendet wird. Dies ist ein Vorteil, wenn Sie die neuesten Zeitdaten benötigen (praktisch ein direktes Auslesen der Zeit). Wenn Sie jedoch während eines Programmzyklus konsistente Werte nutzen möchten, ordnen Sie Ihre Zeitanweisung vor allen anderen Anweisungen, die diese Werte benötigen, an und verwenden statt der Elemente ELAPSED und Q der Zeitstruktur im DB die Variablen der Ausgänge Q und ET der Zeitanweisung.
- Sie können Zyklen ohne Zeitaktualisierung haben. Es ist möglich, Ihre Zeit in einer Funktion zu starten und diese Funktion dann für einen oder weitere Zyklen nicht mehr aufzurufen. Wenn keine anderen Anweisungen ausgeführt werden, die die Elemente ELAPSED oder Q der Zeitstruktur referenzieren, wird die Zeit nicht mehr aktualisiert. Eine neue Aktualisierung tritt erst auf, wenn entweder die Zeitanweisung erneut ausgeführt wird oder eine andere Anweisung ausgeführt wird, die ELAPSED oder Q aus der Zeitstruktur als Parameter nutzt.
- Es ist zwar nicht gerade üblich, aber Sie können mehreren Zeitanweisungen die gleiche DB-Zeitstruktur zuweisen. In Allgemeinen sollten Sie, um unerwartete Wechselwirkungen zu vermeiden, nur eine Zeit (TP, TON, TOF, TONR) pro DB-Zeitstruktur verwenden.
- Selbstrücksetzende Zeiten sind bei Trigger-Aktionen nützlich, die regelmäßig auftreten sollen. Typischerweise werden selbstrücksetzende Zeiten erstellt, indem ein Öffnerkontakt, der das Zeitbit referenziert, vor der Zeitanweisung angeordnet wird. Dieses Zeitnetzwerk befindet sich typischerweise oberhalb eines oder mehrerer abhängiger Netzwerke, die mit dem Zeitbit Aktionen auslösen. Wenn die Zeit abläuft (die abgelaufene Zeit erreicht den voreingestellten Wert), ist das Zeitbit einen Zyklus lang EIN, so dass die Logik der vom Zeitbit gesteuerten abhängigen Netzwerke ausgeführt werden kann. Bei der nächsten Ausführung des Zeitnetzwerks ist der Öffnerkontakt AUS, wodurch die Zeit zurückgesetzt und das Zeitbit gelöscht wird. Im nächsten Zyklus ist der Öffnerkontakt EIN, weshalb die Zeit neu gestartet wird. Beim Erstellen von selbstrücksetzenden Zeiten wie dieser verwenden Sie das Element "Q" der Zeitstruktur im DB nicht als den Parameter für den Öffnerkontakt vor der Zeitanweisung. Verwenden Sie hierfür stattdessen die an den Ausgang "Q" der Zeitanweisung angeschlossene Variable. Der Grund, weshalb vermieden werden sollte, auf das Element Q der DB-Zeitstruktur zuzugreifen, ist der, dass dadurch eine Aktualisierung der Zeit verursacht wird. Und wenn die Zeit durch den Öffnerkontakt aktualisiert wird, setzt der Kontakt die Zeitanweisung sofort zurück. Der Ausgang Q der Zeitanweisung ist während dieses einen Zyklus nicht EIN, und die abhängigen Netzwerke werden nicht ausgeführt.

Speichern von Zeitdaten nach einem RUN-STOP-RUN-Wechsel oder einem Neustart der CPU

Wenn eine Session im Betriebszustand RUN mit dem Betriebszustand STOP oder einem Neustart der beendet wird und eine neue Session im Betriebszustand RUN gestartet wird, gehen die in der vorherigen RUN-Session gespeicherten Zeitdaten verloren, sofern die Zeitdatenstruktur nicht als remanent definiert ist (Zeiten TP, TON, TOF und TONR).

Wenn Sie beim Einfügen einer Zeitanweisung im Programmiereditor im Dialog der Aufrufoptionen die Standardeinstellung übernehmen, wird automatisch ein Instanz-DB zugewiesen, der **nicht als remanent definiert werden kann**. Um Ihre Zeitdaten als remanent zu definieren, müssen Sie entweder einen globalen DB oder einen Multiinstanz-DB nutzen.

Zeitdaten durch Zuweisen eines globalen DB als remanente Daten speichern

Diese Option funktioniert unabhängig davon, wo die Zeit platziert wird (OB, FC oder FB).

1. Globalen DB erstellen:
 - Doppelklicken Sie in der Projektnavigation auf "Neuen Baustein hinzufügen".
 - Klicken Sie auf das Symbol für Datenbaustein (DB).
 - Wählen Sie globalen DB als Typ.
 - Wenn Sie einzelne Datenelemente im DB als remanent definieren möchten, müssen Sie für den DB-Typ das Kontrollkästchen "Optimiert" aktivieren. Die andere Option für den DB-Typ "Standard - kompatibel mit S7-300/400" gestattet es lediglich, entweder alle Datenelemente des DBs oder keine Datenelemente als remanent zu definieren.
 - Klicken Sie auf "OK".
2. Zeitstruktur(en) zum DB hinzufügen:
 - Fügen Sie im neuen globalen DB eine neue statische Variable vom Datentyp IEC_Timer ein.
 - Aktivieren Sie das Kontrollkästchen in der Spalte "Remanent", damit diese Struktur remanent ist.
 - Wiederholen Sie diesen Vorgang, um für alle Zeiten, die Sie in diesem DB speichern möchten, Strukturen anzulegen. Sie können entweder jede Zeitstruktur in einem eindeutigen globalen DB anordnen, oder Sie können mehrere Zeitstrukturen in demselben globalen DB platzieren. Sie können in diesem globalen DB außer Zeiten auch andere statische Variablen anordnen. Durch Platzieren mehrerer Zeitstrukturen in demselben globalen DB können Sie die Gesamtzahl der Bausteine verringern.
 - Benennen Sie die Zeitstrukturen ggf. um.
3. Öffnen Sie den Programmbaustein (OB, FC oder FB), in den Sie eine remanente Zeit einfügen möchten, für die Bearbeitung.
4. Platzieren Sie die Zeitanweisung an der gewünschten Stelle.
5. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf "Abbrechen".
6. Geben Sie oben in der neuen Zeitanweisung den Namen des globalen DB und der Zeitstruktur ein, die Sie oben erstellt haben (wählen Sie keinen Namen aus). Beispiel: "Data_block_3.Static_1".

Zeitdaten durch Zuweisen eines Multiinstanz-DB als remanente Daten speichern

Diese Möglichkeit funktioniert nur, wenn Sie die Zeit in einem FB ablegen.

Diese Variante ist davon abhängig, ob in den FB-Eigenschaften "Optimierter Bausteinzugriff" festgelegt ist (nur die symbolische Adressierung ist möglich). Um zu prüfen, wie das Zugriffsattribut eines vorhandenen FB konfiguriert ist, klicken Sie mit der rechten Maustaste in der Projektnavigation auf den FB, wählen "Eigenschaften" und dann "Attribute".

Wenn für den FB "Optimierter Bausteinzugriff" festgelegt ist (nur die symbolische Adressierung ist möglich):

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zeitanweisung an der gewünschten Stelle im FB.

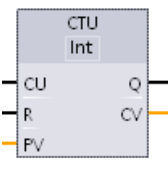
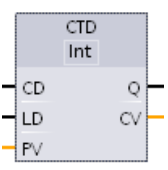
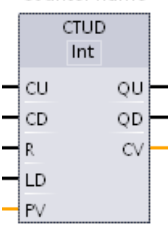
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie der Zeit im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zeitanweisung erscheint im Editor und die IEC_TIMER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Öffnen Sie ggf. den FB-Schnittstelleneditor (Sie müssen möglicherweise auf den kleinen Pfeil klicken, um die Ansicht zu vergrößern).
7. Suchen Sie unter "Statisch" die Zeitstruktur, die gerade für Sie angelegt wurde.
8. Ändern Sie in der Spalte "Remanent" dieser Zeitstruktur die Auswahl in "Remanent". Immer wenn dieser FB später in einem anderen Programmbaustein aufgerufen wird, wird ein Instanz-DB mit dieser Schnittstellendefinition angelegt, bei der die Zeitstruktur als remanent definiert ist.

Ist die Option "Optimierter Bausteinzugriff" für den FB nicht angegeben, wird der Standardbausteinzugriff verwendet, der mit klassischen S7-300/400 Konfigurationen kompatibel ist und symbolischen sowie direkten Zugriff gestattet. Um einem FB mit Standardbausteinzugriff eine Multiinstanz zuzuweisen, gehen Sie wie folgt vor:

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zeitanweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie der Zeit im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zeitanweisung erscheint im Editor und die IEC_TIMER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Öffnen Sie den Baustein, der diesen FB verwenden soll.
7. Platzieren Sie diesen FB an der gewünschten Stelle. Dadurch wird ein Instanz-Datenbaustein für diesen FB angelegt.
8. Öffnen Sie den Instanz-Datenbaustein, der angelegt wurde, als Sie den FB im Editor platziert haben.
9. Suchen Sie unter "Statisch" die gewünschte Zeitstruktur. Aktivieren Sie in der Spalte "Remanent" dieser Zeitstruktur das Kontrollkästchen, um diese Struktur als remanent zu definieren.

8.3 Funktionsweise der Zähler

Tabelle 8-23 Zähleranweisungen

KOP/FUP	SCL	Beschreibung
<p>"Counter name"</p> 	<pre>"IEC_Counter_0_DB".CTU(CU:=_bool_in, R:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	<p>Mit den Zähleranweisungen können Sie programminterne Ereignisse und externe Prozessereignisse zählen. Jeder Zähler nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten des Zählers zu speichern. Sie weisen den Datenbaustein zu, wenn Sie die Zähloperation im Editor einfügen.</p> <ul style="list-style-type: none"> • CTU zählt vorwärts. • CTD zählt rückwärts. • CTUD zählt vorwärts und rückwärts.
<p>"Counter name"</p> 	<pre>"IEC_Counter_0_DB".CTD(CD:=_bool_in, LD:=_bool_in, PV:=_in, Q=>_bool_out, CV=>_out);</pre>	
<p>"Counter name"</p> 	<pre>"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);</pre>	

1 In KOP und FUP: Wählen Sie den Datentyp für den Zählwert aus der Klappliste unterhalb des Anweisungsnamens aus.

2 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

3 In den SCL-Beispielen ist "IEC_Counter_0_DB" der Name des Instanz-DBs.

Tabelle 8-24 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
CU, CD	Bool	Aufwärts- oder Abwärtszählen um jeweils eine Einheit
R (CTU, CTUD)	Bool	Zählwert auf Null zurücksetzen
LD (CTD, CTUD)	Bool	Ladesteuerung für den voreingestellten Wert
PV	SInt, Int, DInt, USInt, UInt, UDInt	Voreingestellter Zählwert
Q, QU	Bool	Wahr, wenn CV >= PV
QD	Bool	Wahr, wenn CV <= 0
CV	SInt, Int, DInt, USInt, UInt, UDInt	Aktueller Zählwert

1 Der numerische Bereich der Zählwerte hängt vom ausgewählten Datentyp ab. Ist der Zählwert ein ganzzahliger Wert ohne Vorzeichen, können Sie bis Null herunter- oder bis zur Bereichsgrenze hochzählen. Ist der Zählwert ein ganzzahliger Wert mit Vorzeichen, können Sie bis zum unteren Grenzwert herunter- und bis zum oberen Grenzwert hochzählen.

Die Anzahl der Zähler, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Zähler benötigen den folgenden Speicherplatz:

- Bei den Datentypen SInt oder USInt benötigt die Zähleranweisung 3 Byte.
- Bei den Datentypen Int oder UInt benötigt die Zähleranweisung 6 Byte.
- Bei den Datentypen DInt oder UDInt benötigt die Zähleranweisung 12 Byte.

Diese Anweisungen nutzen Softwarezähler, deren maximale Zählgeschwindigkeit durch die Ausführungsrate des OBs, in den sie eingefügt wurden, begrenzt ist. Der OB, in den die Anweisungen eingefügt werden, muss häufig genug ausgeführt werden, um alle Transitionen der Eingänge CU oder CD zu erkennen. Um schneller zu zählen, nutzen Sie die Anweisung CTRL_HSC (Seite 543).

Hinweis

Wenn Sie Zähleranweisungen in einem FB platzieren, können Sie die Option "Multiinstanz-DB" auswählen. Die Namen der Zählerstrukturen können bei verschiedenen Datenstrukturen unterschiedlich sein, doch die Zählerdaten befinden sich in einem einzigen DB, und es ist nicht für jeden Zähler ein eigener DB erforderlich. Dadurch verringert sich die Verarbeitungszeit und der für die Zähler erforderliche Datenspeicher. Zwischen den Datenstrukturen der Zähler im gemeinsam genutzten Multiinstanz-DB gibt es keine Wechselwirkungen.

Funktionsweise der Zähler

Tabelle 8-25 Funktionsweise von CTU (Vorwärtszählen)

Zähler	Bedienung
<p>Der Zähler CTU zählt um 1 vorwärts, wenn der Wert des Parameters CU von 0 nach 1 wechselt. Das CTU-Zeitdiagramm zeigt die Funktionsweise bei einem vorzeichenlosen ganzzahligen Zählwert (dabei ist PV = 3).</p> <ul style="list-style-type: none"> • Ist der Wert des Parameters CV (aktueller Zählwert) größer oder gleich dem Wert des Parameters PV (voreingestellter Zählwert), dann lautet der Parameter für den Zählerausgang Q = 1. • Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt. 	<p>The diagram shows four signals over time: CU (counting input), R (reset input), CV (current count), and Q (output). CU is a periodic square wave. R is a single pulse. CV starts at 0 and increases by 1 on each rising edge of CU. When R is active, CV resets to 0. The output Q is a pulse that occurs when CV reaches the preset value PV (3) and CU is active. The diagram is divided into four numbered sections (1-4) by vertical dashed lines.</p>

Tabelle 8-26 Funktionsweise von CTD (Rückwärtszählen)

Zähler	Bedienung
<p>Der Zähler CTD zählt um 1 rückwärts, wenn der Wert des Parameters CD von 0 nach 1 wechselt. Das CTD-Zeitdiagramm zeigt die Funktionsweise bei einem vorzeichenlosen ganzzahligen Zählwert (dabei ist PV = 3).</p> <ul style="list-style-type: none"> Ist der Wert des Parameters CV (aktueller Zählwert) kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang Q = 1. Wechselt der Wert von Parameter LOAD von 0 nach 1, wird der Wert an Parameter PV (voreingestellter Wert) als neuer CV (aktueller Zählwert) in den Zähler geladen. 	

Tabelle 8-27 Funktionsweise von CTUD (Vorwärts- und Rückwärtszählen)

Zähler	Bedienung
<p>Der Zähler CTUD zählt um 1 vorwärts oder rückwärts, wenn der Vorwärtszähleingang (CU) oder der Rückwärtszähleingang (CD) von 0 nach 1 wechselt. Das CTUD-Zeitdiagramm zeigt die Funktionsweise bei ganzzahligem Zählwert ohne Vorzeichen (und PV = 4).</p> <ul style="list-style-type: none"> Ist der Wert des Parameters CV größer oder gleich dem Wert des Parameters PV, dann lautet der Parameter für den Zählerausgang QU = 1. Ist der Wert des Parameters CV kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang QD = 1. Wechselt der Wert von Parameter LOAD von 0 nach 1, wird der Wert an Parameter PV als neuer CV in den Zähler geladen. Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt. 	

Speichern von Zählerdaten nach einem RUN-STOP-RUN-Wechsel oder einem Neustart der CPU

Wenn eine Session im Betriebszustand RUN mit dem Betriebszustand STOP oder einem Neustart der CPU beendet wird und eine neue Session im Betriebszustand RUN gestartet wird, gehen die in der vorherigen RUN-Session gespeicherten Zählerdaten verloren, sofern die Zählerdatenstruktur nicht als remanent definiert ist (Zeiten CTU, CTD und CTUD).

Wenn Sie beim Einfügen einer Zähleranweisung im Programmiereditor im Dialog der Aufrufoptionen die Standardeinstellung übernehmen, wird automatisch ein Instanz-DB zugewiesen, der **nicht als remanent definiert werden kann**. Um Ihre Zählerdaten als remanent zu definieren, müssen Sie entweder einen globalen DB oder einen Multiinstanz-DB nutzen.

Zählerdaten durch Zuweisen eines globalen DBs als remanente Daten speichern

Diese Option funktioniert unabhängig davon, wo der Zähler platziert wird (OB, FC oder FB).

1. Globalen DB erstellen:
 - Doppelklicken Sie in der Projektnavigation auf "Neuen Baustein hinzufügen".
 - Klicken Sie auf das Symbol für Datenbaustein (DB).
 - Wählen Sie globalen DB als Typ.
 - Wenn Sie einzelne Elemente im DB als remanent definieren möchten, müssen Sie das Kontrollkästchen "Nur symbolisch adressierbar" aktivieren.
 - Klicken Sie auf "OK".
2. Zählerstruktur(en) zum DB hinzufügen:
 - Fügen Sie im neuen globalen DB eine neue statische Variable vom Datentyp Zähler ein. Berücksichtigen Sie dabei den Typ für die Voreinstellung und den Zählerwert.
 - Aktivieren Sie das Kontrollkästchen in der Spalte "Remanent", damit diese Struktur remanent ist.
 - Wiederholen Sie diesen Vorgang, um für alle Zähler, die Sie in diesem DB speichern möchten, Strukturen anzulegen. Sie können entweder jede Zählerstruktur in einem eindeutigen globalen DB anordnen, oder Sie können mehrere Zählerstrukturen in demselben globalen DB platzieren. Sie können in diesem globalen DB außer Zählern auch andere statische Variablen anordnen. Durch Platzieren mehrerer Zählerstrukturen in demselben globalen DB können Sie die Gesamtzahl der Bausteine verringern.
 - Benennen Sie die Zählerstrukturen ggf. um.
3. Öffnen Sie den Programmbaustein (OB, FC oder FB), in den Sie einen remanenten Zähler einfügen möchten, für die Bearbeitung.
4. Platzieren Sie die Zähleranweisung an der gewünschten Stelle.
5. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf "Abbrechen". Nun sollte die neue Zähleranweisung mit dem Platzhalter "???" oberhalb und unterhalb des Anweisungsnamens angezeigt werden.
6. Geben Sie oben in der neuen Zähleranweisung den Namen des globalen DBs und der Zählerstruktur ein, die Sie oben erstellt haben (wählen Sie keinen Namen aus). Beispiel: "Data_block_3.Static_1". Daraufhin wird der entsprechende Typ für die Voreinstellung und den Zählerwert vorgegeben (Beispiel: UInt für eine IEC_UCounter-Struktur).

Zählerdatentyp	Entsprechender Typ für die Voreinstellung und den Zählerwert
IEC_Counter	INT
IEC_SCounter	SINT
IEC_DCounter	DINT
IEC_UCounter	UINT
IEC_USCounter	USINT
IEC_UDCounter	UDINT

Zählerdaten durch Zuweisen eines Multiinstanz-DB als remanente Daten speichern

Diese Möglichkeit funktioniert nur, wenn Sie den Zähler in einem FB ablegen.

Diese Variante ist davon abhängig, ob in den FB-Eigenschaften "Optimierter Bausteinzugriff" festgelegt ist (nur die symbolische Adressierung ist möglich). Um zu prüfen, wie das Zugriffsattribut eines vorhandenen FB konfiguriert ist, klicken Sie mit der rechten Maustaste in der Projektnavigation auf den FB, wählen "Eigenschaften" und dann "Attribute".

Wenn für den FB "Optimierter Bausteinzugriff" festgelegt ist (nur die symbolische Adressierung ist möglich):

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zähleranweisung an der gewünschten Stelle im FB.
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie dem Zähler im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zähleranweisung erscheint im Editor mit dem Typ INT für die Voreinstellung und den Zählwert, und die IEC_COUNTER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Ändern Sie ggf. den Typ in der Zähleranweisung von INT in einen anderen Typ. Die Zählerstruktur wird entsprechend verändert.
7. Öffnen Sie ggf. den FB-Schnittstelleneditor (Sie müssen möglicherweise auf den kleinen Pfeil klicken, um die Ansicht zu vergrößern).
8. Suchen Sie unter "Statisch" die Zählerstruktur, die gerade für Sie angelegt wurde.
9. Ändern Sie in der Spalte "Remanent" dieser Zählerstruktur die Auswahl in "Remanent". Immer wenn dieser FB später in einem anderen Programmbaustein aufgerufen wird, wird ein Instanz-DB mit dieser Schnittstellendefinition angelegt, bei der die Zählerstruktur als remanent definiert ist.

Ist die Option "Optimierter Bausteinzugriff" für den FB nicht angegeben, wird der Standardbausteinzugriff verwendet, der mit klassischen S7-300/400 Konfigurationen kompatibel ist und symbolischen sowie direkten Zugriff gestattet. Um einem FB mit Standardbausteinzugriff eine Multiinstanz zuzuweisen, gehen Sie wie folgt vor:

1. Öffnen Sie den FB zum Bearbeiten.
2. Platzieren Sie die Zähleranweisung an der gewünschten Stelle im FB.

8.4 Funktionsweise von Vergleichen


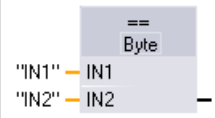
3. Wenn der Dialog mit den Aufrufoptionen angezeigt wird, klicken Sie auf das Multiinstanz-Symbol. Die Multiinstanz-Option ist nur verfügbar, wenn die Anweisung in einen FB eingefügt wird.
4. Geben Sie dem Zähler im Dialog der Aufrufoptionen ggf. einen Namen.
5. Klicken Sie auf "OK". Die Zähleranweisung erscheint im Editor mit dem Typ INT für die Voreinstellung und den Zählwert, und die IEC_COUNTER-Struktur erscheint in der FB-Schnittstelle unter "Statisch".
6. Ändern Sie ggf. den Typ in der Zähleranweisung von INT in einen anderen Typ. Die Zählerstruktur wird entsprechend verändert.
7. Öffnen Sie den Baustein, der diesen FB verwenden soll.
8. Platzieren Sie diesen FB an der gewünschten Stelle. Dadurch wird ein Instanz-Datenbaustein für diesen FB angelegt.
9. Öffnen Sie den Instanz-Datenbaustein, der angelegt wurde, als Sie den FB im Editor platziert haben.
10. Suchen Sie unter "Statisch" die gewünschte Zählerstruktur. Aktivieren Sie in der Spalte "Remanent" dieser Zählerstruktur das Kontrollkästchen, um diese Struktur als remanent zu definieren.

Typ in Zähleranweisung (für Voreinstellung und Zählwert)	Entsprechender Struktur-Typ in FB-Schnittstelle
INT	IEC_Counter
SINT	IEC_SCounter
DINT	IEC_DCounter
UINT	IEC_UCounter
USINT	IEC_USCounter
UDINT	IEC_UDCounter

8.4 Funktionsweise von Vergleichen

8.4.1 Vergleichsoperationen

Tabelle 8-28 Vergleichsoperationen

KOP	FUP	SCL	Beschreibung
 <p>"IN1" == Byte "IN2"</p>	 <p>== Byte "IN1" — IN1 "IN2" — IN2</p>	<pre>out := in1 = in2; or IF in1 = in2 THEN out := 1; ELSE out := 0; END_IF;</pre>	<p>Vergleicht zwei Werte desselben Datentyps. Hat der KOP-Kontaktvergleich das Ergebnis WAHR, wird der Kontakt aktiviert. Ist das Ergebnis des FUP-Boxvergleichs WAHR, ist der Box-Ausgang ebenfalls WAHR.</p>

¹ In KOP und FUP: Klicken Sie auf den Anweisungsnamen (wie "=="), um den Vergleichstyp über die Klappliste zu ändern. Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-29 Datentypen für die Parameter

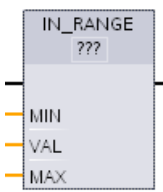
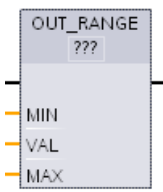
Parameter	Datentyp	Beschreibung
IN1, IN2	Byte, Word, DWord, SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, String, WString, Char, Char, Time, Date, TOD, DTL, Konstante	Zu vergleichende Werte

Tabelle 8-30 Beschreibungen der Vergleiche

Beziehungstyp	Der Vergleich ist wahr, wenn...
=	IN1 gleich IN2 ist
<>	IN1 nicht gleich IN2 ist
>=	IN1 größer oder gleich IN2 ist
<=	IN1 kleiner oder gleich IN2 ist
>	IN1 größer als IN2 ist
<	IN1 kleiner als IN2 ist

8.4.2 IN_Range (Wert innerhalb Bereich) und OUT_Range (Wert außerhalb Bereich)

Tabelle 8-31 Anweisungen Wert innerhalb eines Bereichs und Wert außerhalb eines Bereichs

KOP/FUP	SCL	Beschreibung
	<pre>out := IN_RANGE (min, val, max);</pre>	Prüft, ob ein Eingabewert innerhalb oder außerhalb eines angegebenen Wertebereichs liegt. Ist der Vergleich WAHR, ist der Box-Ausgang WAHR.
	<pre>out := OUT_RANGE (min, val, max);</pre>	

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-32 Datentypen für die Parameter

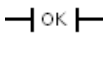
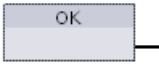
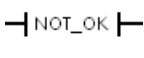
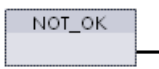
Parameter	Datentyp ¹	Beschreibung
MIN, VAL, MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Vergleichereingänge

¹ Die Eingangsparameter MIN, VAL und MAX müssen denselben Datentyp haben.

- Der Vergleich IN_RANGE ist wahr, wenn: MIN <= VAL <= MAX
- Der Vergleich OUT_RANGE ist wahr, wenn: VAL < MIN oder VAL > MAX

8.4.3 OK (Gültigkeit prüfen) und NOT_OK (Ungültigkeit prüfen)

Tabelle 8-33 Anweisungen OK (Gültigkeit prüfen) und NOT OK (Ungültigkeit prüfen)

KOP	FUP	SCL	Beschreibung
		Nicht verfügbar	Prüft, ob eine Eingabedatenreferenz eine gültige Realzahl nach der IEEE-Spezifikation 754 ist.
		Nicht verfügbar	

¹ In KOP und FUP: Ist der KOP-Kontakt WAHR, wird der Kontakt aktiviert und leitet Signalfloss. Ist die FUP-Box WAHR, ist der Box-Ausgang WAHR.

Tabelle 8-34 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingangsdaten

Tabelle 8-35 Funktionsweise

Anweisung	Die Prüfung auf eine reale Zahl ist WAHR, wenn:
OK	Der Eingabewert ist eine gültige Realzahl ¹ .
NOT_OK	Der Eingabewert ist keine gültige Realzahl ¹ .

¹ Ein Wert Real oder LReal ist ungültig, wenn er +/- INF (unendlich), NaN (Not a Number, keine Zahl) oder ein nicht normierter Wert ist. Ein nicht normierter Wert ist eine Zahl sehr nah bei Null. In Berechnungen ersetzt die CPU einen nicht normierten Wert durch Null.

8.4.4 Variant- und Array-Vergleichsoperationen

8.4.4.1 Gleich- und Ungleich-Vergleichsoperationen

Die S7-1200 CPU bietet Anweisungen für die Abfrage des Datentyps einer Variablen, auf die ein Variant-Operand zeigt und für den Vergleich, ob diese mit dem Datentyp des anderen Operanden übereinstimmt (gleich) oder nicht (ungleich).

Darüber hinaus bietet die S7-1200 CPU Anweisungen für die Abfrage des Datentyps eines Array-Elements auf Gleichheit oder Ungleichheit mit dem Datentyp des anderen Operanden.

Bei diesen Operationen wird <Operand1> mit <Operand2> verglichen. <Operand1> muss Datentyp Variant aufweisen. <Operand2> kann ein elementarer Datentyp eines PLC-Datentyps sein. In KOP und FUP ist <Operand1> der Operand über der Anweisung. In KOP ist <Operand2> der Operand unter der Anweisung.

Für alle Anweisungen ist das Verknüpfungsergebnis (VKE) 1 (wahr), wenn der Gleich- oder Ungleich-Vergleich erfolgreich war, andernfalls ist es 0 (falsch).

Die Gleich- und Ungleich-Vergleichsanweisungen sind nachfolgend beschrieben:

- EQ_Type (Datentyp mit dem Datentyp einer Variablen auf GLEICH vergleichen)
- NE_Type (Datentyp mit dem Datentyp einer Variablen auf UNGLEICH vergleichen)
- EQ_ElemType (Datentyp eines ARRAY-Elements mit dem Datentyp einer Variablen auf GLEICH vergleichen)
- NE_ElemType (Datentyp eines ARRAY-Elements mit dem Datentyp einer Variablen auf UNGLEICH vergleichen)

Tabelle 8-36 Anweisungen EQ und NE

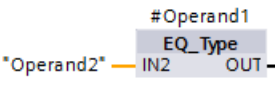
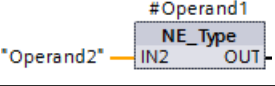
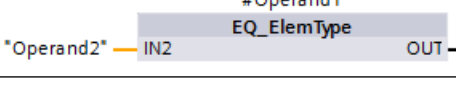
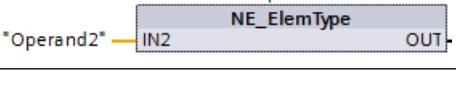
KOP	FUP	SCL	Beschreibung
#Operand1 EQ_Type "Operand2"		Nicht verfügbar	Prüft, ob die Variable, auf die in Operand1 gezeigt wird, den gleichen Datentyp wie die Variable an Operand2 hat.
#Operand1 NE_Type "Operand2"		Nicht verfügbar	Prüft, ob die Variable, auf die in Operand1 gezeigt wird, einen anderen Datentyp als die Variable an Operand2 hat.
#Operand1 EQ_ElemType "Operand2"		Nicht verfügbar	Prüft, ob das Array-Element, auf das in Operand1 gezeigt wird, den gleichen Datentyp wie die Variable an Operand2 hat.
#Operand1 NE_ElemType "Operand2"		Nicht verfügbar	Prüft, ob das Array-Element, auf das in Operand1 gezeigt wird, einen anderen Datentyp als die Variable an Operand2 hat.

Tabelle 8-37 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Operand1	Variant	Erster Operand
Operand2	Datentypen Bitzeichenfolgen, Ganzzahlen, Gleitpunktzahlen, Zeiten, Datum und Uhrzeit, Zeichenfolgen, ARRAY, PLC	Zweiter Operand

8.4.4.2 Null-Vergleichsoperationen

Die Anweisungen IS_NULL und NOT_NULL können verwendet werden, um festzustellen, ob der Eingang auf ein Objekt zeigt oder nicht.

Für beide Anweisungen muss <Operand> den Datentyp Variant haben.

Tabelle 8-38 IS_NULL (Abfrage nach Pointer GLEICH NULL) und NOT_NULL (Abfrage nach Pointer UNGLEICH NULL)

KOP	FUP	SCL	Beschreibung
#Operand ┌IS_NULL┐	#Operand IS_NULL OUT-	Nicht verfügbar	Prüft, ob die Variable, auf die das Element Variant an Operand zeigt, Null und somit kein Objekt ist.
#Operand ┌NOT_NULL┐	#Operand NOT_NULL OUT-	Nicht verfügbar	Prüft, ob die Variable, auf die Element Variant an Operand zeigt, nicht Null und somit ein Objekt ist.

Tabelle 8-39 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Operand	Variant	Operand für die Prüfung auf Null oder nicht Null.

8.4.4.3 IS_ARRAY (Auf ARRAY prüfen)

Mit Anweisung "Auf ARRAY prüfen" kann abgefragt werden, ob Element Variant auf eine Variable des Datentyps Array zeigt.

Der <Operand> muss Datentyp Variant haben.

Die Anweisungen geben 1 (wahr) zurück, wenn der Operand ein Array ist.

Tabelle 8-40 IS_ARRAY (Auf ARRAY prüfen)

KOP	FUP	SCL	Beschreibung
#Operand ┌IS_ARRAY┐	#Operand IS_ARRAY OUT-	IS_ARRAY(_variant_in_)	Prüft, ob die Variable, auf die Element Variant an Operand zeigt, ein Array ist.

Tabelle 8-41 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Operand	Variant	Operand zur Prüfung, ob es sich um einen Array handelt.

8.5 Arithmetische Funktionen

8.5.1 CALCULATE (Berechnen)

Tabelle 8-42 Anweisung CALCULATE

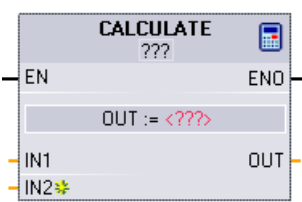
KOP/FUP	SCL	Beschreibung
	Verwenden Sie die herkömmlichen mathematischen SCL-Ausdrücke, um die Gleichung zu erstellen.	<p>Mit der Anweisung CALCULATE können Sie eine mathematische Funktion erstellen, die Eingänge (IN1, IN2, .. INn) verarbeitet und das Ergebnis bei OUT entsprechend der von Ihnen vorgegebenen Gleichung ausgibt.</p> <ul style="list-style-type: none"> Wählen Sie zunächst einen Datentyp aus. Alle Eingänge und der Ausgang müssen den gleichen Datentyp haben. Um einen weiteren Eingang hinzuzufügen, klicken Sie auf das Symbol am letzten Eingang.

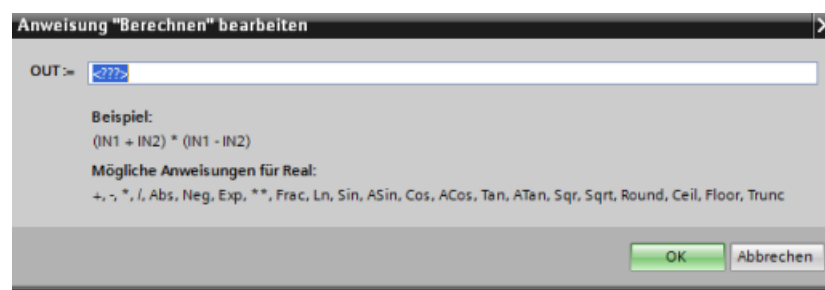
Tabelle 8-43 Datentypen für die Parameter

Parameter	Datentyp 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 Die Parameter IN und OUT müssen denselben Datentyp haben (mit impliziten Umwandlungen der Eingangsparameter).
Beispiel: Ein Wert vom Typ SINT für einen Eingang würde in einen Wert vom Typ INT oder REAL umgewandelt werden, wenn OUT vom Typ INT oder REAL ist.

Klicken Sie auf das Taschenrechnersymbol, um den Dialog aufzurufen und ihre mathematische Funktion zu definieren. Sie geben Ihre Gleichung als Eingänge (wie IN1 und IN2) und Operationen ein. Wenn Sie auf "OK" klicken, um die Funktion zu speichern, erstellt der Dialog automatisch die Eingänge für die Anweisung CALCULATE.

Dieser Dialog zeigt ein Beispiel und eine Liste möglicher Anweisungen, die Sie basierend auf dem Datentyp des Parameters OUT aufnehmen können:



Hinweis

Sie müssen außerdem einen Eingang für die Konstanten in Ihrer Funktion anlegen. Der konstante Wert wird dann in den zugewiesenen Eingang der Anweisung CALCULATE eingegeben.

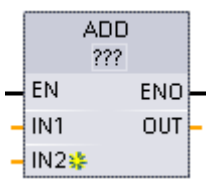
Indem Sie Konstanten als Eingänge eingeben, können Sie die Anweisung CALCULATE an andere Stellen in Ihrem Anwenderprogramm kopieren, ohne die Funktion ändern zu müssen. Sie können dann die Werte oder Variablen der Eingänge für die Anweisung ändern, ohne die Funktion zu verändern.

Wenn die Anweisung CALCULATE ausgeführt wird und alle einzelnen Anweisungen in der Berechnung erfolgreich durchgeführt werden, ist ENO = 1. Andernfalls ist ENO = 0.

Ein Beispiel der Anweisung CALCULATE finden Sie in "AUTOHOTSPOT".

8.5.2 Anweisungen Addieren, Subtrahieren, Multiplizieren und Dividieren

Tabelle 8-44 Anweisungen Addieren, Subtrahieren, Multiplizieren und Dividieren

KOP/FUP	SCL	Beschreibung
	<pre> out := in1 + in2; out := in1 - in2; out := in1 * in2; out := in1 / in2; </pre>	<ul style="list-style-type: none"> • ADD: Addieren ($IN1 + IN2 = OUT$) • SUB: Subtrahieren ($IN1 - IN2 = OUT$) • MUL: Multiplizieren ($IN1 * IN2 = OUT$) • DIV: Dividieren ($IN1 / IN2 = OUT$) <p>Bei einer ganzzahligen Division werden die Nachkommastellen des Quotienten so verkürzt, dass ein ganzzahliger Ausgangswert entsteht.</p>

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-45 Datentypen für die Parameter (KOP und FUP)

Parameter	Datentyp ¹	Beschreibung
IN1, IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Eingänge der arithmetischen Operation
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Ausgang der arithmetischen Operation

¹ Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.



Um einen Eingang ADD oder MUL hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

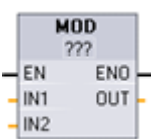
Die arithmetische Anweisung führt, wenn sie aktiviert ist (EN = 1), die angegebene Funktion für die Eingangswerte (IN1 und IN2) aus und speichert das Ergebnis in der vom Ausgangsparameter (OUT) angegebenen Speicheradresse. Nachdem die Operation erfolgreich ausgeführt ist, wird ENO = 1 gesetzt.

Tabelle 8-46 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert der arithmetischen Operation liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Der niederwertigste Teil des Ergebnisses, der in die Zielgröße passt, wird zurückgegeben.
0	Division durch 0 (IN2 = 0): Das Ergebnis ist undefiniert und Null wird zurückgegeben.
0	Real/LReal: Ist einer der Werte NaN (not a number, keine Zahl), so wird der Wert NaN zurückgegeben
0	ADD Real/LReal: Sind beide Eingangswerte INF (unendlich) mit unterschiedlichen Vorzeichen, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	SUB Real/LReal: Sind beide Eingangswerte INF (unendlich) mit demselben Vorzeichen, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	MUL Real/LReal: Ist ein Eingangswert Null und der andere INF, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.
0	DIV Real/LReal: Sind beide Eingangswerte Null oder INF, ist diese Operation ungültig und es wird als Ergebnis NaN zurückgegeben.

8.5.3 MOD (Divisionsrest einer Division)

Tabelle 8-47 Anweisung Modulo (Divisionsrest einer Division)

KOP/FUP	SCL	Beschreibung
	<pre>out := in1 MOD in2;</pre>	<p>Mit der Anweisung MOD können Sie den Divisionsrest einer Ganzzahldivision ausgeben. Der Wert am Eingang IN1 wird durch den Wert am Eingang IN2 dividiert, und der Divisionsrest wird am Ausgang OUT ausgegeben.</p>

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-48 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
IN1 und IN2	SInt, Int, DInt, USInt, UInt, UDInt, Konstante	Modulo-Eingänge
OUT	SInt, Int, DInt, USInt, UInt, UDInt	Modulo-Ausgang

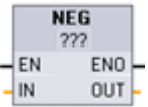
¹ Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.

Tabelle 8-49 ENO-Werte

ENO	Beschreibung
1	Kein Fehler
0	Wert IN2 = 0, OUT wird der Wert Null zugewiesen

8.5.4 NEG (Zweierkomplement erstellen)

Tabelle 8-50 Anweisung NEG (Zweierkomplement erstellen)

KOP/FUP	SCL	Beschreibung
	- (i.n) ;	Mit der Anweisung NEG wird das arithmetische Vorzeichen des Werts von Parameter IN umgekehrt und das Ergebnis im Parameter OUT gespeichert.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-51 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
IN	SInt, Int, DInt, Real, LReal, Konstante	Eingang der arithmetischen Operation
OUT	SInt, Int, DInt, Real, LReal	Ausgang der arithmetischen Operation


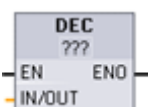
¹ Die Parameter IN und OUT müssen denselben Datentyp haben.

Tabelle 8-52 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: NEG (-128) resultiert in +128, was das Maximum für den Datentyp überschreitet.

8.5.5 INC (Inkrementieren) und DEC (Dekrementieren)

Tabelle 8-53 Anweisungen INC und DEC

KOP/FUP	SCL	Beschreibung
	<code>in_out := in_out + 1;</code>	Erhöht einen ganzzahligen Wert mit oder ohne Vorzeichen: IN_OUT -Wert +1 = IN_OUT -Wert
	<code>in_out := in_out - 1;</code>	Verringert einen ganzzahligen Wert mit oder ohne Vorzeichen: IN_OUT -Wert - 1 = IN_OUT -Wert

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-54 Datentypen für die Parameter

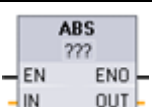
Parameter	Datentyp	Beschreibung
IN/OUT	SInt, Int, DInt, USInt, UInt, UDIInt	Ein- und Ausgang der arithmetischen Operation

Tabelle 8-55 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: INC (+127) ergibt +128, was das Maximum für den Datentyp überschreitet.

8.5.6 ABS (Absolutwert bilden)

Tabelle 8-56 Anweisung ABS (Absolutwert bilden)

KOP/FUP	SCL	Beschreibung
	<code>out := ABS(in);</code>	Berechnet den Absolutwert einer vorzeichenbehafteten Ganzzahl oder Realzahl an Parameter IN und speichert das Ergebnis in Parameter OUT.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-57 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
IN	SInt, Int, DInt, Real, LReal	Eingang der arithmetischen Operation
OUT	SInt, Int, DInt, Real, LReal	Ausgang der arithmetischen Operation



¹ Die Parameter IN und OUT müssen denselben Datentyp haben.

Tabelle 8-58 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Der resultierende Wert der arithmetischen Operation liegt außerhalb des gültigen Zahlenbereichs für den ausgewählten Datentyp. Beispiel für SInt: ABS (-128) ergibt +128, was das Maximum für den Datentyp überschreitet.

8.5.7 MIN (Minimum abrufen) und MAX (Maximum abrufen)

Tabelle 8-59 Anweisungen MIN (Minimum abrufen) und MAX (Maximum abrufen)

KOP/FUP	SCL	Beschreibung
	<pre>out:= MIN(in1:=_variant_in_, in2:=_variant_in_ [...in32]);</pre>	Die Anweisung MIN vergleicht den Wert zweier Parameter IN1 und IN2 und weist den kleineren Wert dem Parameter OUT zu.
	<pre>out:= MAX(in1:=_variant_in_, in2:=_variant_in_ [...in32]);</pre>	Die Anweisung MAX vergleicht den Wert zweier Parameter IN1 und IN2 und weist den größeren Wert dem Parameter OUT zu.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-60 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
IN1, IN2 [...IN32]	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD, Konstante	Eingänge der arithmetischen Operation (max. 32 Eingänge)
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Ausgang der arithmetischen Operation

¹ Die Parameter IN1, IN2 und OUT müssen denselben Datentyp haben.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 8-61 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Nur beim Datentyp Real: <ul style="list-style-type: none"> • Mindestens ein Eingang ist keine Realzahl (NaN). • Der resultierende Ausgang OUT ist +/- INF (unendlich).

8.5.8 LIMIT (Grenzwert setzen)

Tabelle 8-62 Anweisung LIMIT (Grenzwert setzen)

KOP/FUP	SCL	Beschreibung
	<pre>LIMIT (MN:=_variant_in_, IN:=_variant_in_, MX:=_variant_in_, OUT:=_variant_out_);</pre>	Mit der Anweisung Limit können Sie prüfen, ob der Wert von Parameter IN innerhalb des mit Parameter MIN und MAX and if not, clamps the value at MIN or MAX. vorgegebenen Wertebereichs liegt.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-63 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
MN, IN und MX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD-Konstante	Eingänge der arithmetischen Operation
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, Date, TOD	Ausgang der arithmetischen Operation

¹ Die Parameter MN, IN, MX und OUT müssen denselben Datentyp haben.

Liegt der Wert von Parameter IN innerhalb des angegebenen Bereichs, so wird der Wert für IN in Parameter OUT gespeichert. Liegt der Wert von Parameter IN außerhalb des angegebenen Bereichs, wird in OUT der Wert von Parameter MIN (wenn der Wert IN kleiner als der Wert MIN

ist) oder der Wert von Parameter MAX (wenn der Wert IN größer als der Wert MAX ist) ausgegeben.

Tabelle 8-64 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Real: Ist einer der Werte für MIN, IN oder MAX keine Zahl (NaN), wird NaN zurückgegeben.
0	Ist MIN größer als MAX, wird der Wert IN dem Ausgang OUT zugewiesen.

SCL-Beispiele:



- MyVal := LIMIT(MN:=10,IN:=53, MX:=40); //Ergebnis: MyVal = 40
- MyVal := LIMIT(MN:=10,IN:=37, MX:=40); //Ergebnis: MyVal = 37
- MyVal := LIMIT(MN:=10,IN:=8, MX:=40); //Ergebnis: MyVal = 10

8.5.9 Exponential-, Logarithmus- und Trigonometrieanweisungen

Mit den Gleitpunktanweisungen können Sie arithmetische Funktionen mit dem Datentyp Real oder LReal programmieren:

- SQR: Quadrat bilden ($IN^2 = OUT$)
- SQRT: Quadratwurzel bilden ($\sqrt{IN} = OUT$)
- LN: Natürlichen Logarithmus bilden ($LN(IN) = OUT$)
- EXP: Exponentialwert bilden ($e^{IN} = OUT$), mit der Basis $e = 2,71828182845904523536$
- EXPT: Potenzieren ($IN1^{IN2} = OUT$)
die EXPT-Parameter IN1 und OUT haben immer den gleichen Datentyp, für den Sie Real oder LReal auswählen müssen. Für den Exponentialparameter IN2 können Sie den Datentyp aus vielen Datentypen wählen.
- FRAC: Nachkommastellen zurückgeben (Nachkommastelle der Gleitpunktzahl $IN = OUT$)
- SIN: Sinuswert bilden ($\sin(IN \text{ radians}) = OUT$)
- ASIN: Arcussinuswert bilden ($\arcsine(IN) = OUT \text{ radians}$), wobei $\sin(OUT \text{ radians}) = IN$
- COS: Cosinuswert bilden ($\cos(IN \text{ radians}) = OUT$)
- ACOS: Arcuscosinuswert bilden ($\arcsine(IN) = OUT \text{ radians}$), wobei $\cos(OUT \text{ radians}) = IN$
- TAN: Tangenswert bilden ($\tan(IN \text{ radians}) = OUT$)
- ATAN: Arcustangenswert bilden ($\arctan(IN) = OUT \text{ radians}$), wobei $\tan(OUT \text{ radians}) = IN$

Tabelle 8-65 Beispiele für arithmetische Gleitpunktanweisungen

KOP/FUP	SCL	Beschreibung
	<pre>out := SQR(in);</pre> oder <pre>out := in * in;</pre>	Quadrat: $IN^2 = OUT$ Beispiel: Wenn $IN = 9$, dann $OUT = 81$.
	<pre>out := in1 ** in2;</pre>	Potenzieren: $IN1^{IN2} = OUT$ Beispiel: Wenn $IN1 = 3$ und $IN2 = 2$, dann $OUT = 9$.

- ¹ In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.
- ² In SCL: Sie können die mathematischen Ausdrücke in SCL auch mit den grundlegenden mathematischen Operatoren erstellen.

Tabelle 8-66 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN, IN1	Real, LReal, Konstante	Eingänge
IN2	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Konstante	Eingang EXPT-Exponent
OUT	Real, LReal	Ausgänge

Tabelle 8-67 ENO-Zustand

ENO	Anweisung	Bedingung	Ergebnis (OUT)
1	Alle	Kein Fehler	Gültiges Ergebnis
0	SQR	Ergebnis überschreitet den gültigen Bereich für Real/LReal	+INF
		IN ist +/- NaN (keine Zahl)	+NaN
	SQRT	IN ist negativ	-NaN
		IN ist +/- INF (unendlich) oder +/- NaN	+/- INF oder +/- NaN
	LN	IN ist 0,0, negativ, -INF oder -NaN	-NaN
		IN ist +INF oder +NaN	+INF oder +NaN
	EXP	Ergebnis überschreitet den gültigen Bereich für Real/LReal	+INF
		IN ist +/- NaN	+/- NaN
	SIN, COS, TAN	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN
	ASIN, ACOS	IN ist außerhalb des gültigen Bereichs von -1,0 bis +1,0	+NaN
		IN ist +/- NaN	+/- NaN
	ATAN	IN ist +/- NaN	+/- NaN
	FRAC	IN ist +/- INF oder +/- NaN	+NaN
	EXPT	IN1 ist +INF und IN2 ist nicht -INF	+INF
		IN1 ist negativ oder -INF	+NaN, wenn IN2 = Real/LReal, ansonsten -INF
		IN1 oder IN2 ist +/- NaN	+NaN
IN1 ist 0,0 und IN2 ist Real/LReal (nur)		+NaN	

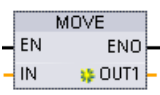
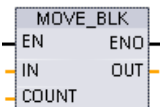
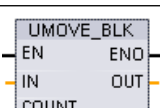
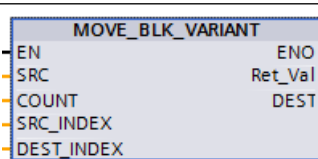
8.6 Anweisungen zum Übertragen von Daten

8.6.1 MOVE (Wert kopieren), MOVE_BLK (Bereich kopieren), UMOVE_BLK (Bereich ununterbrechbar kopieren) und MOVE_BLK_VARIANT (Bereich kopieren)

Mit den Übertragungsanweisungen kopieren Sie Datenelemente in eine neue Adresse im Speicher und wandeln die Daten von einem Datentyp in einen anderen um. Die Quelldaten werden dadurch nicht verändert.

- Mit der Anweisung MOVE wird ein einzelnes Datenelement von der mit Parameter IN angegebenen Quelladresse in die mit Parameter OUT angegebenen Zieladressen kopiert.
- Die Anweisungen MOVE_BLK und UMOVE_BLK verfügen zusätzlich über einen Parameter COUNT. Mit COUNT wird festgelegt, wie viele Datenelemente kopiert werden sollen. Die Anzahl der Bytes pro kopiertem Element hängt davon ab, welcher Datentyp den Variablenamen der Parameter IN und OUT in der PLC-Variablentabelle zugewiesen ist.

Tabelle 8-68 Anweisungen MOVE, MOVE_BLK, UMOVE_BLK und MOVE_BLK_VARIANT

KOP/FUP	SCL	Beschreibung
	<pre>out1 := in;</pre>	Kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse oder in mehrere Adressen. ¹
	<pre>MOVE_BLK (in:= _variant_in, count:= _uint_in, out=> _variant_out);</pre>	Unterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.
	<pre>UMOVE_BLK (in:= _variant_in, count:= _uint_in, out=> _variant_out);</pre>	Ununterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.
	<pre>MOVE_BLK (SRC:= _variant_in, COUNT:= _uint_in, SRC_INDEX:= _dint_in, DEST_INDEX:= _dint_in, DEST=> _variant_out);</pre>	Kopiert den Inhalt eines Quellspeicherbereichs in einen Zielspeicherbereich. Es kann ein komplettes Feld oder Elemente eines Felds in ein anderes Feld mit dem gleichen Datentyp kopiert werden. Die Größen (Anzahl der Elemente) von Quell- und Zielfeld können unterschiedlich sein. Es können mehrere oder einzelne Elemente eines Felds kopiert werden. Es werden Variant-Datentypen verwendet, um auf Quell- und Zielfelder zu zeigen.

¹ Anweisung MOVE: Um einen weiteren Ausgang in KOP oder FUP hinzuzufügen, klicken Sie auf das Symbol "Erstellen" neben dem Ausgangsparameter. In SCL verwenden Sie mehrere Zuweisungsanweisungen. Sie können auch eine der Schleifenkonstruktionen verwenden.

Tabelle 8-69 Datentypen für die Anweisung MOVE

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC Datentypen, PLC-Datentypen	Quelladresse
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Char, WChar, Array, Struct, DTL, Time, Date, TOD, IEC Datentypen, PLC-Datentypen	Zieladresse



Um Ausgänge MOVE hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss und wählen den Befehl "Ausgang einfügen".

8.6 Anweisungen zum Übertragen von Daten

Um einen Ausgang zu löschen, klicken Sie bei einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss (sofern mehr als die zwei ursprünglichen Ausgänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 8-70 Datentypen für die Anweisungen MOVE_BLK und UMOVE_BLK

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal Byte, Word, DWord, Time, Date, TOD, WChar	Anfangsadresse der Quelle
COUNT	UInt	Anzahl der zu kopierenden Datenelemente
OUT	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, WChar	Anfangsadresse des Ziels

Tabelle 8-71 Datentypen für die Anweisung MOVE_BLK_VARIANT

Parameter	Datentyp	Beschreibung
SRC	Variant (zeigt auf ein Feld oder ein einzelnes Feldelement)	Quellbereich für das Kopieren
COUNT	UDInt	Anzahl der zu kopierenden Datenelemente
SRC_INDEX	DInt	Nullbasierter Index in das SRC-Feld
DEST_INDEX	DInt	Nullbasierter Index in das DEST-Feld
RET_VAL	Int	Fehlerinformationen
DEST	Variant (zeigt auf ein Feld oder ein einzelnes Feldelement)	Zielbereich, in den die Inhalte des Quellbereichs kopiert werden

Hinweis**Regeln für die Anweisungen zum Übertragen von Daten**

- Um Daten vom Datentyp Bool zu kopieren, verwenden Sie SET_BF, RESET_BF, R, S oder eine Ausgangsspule (KOP) (Seite 212).
- Um Daten eines einzelnen elementaren Datentyps zu kopieren, verwenden Sie MOVE.
- Um ein Feld eines elementaren Datentyps zu kopieren, verwenden Sie MOVE_BLK oder UMOVE_BLK.
- Um eine Struktur zu kopieren, verwenden Sie MOVE.
- Um eine Zeichenkette zu kopieren, verwenden Sie S_MOVE (Seite 337).
- Um ein einzelnes Zeichen in einer Zeichenkette zu kopieren, verwenden Sie MOVE.
- Die Operationen MOVE_BLK und UMOVE_BLK können Sie nicht verwenden, um Felder oder Strukturen in die Speicherbereiche E, A oder M zu kopieren.

Die Operationen MOVE_BLK und UMOVE_BLK unterscheiden sich in der Verarbeitung von Alarmen:

- Alarmereignisse werden **in die Warteschlange gestellt** und während der Ausführung von MOVE_BLK verarbeitet. Die Operation MOVE_BLK nutzen Sie, wenn die Daten an der Zieladresse der Übertragung nicht in einem Unterprogramm eines Alarm-OBs verwendet werden, oder, sofern sie verwendet werden, die Zieldaten nicht konsistent sein müssen. Wenn eine Operation MOVE_BLK unterbrochen wird, ist das zuletzt übertragene Datenelement an der Zieladresse vollständig und konsistent. Die Operation MOVE_BLK wird nach Ausführung des Alarm-OBs fortgesetzt.
- Alarmereignisse werden **in die Warteschlange gestellt, aber erst verarbeitet**, wenn die Ausführung von UMOVE_BLK beendet ist. Die Operation UMOVE_BLK nutzen Sie, wenn die Übertragungsoperation beendet und die Zieldaten konsistent sein müssen, bevor das Unterprogramm eines Alarm-OBs ausgeführt wird. Beachten Sie für weitere Informationen den Abschnitt zur Datenkonsistenz (Seite 187).

ENO ist nach der Ausführung der Operation MOVE immer wahr.

Tabelle 8-72 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Alle COUNT-Elemente wurden erfolgreich kopiert.
0	Der Quellbereich (IN) oder der Zielbereich (OUT) überschreitet den verfügbaren Speicherbereich.	Elemente, die passen, werden kopiert. Es werden keine Teilelemente kopiert.

Tabelle 8-73 Bedingungscode für die Anweisung MOVE_BLK_VARIANT

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
80B4	Datentypen passen nicht.
8151	Zugriff auf Parameter SRC nicht möglich.
8152	Der Operand von Parameter SRC hat einen ungültigen Typ.
8153	Fehler bei der Codegenerierung an Parameter SRC
8154	Der Operand von Parameter SRC hat Datentyp Bool.
8281	Parameter COUNT hat einen ungültigen Wert.
8382	Der Wert von Parameter SRC_INDEX liegt außerhalb des Grenzwerts für Variant.
8383	Der Wert von Parameter SRC_INDEX übersteigt den oberen Grenzwert für das Feld.
8482	Der Wert von Parameter DEST_INDEX liegt außerhalb des Grenzwerts für Variant.
8483	Der Wert von Parameter DEST_INDEX übersteigt den oberen Grenzwert für das Feld.
8534	Parameter DEST ist schreibgeschützt.
8551	Zugriff auf Parameter DEST nicht möglich.
8552	Der Operand von Parameter DEST hat einen ungültigen Typ.

RET_VAL (W#16#...)	Beschreibung
8553	Fehler bei der Codegenerierung an Parameter DEST
8554	Der Operand von Parameter DEST hat Datentyp Bool.
*Feldercodes können im Programmierer als Ganzzahlen oder Hexadezimalwerte angezeigt werden.	

8.6.2 Deserialize

Mit der Anweisung "Deserialize" kann die sequentielle Darstellung eines PLC-Datentyps (UDT) in einen PLC-Datentyp zurückverwandelt und sein gesamter Inhalt aufgefüllt werden. Ist der Vergleich WAHR, ist der Box-Ausgang WAHR.

Der Speicherbereich mit der sequentiellen Darstellung eines PLC-Datentyps muss den Datentyp "Array of Byte" aufweisen und der Datenbaustein muss für Standardzugriff (nicht optimierten Zugriff) deklariert sein. Vor der Umwandlung ist zu prüfen, ob genug Speicherplatz zur Verfügung steht.

Mit dieser Anweisung können mehrere sequentielle Darstellungen umgewandelter PLC-Datentypen in ihre ursprünglichen Datentypen zurückgewandelt werden.

Hinweis

Soll nur eine einzelne sequentielle Darstellung eines PLC-Datentyps (UDT) zurückgewandelt werden, kann Anweisung "TRCV: Daten über Kommunikationsverbindung empfangen" verwendet werden.

Tabelle 8-74 Anweisung DESERIALIZE

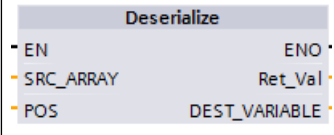
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := Deserialize(SRC_ARRAY:=_variant_in_, DEST_VARIABLE=>_variant_out_ , POS:=_dint_inout_);</pre>	Wandelt die sequentielle Darstellung eines PLC-Datentyps (UDT) zurück in einen PLC-Datentyp und füllt den gesamten Inhalt auf.

Tabelle 8-75 Parameter für die DESERIALIZE-Anweisung

Parameter	Typ	Datentyp	Beschreibung
SRC_ARRAY	IN	Variant	Globaler Datenbaustein mit dem Datenstrom
DEST_VARIABLE	INOUT	Variant	Variable, in der der umgewandelte PLC-Datentyp (UDT) gespeichert werden soll
POS	INOUT	DInt	Anzahl Bytes, die der umgewandelte PLC-Datentyp verwendet
RET_VAL	OUT	Int	Fehlerinformationen

Tabelle 8-76 Parameter RET_VAL

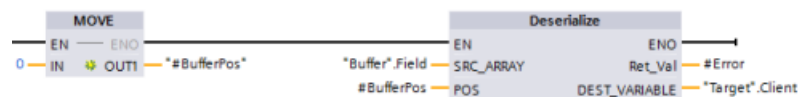
RET_VAL* (W#16#...)	Beschreibung
0000	Kein Fehler
80B0	Die Speicherbereiche für die Parameter SRC_ARRAY und DEST_VARIABLE überlappen sich.
8136	Der Datenbaustein an Parameter DEST_VARIABLE ist kein Baustein mit Standardzugriff.
8150	Datentyp Variant von Parameter SRC_ARRAY enthält keinen Wert.
8151	Fehler bei der Codegenerierung an Parameter SRC_ARRAY.
8153	Nicht genug freier Speicherplatz für Parameter SRC_ARRAY.
8250	Datentyp Variant von Parameter DEST_VARIABLE enthält keinen Wert.
8251	Fehler bei der Codegenerierung an Parameter DEST_VARIABLE.
8254	Ungültiger Datentyp für Parameter DEST_VARIABLE.
8382	Der Wert von Parameter POS liegt außerhalb des Grenzwerts für das Feld.

*Feldercode können im Programmierer als Ganzzahlen oder Hexadezimalwerte angezeigt werden.

Beispiel: Anweisung "Deserialize"

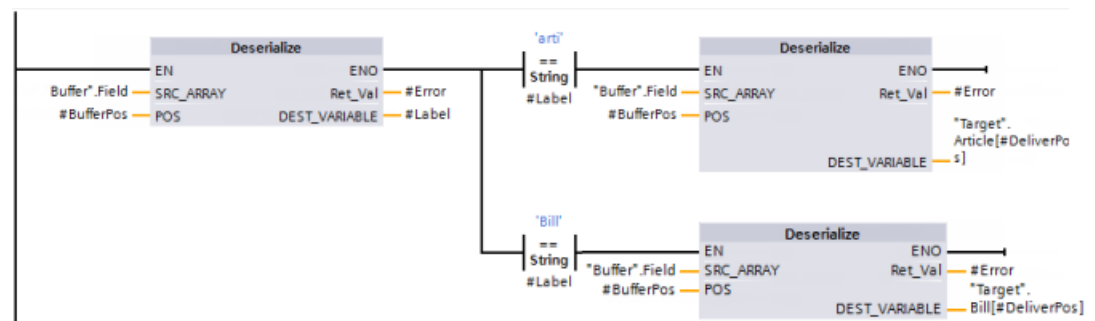
Das folgende Beispiel zeigt, wie die Anweisung funktioniert:

Netzwerk 1:



Mit Anweisung "MOVE" wird Wert "0" in die Variable des Datenbausteins "#BufferPos" kopiert. Mit Anweisung "Deserialize" wird dann die sequentielle Darstellung der Kundendaten aus dem Datenbaustein "Buffer" deserialisiert und in Datenbaustein "Target" geschrieben. Mit Anweisung "Deserialize" wird die Anzahl Bytes berechnet, die von den umgewandelten Daten benutzt werden, und in der Variablen des Datenbausteins "#BufferPos" gespeichert.

Netzwerk 2:



Mit Anweisung "Deserialize" wird die sequentielle Darstellung des Datenstroms, auf den mit "Buffer" gezeigt wird, deserialisiert und die Zeichen in Operand "#Label" geschrieben. Die Logik vergleicht die Zeichen mit Hilfe der Vergleichsoperationen "arti" und "Bill". Ist der Vergleich für "arti" = WAHR, sind die Daten Artikeldaten, die deserialisiert und in Datenstruktur "Article" des Datenbausteins "Target" geschrieben werden. Ist der Vergleich für "Bill" = WAHR, sind die Daten

Abrechnungsdaten, die deserialisiert und in Datenstruktur "Bill" des Datenbausteins "Target" geschrieben werden.

Funktionsbausteinschnittstelle (oder Funktionsschnittstelle):

	Name	Datentyp
1	Input	
2	DeliverPos	Int
3	Output	
4	InOut	
5	Static	
6	Temp	
7	BufferPos	DInt
8	Error	Int
9	Label	String[4]

Benutzerdefinierte PLC-Datentypen:

Die Strukturen der beiden PLC-Datentypen (UDT) für dieses Beispiel sind nachstehend dargestellt:

Article		
	Name	Datentyp
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Name	Datentyp
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Datenbausteine:

Die beiden Datenbausteine für dieses Beispiel sind nachfolgend dargestellt:

Target		
	Name	Datentyp
1	Static	
2	Client	*Client*
3	Article	Array[0..10] of *Article*
4	Bill	Array[0..10] of Int

Buffer		
	Name	Datentyp
1	Static	
2	Field	Array[0..294] of Byte

8.6.3 Serialize

Mit der Anweisung "Serialize" können mehrere PLC-Datentypen (UDT) ohne jeglichen Strukturverlust in eine sequentielle Darstellung umgewandelt werden.

Die Anweisung kann verwendet werden, um mehrere strukturierte Datenelemente aus Ihrem Programm vorübergehend in einem Puffer abzulegen, zum Beispiel einem globalen Datenbaustein, und sie dann in eine andere CPU zu übertragen. Der Speicherbereich, in dem die umgewandelten PLC-Datentypen abgelegt werden, muss Datentyp "ARRAY of BYTE" aufweisen und mit Standardzugriff deklariert sein. Vor der Umwandlung ist zu prüfen, ob genug Speicherplatz zur Verfügung steht.

Der POS-Parameter enthält Informationen über die Anzahl Bytes, die die umgewandelten PLC-Datentypen verwenden.

Hinweis

Soll nur ein einzelner PLC-Datentyp (UDT) übertragen werden, kann dafür Anweisung "TSEND: Daten über Kommunikationsverbindung senden" verwendet werden.

Tabelle 8-77 Anweisung SERIALIZE

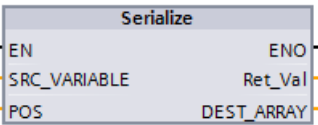
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := Serialize(SRC_VARIABLE=>_variant_in_, DEST_ARRAY:=_variant_out_, POS:=_dint_inout_);</pre>	Wandelt einen PLC-Datentyp (UDT) in eine sequentielle Darstellung um.

Tabelle 8-78 Parameter für die SERIALIZE-Anweisung

Parameter	Typ	Datentyp	Beschreibung
SRC_VARIABLE	IN	Variant	PLC-Datentyp (UDT), der in eine serielle Darstellung umgewandelt werden soll
DEST_ARRAY	INOUT	Variant	Datenbaustein, in dem der generierte Datenstrom gespeichert werden soll
POS	INOUT	DInt	Anzahl Bytes, die die umgewandelten PLC-Datentypen verwenden. Der berechnete POS-Parameter ist nullbasiert.
RET_VAL	OUT	Int	Fehlerinformationen

Tabelle 8-79 Parameter RET_VAL

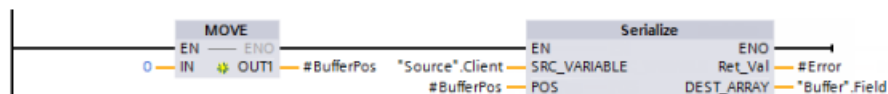
RET_VAL* (W#16#...)	Beschreibung
0000	Kein Fehler
80B0	Die Speicherbereiche für die Parameter SRC_VARIABLE und DEST_ARRAY überlappen sich.
8150	Datentyp Variant von Parameter SRC_VARIABLE enthält keinen Wert.
8152	Fehler bei der Codegenerierung an Parameter SRC_VARIABLE.
8236	Der Datenbaustein an Parameter DEST_ARRAY ist kein Baustein mit Standardzugriff.
8250	Datentyp Variant von Parameter DEST_ARRAY enthält keinen Wert.
8252	Fehler bei der Codegenerierung an Parameter DEST_ARRAY.
8253	Nicht genug freier Speicherplatz für Parameter DEST_ARRAY.
8254	Ungültiger Datentyp für Parameter DEST_VARIABLE.

RET_VAL* (W#16#...)	Beschreibung
8382	Der Wert von Parameter POS liegt außerhalb des Grenzwerts für das Feld.
*Feldercode können im Programmeditor als Ganzzahlen oder Hexadezimalwerte angezeigt werden.	

Beispiel: Anweisung Serialize

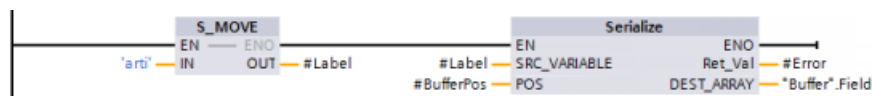
Das folgende Beispiel zeigt, wie die Anweisung funktioniert:

Netzwerk 1:



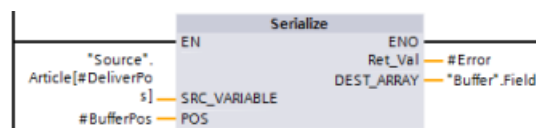
Mit Anweisung "MOVE" wird Wert "0" in Parameter "#BufferPos" kopiert. Mit Anweisung "Serialize" werden die Kundendaten aus dem Datenbaustein "Source" serialisiert und in sequentieller Darstellung in Datenbaustein "Buffer" geschrieben. Die Anweisung speichert die Anzahl der von der sequentiellen Darstellung verwendeten Bytes in Parameter "#BufferPos".

Netzwerk 2:



Die Logik fügt dann einen Trenntext ein, damit die sequentielle Darstellung später einfacher deserialisiert werden kann. Mit Anweisung "S_MOVE" wird die Zeichenfolge "arti" in Parameter "#Label" kopiert. Anweisung "Serialize" schreibt diese Zeichen nach den Quelldaten in Datenbaustein "Buffer". Die Anweisung fügt die Anzahl Bytes in der Textzeichenfolge "arti" zu der in Parameter "#BufferPos" bereits gespeicherten Anzahl hinzu.

Netzwerk 3:



Anweisung "Serialize" serialisiert die Daten eines bestimmten Artikels, die während der Laufzeit berechnet werden, aus Datenbaustein "Source" und schreibt sie in sequentieller Darstellung in Datenbaustein "Buffer" nach den Zeichen "arti".

Bausteinschnittstelle:

	Name	Datentyp
1	▼ Input	
2	■ DeliverPos	Int
3	► Output	
4	► InOut	
5	► Static	
6	▼ Temp	
7	■ BufferPos	DInt
8	■ Error	Int
9	■ Label	String[4]

Benutzerdefinierte PLC-Datentypen:

Die Strukturen der beiden PLC-Datentypen (UDT) für dieses Beispiel sind nachstehend dargestellt:

Article		
	Name	Datentyp
1	Number	DInt
2	Declaration	String
3	Colli	Int

Client		
	Name	Datentyp
1	Title	Int
2	Firstname	String[10]
3	Surname	String[10]

Datenbausteine:

Die beiden Datenbausteine für dieses Beispiel sind nachstehend dargestellt:

Source		
	Name	Datentyp
1	▼ Static	
2	► Client	*Client*
3	► Article	Array[0..10] of *Article*

Buffer		
	Name	Datentyp
1	▼ Static	
2	► Field	Array[0..294] of Byte

8.6.4 FILL_BLK (Speicher mit Bitmuster belegen) und UFILL_BLK (Speicher ununterbrechbar mit Bitmuster belegen)

Tabelle 8-80 Anweisungen FILL_BLK und UFILL_BLK

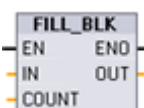
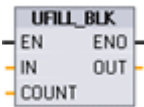
KOP/FUP	SCL	Beschreibung
	<pre>FILL_BLK (in:=_variant_in, count:=int, out=>_variant_out);</pre>	Anweisung Unterbrechbar befüllen: Befüllt einen Adressbereich mit Kopien eines angegebenen Datenelements.
	<pre>UFILL_BLK (in:=_variant_in, count:=int, out=>_variant_out);</pre>	Anweisung Ununterbrechbar befüllen: Befüllt einen Adressbereich mit Kopien eines angegebenen Datenelements.

Tabelle 8-81 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Datenquelladresse
COUNT	UDInt, USInt, UInt	Anzahl der zu kopierenden Datenelemente
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Datenzieladresse

Hinweis**Regeln für die Anweisungen Speicher mit Bitmuster belegen**

- Um Daten vom Datentyp BOOL zu kopieren, verwenden Sie SET_BF, RESET_BF, R, S oder eine Ausgangsspule (KOP).
- Um den Speicher mit einem einzelnen elementaren Datentyp zu belegen, verwenden Sie MOVE.
- Um ein Feld mit einem elementaren Datentyp zu belegen, verwenden Sie FILL_BLK oder UFILL_BLK.
- Um ein einzelnes Zeichen in einer Zeichenkette zu kopieren, verwenden Sie MOVE.
- Die Anweisungen FILL_BLK und UFILL_BLK können Sie nicht verwenden, um Felder in den Speicherbereichen E, A oder M zu belegen.

Mit den Anweisungen FILL_BLK und UFILL_BLK wird das Quelldatenelement IN in die Zieladresse kopiert, wobei die Anfangsadresse im Parameter OUT festgelegt ist. Das Kopieren läuft so lange in den jeweils benachbarten Adressbereichen weiter, bis die Anzahl der Kopien mit dem Wert des Parameters COUNT übereinstimmt.

Die Anweisungen FILL_BLK und UFILL_BLK unterscheiden sich in der Verarbeitung von Alarmen:

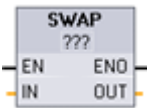
- Alarmereignisse werden **in die Warteschlange gestellt** und während der Ausführung von FILL_BLK verarbeitet. Die Anweisung FILL_BLK nutzen Sie, wenn die Daten an der Zieladresse der Übertragung nicht in einem Unterprogramm eines Alarm-OBS verwendet werden, oder, sofern sie verwendet werden, die Zieldaten nicht konsistent sein müssen.
- Alarmereignisse werden **in die Warteschlange gestellt, aber erst verarbeitet**, wenn die Ausführung von UFILL_BLK beendet ist. Die Anweisung UFILL_BLK nutzen Sie, wenn die Übertragungsanweisung beendet und die Zieldaten konsistent sein müssen, bevor das Unterprogramm eines Alarm-OBS ausgeführt wird.

Tabelle 8-82 ENO-Zustand

ENO	Bedingung	Ergebnis
1	Kein Fehler	Das Element IN wurde erfolgreich in alle COUNT-Ziele kopiert.
0	Der Zielbereich (OUT) überschreitet den verfügbaren Speicherbereich	Elemente, die passen, werden kopiert. Es werden keine Teilelemente kopiert.

8.6.5 SWAP (Anordnung ändern)

Tabelle 8-83 Anweisung SWAP

KOP/FUP	SCL	Beschreibung
	<code>out := SWAP(in);</code>	Kehrt die Bytereihenfolge bei Zwei-Byte- und Vier-Byte-Datenelementen um. Die Bitfolge innerhalb eines Bytes wird nicht geändert. ENO ist nach der Ausführung der Anweisung SWAP immer WAHR.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-84 Datentypen für die Parameter

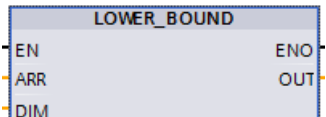
Parameter	Datentyp	Beschreibung
IN	Word, DWord	Normal angeordnete Datenbytes in Parameter IN
OUT	Word, DWord	Vertauschte Reihenfolge der Datenbytes in Parameter OUT

Beispiel 1	Parameter IN = MB0 (vor Ausführung)	Parameter OUT = MB4, (nach Ausführung)
Adresse	MW0 MB1	MW4 MB5
W#16#1234	12 34	34 12
WORD	MSB LSB	MSB LSB

Beispiel 2	Parameter IN = MB0 (vor Ausführung)	Parameter OUT = MB4, (nach Ausführung)
Adresse	MDO MB1 MB2 MB3	MD4 MB5 MB6 MB7
DW#16#	12 34 56 78	78 56 34 12
12345678	MSB	LSB MSB
DWORD		LSB

8.6.6 LOWER_BOUND: (Untere ARRAY-Grenze auslesen)

Tabelle 8-85 Anweisung LOWER_BOUND

KOP/FUP	SCL	Beschreibung
	<code>out := LOWER_BOUND(ARR:=_variant_in_, DIM:=_udint_in_);</code>	In der Bausteinschnittstelle können Sie Variablen mit ARRAY[*] deklarieren. Für diese lokalen Variablen können Sie die Grenzwerte des ARRAY auslesen. Sie müssen die erforderliche Dimension im Parameter DIM angeben. Mit der Anweisung LOWER_BOUND (Untere ARRAY-Grenze auslesen) können Sie den unteren Variablengrenzwert des ARRAY auslesen.

Parameter

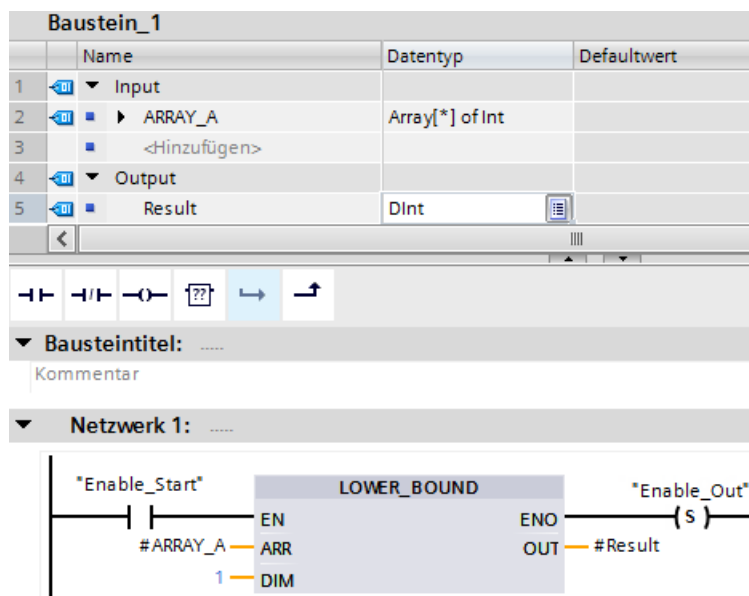
Die folgende Tabelle zeigt die Parameter der Anweisung LOWER_BOUND: Untere ARRAY-Grenze auslesen

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Input	BOOL	E, A, M, D, L	Freigabeeingang
ENO	Output	BOOL	E, A, M, D, L	Der Freigabeausgang ENO hat den Signalzustand 0, wenn eine der folgenden Bedingungen zutrifft: <ul style="list-style-type: none"> Der Freigabeeingang EN hat den Signalzustand 0. Die am Eingang DIM angegebene Dimension ist nicht vorhanden.
ARR	Input	ARRAY [*]	FB: Bereich InOut FC: Bereiche Input und InOut	ARRAY, dessen untere Variablenlänge gelesen werden soll.
DIM	Input	UDINT	E, A, M, D, L oder Konstante	Dimension des ARRAY, dessen untere Variablenlänge gelesen werden soll.
OUT	Output	DINT	E, A, M, D, L	Ergebnis

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)":

Beispiel

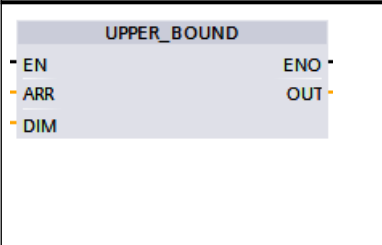
In der Bausteinschnittstelle der Funktion (FC) ist der Eingangsparameter ARRAY_A ein eindimensionales Array mit Variablendimensionen.



Wenn der Operand "Enable_Start" den Signalzustand 1 ausgibt, führt die CPU die Anweisung LOWER_BOUND aus. Die CPU liest die untere Variablengrenze des ARRAY #ARRAY_A aus dem eindimensionalen Array aus. Wird die Anweisung mit Fehlern ausgeführt, werden der Operand "Enable_Out" gesetzt und der Operand "Result" auf den unteren Grenzwert des Arrays gesetzt.

8.6.7 UPPER_BOUND: (Obere ARRAY-Grenze auslesen)

Tabelle 8-86 Anweisung LOWER_BOUND

KOP/FUP	SCL	Beschreibung
	<pre>out := UPPER_BOUND(ARR:=_variant_in_, DIM:=_udint_in_);</pre>	<p>In der Bausteinschnittstelle können Sie Variablen mit ARRAY[*] deklarieren. Für diese lokalen Variablen können Sie die Grenzwerte des ARRAY auslesen. Sie müssen die erforderliche Dimension im Parameter DIM angeben.</p> <p>Mit der Anweisung UPPER_BOUND (Obere ARRAY-Grenze auslesen) können Sie den oberen Variablengrenzwert des ARRAY auslesen.</p>

Parameter

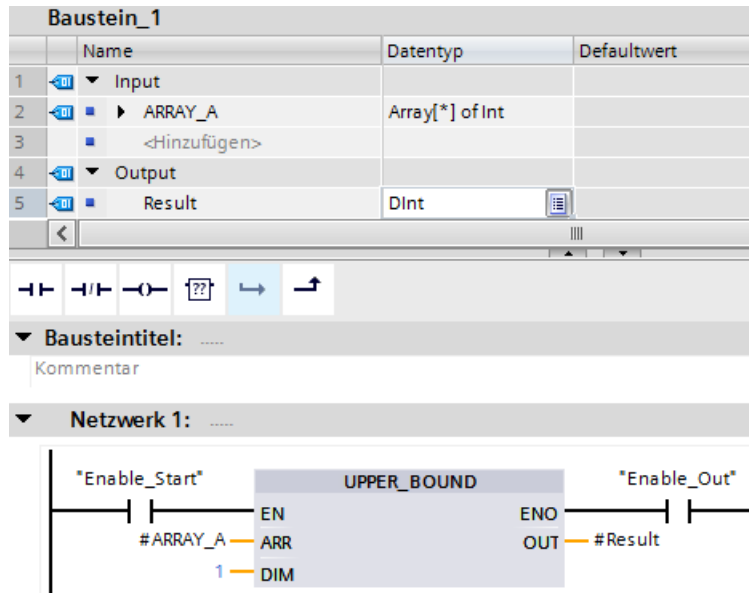
Die folgende Tabelle zeigt die Parameter der Anweisung UPPER_BOUND: (Obere ARRAY-Grenze auslesen):

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
EN	Input	BOOL	E, A, M, D, L	Freigabeeingang
ENO	Output	BOOL	E, A, M, D, L	Freigabeausgang
ARR	Input	ARRAY [*]	FB: Bereich InOut FC: Bereiche Input und InOut	ARRAY, dessen obere Variablengrenze gelesen werden soll.
DIM	Input	UDINT	E, A, M, D, L oder Konstante	Dimension des ARRAY, dessen obere Variablengrenze gelesen werden soll.
OUT	Output	DINT	E, A, M, D, L	Ergebnis

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)":

Beispiel

In der Bausteinschnittstelle der Funktion (FC) ist der Eingangsparameter ARRAY_A ein eindimensionales Array mit Variablendimensionen.



Wenn der Operand "Enable_Start" den Signalzustand 1 ausgibt, führt die CPU die Anweisung aus. Sie liest die obere Variablengrenze des ARRAY #ARRAY_A aus dem eindimensionalen Array aus. Wird die Anweisung mit Fehlern ausgeführt, werden der Operand "Enable_Out" und der Operand "Result" gesetzt.

8.6.8 Anweisungen Speicher lesen / in Speicher schreiben

8.6.8.1 PEEK und POKE (nur SCL)

SCL bietet die Anweisungen PEEK und POKE, mit denen Sie aus Datenblöcken, E/A oder dem Speicher lesen oder in diese schreiben können. Sie geben für die Operation Parameter mit einem spezifischen Byte- oder Bit-Versatz an.

Hinweis

Um die Anweisungen PEEK und POKE mit Datenbausteinen zu verwenden, müssen Sie Standarddatenbausteine (keine optimierten Datenbausteine) verwenden. Beachten Sie zudem, dass die Anweisungen PEEK und POKE lediglich Daten übertragen. Die Datentypen an den Adressen werden nicht beachtet.

```
PEEK(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_);
```

Liest das von byteOffset angegebene Byte aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel für den Verweis auf einen Datenbaustein:

```
%MB100 := PEEK(area:=16#84,
               dbNumber:=1, byteOffset:=#i);
```

Beispiel für den Verweis auf Eingang EB3:

```
%MB100 := PEEK(area:=16#81,
               dbNumber:=0, byteOffset:=#i); // when
#i = 3
```

```
PEEK_WORD(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_);
```

Liest das von byteOffset angegebene Wort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MW200 := PEEK_WORD(area:=16#84,
                    dbNumber:=1, byteOffset:=#i);
```

```
PEEK_DWORD(area:=_in_,
            dbNumber:=_in_,
            byteOffset:=_in_);
```

Liest das von byteOffset angegebene Doppelwort aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MD300 := PEEK_DWORD(area:=16#84,
                     dbNumber:=1, byteOffset:=#i);
```

```
PEEK_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_);
```

Liest einen von bitOffset und byteOffset angegebenen Booleschen Wert aus dem angegebenen Datenbaustein, aus den E/A oder dem Speicherbereich.

Beispiel:

```
%MB100.0 := PEEK_BOOL(area:=16#84,
                      dbNumber:=1, byteOffset:=#ii,
                      bitOffset:=#j);
```

```
POKE(area:=_in_,
      dbNumber:=_in_,
      byteOffset:=_in_,
      value:=_in_);
```

Schreibt den Wert (Byte, Wort oder Doppelwort) in den angegebenen byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel für den Verweis auf einen Datenbaustein:

```
POKE(area:=16#84, dbNumber:=2,
      byteOffset:=3, value:="Tag_1");
```

Beispiel für den Verweis auf Ausgang AB3:

```
POKE(area:=16#82, dbNumber:=0,
      byteOffset:=3, value:="Tag_1");
```

```
POKE_BOOL(area:=_in_,
           dbNumber:=_in_,
           byteOffset:=_in_,
           bitOffset:=_in_,
           value:=_in_);
```

```
POKE_BLK(area_src:=_in_,
          dbNumber_src:=_in_,
          byteOffset_src:=_in_,
          area_dest:=_in_,
          dbNumber_dest:=_in_,
          byteOffset_dest:=_in_,
          count:=_in_);
```

Schreibt den Booleschen Wert in den angegebenen bitOffset und byteOffset des angegebenen Datenbausteins, der E/A oder des Speicherbereichs.

Beispiel:

```
POKE_BOOL(area:=16#84, dbNumber:=2,
           byteOffset:=3, bitOffset:=5,
           value:=0);
```

Schreibt die unter "count" angegebene Anzahl von Bytes beginnend mit dem angegebenen byteOffset des angegebenen Quelldatenbausteins, der Quell-E/A oder des Quellspeicherbereichs in den angegebenen byteOffset des angegebenen Zieldatenbausteins, der Ziel-E/A oder des Zielspeicherbereichs.

Beispiel:

```
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);
```

Bei den Anweisungen PEEK und POKE gelten die folgenden Werte für die Parameter "area", "area_src" und "area_dest". Für andere Bereiche als Datenbausteine muss der Parameter dbNumber 0 sein.

16#81	I
16#82	Q
16#83	M
16#84	DB

8.6.8.2 Anweisungen Big- und Little-Endian-Format lesen und schreiben (SCL)

Die S7-1200 CPU stellt SCL-Anweisungen zum Lesen und Schreiben von Daten in den Formaten Little Endian und Big Endian bereit. Little-Endian-Format bedeutet, dass das Byte mit dem niederwertigsten Bit an der kleinsten Speicheradresse gespeichert wird. Big-Endian-Format bedeutet, dass das Byte mit dem höchstwertigen Bit an der kleinsten Speicheradresse gespeichert wird.

Die vier SCL-Anweisungen für das Lesen und Schreiben von Daten im Little- und Big-Endian-Format sind nachstehend beschrieben:

- READ_LITTLE (Daten in Little-Endian-Format lesen)
- WRITE_LITTLE (Daten in Little-Endian-Format schreiben)
- READ_BIG (Daten in Big-Endian-Format lesen)
- WRITE_BIG (Daten in Big-Endian-Format schreiben)

Tabelle 8-87 Anweisungen Big- und Little-Endian-Format lesen und schreiben

KOP/FUP	SCL	Beschreibung
Nicht verfügbar	<code>READ_LITTLE(</code> <code> src_array:=_variant_in_</code> <code> dest_variable =>_out_</code> <code> pos:=_dint_inout)</code>	Liest Daten aus einem Speicherbereich und schreibt sie in eine einzelne Variable im Little-Endian-Byte-Format.
Nicht verfügbar	<code>WRITE_LITTLE(</code> <code> src_variable:=_in_</code> <code> dest_array =>_variant_inout_</code> <code> pos:=_dint_inout)</code>	Schreibt Daten aus einer einzelnen Variablen in einen Speicherbereich im Little-Endian-Byte-Format.
Nicht verfügbar	<code>READ_BIG(</code> <code> src_array:=_variant_in_</code> <code> dest_variable =>_out_</code> <code> pos:=_dint_inout)</code>	Liest Daten aus einem Speicherbereich und schreibt sie in eine einzelne Variable im Big-Endian-Byte-Format.
Nicht verfügbar	<code>WRITE_BIG(</code> <code> src_variable:=_in_</code> <code> dest_array =>_variant_inout_</code> <code> pos:=_dint_inout)</code>	Schreibt Daten aus einer einzelnen Variablen in einen Speicherbereich im Big-Endian-Byte-Format.

Tabelle 8-88 Parameter für die READ_LITTLE and READ_BIG -Anweisung

Parameter	Datentyp	Beschreibung
src_array	Array of Byte	Speicherbereich, aus dem Daten gelesen werden sollen
dest_variable	Bitzeichenfolgen, Ganzzahlen, Gleitpunktzahlen, Zeiten, Datum und Uhrzeit, Zeichenfolgen	Zielvariable, in die Daten zu schreiben sind
pos	DINT	Nullbasierte Position, ab der Daten aus Eingang src_array gelesen werden sollen.

Tabelle 8-89 Parameter für die WRITE_LITTLE and WRITE_BIG -Anweisung

Parameter	Datentyp	Beschreibung
src_variable	Bitzeichenfolgen, Ganzzahlen, Gleitpunktzahlen, TOD, DATA, Char, WChar	Quelldaten aus Variable
dest_array	Array of Byte	Speicherbereich, in den Daten geschrieben werden sollen
pos	DINT	Nullbasierte Position, ab der Daten in Ausgang dest_array geschrieben werden sollen.

Tabelle 8-90 Parameter RET_VAL

RET_VAL* (W#16#...)	Beschreibung
0000	Kein Fehler
80B4	SRC_ARRAY oder DEST_ARRAY ist kein Array of Byte

RET_VAL* (W#16#...)	Beschreibung
8382	Der Wert von Parameter POS liegt außerhalb des Grenzwerts für das Feld.
8383	Der Wert von Parameter POS liegt innerhalb der Grenzwerte des Felds, aber die Größe des Speicherbereichs überschreitet die Obergrenze des Felds.

*Feldercode können im Programmmeditor als Ganzzahlen oder Hexadezimalwerte angezeigt werden.

8.6.9 Variant-Anweisungen

8.6.9.1 VariantGet (VARIANT Variablenwert lesen)

Mit der Anweisung "Variant Variablenwert lesen" kann der Wert der Variablen ausgelesen werden, auf die der Pointer Variant am Parameter SRC zeigt, und in die Variable von Parameter DST geschrieben werden.

Parameter SRC hat Datentyp Variant. Für Parameter DST kann jeder Datentyp außer Variant programmiert werden.

Der Datentyp der Variablen in Parameter DST muss mit dem Datentyp übereinstimmen, auf den Variant zeigt.

Tabelle 8-91 Anweisung VariantGet

KOP/FUP	SCL	Beschreibung
VariantGet EN ENO SRC DST	<pre>VariantGet (SRC:=_variant_in_, DST=>_variant_out_);</pre>	Liest die Variable, auf die Parameter SRC zeigt und schreibt sie in die Variable in Parameter DST.

Hinweis

Um Strukturen und Felder zu kopieren, kann Anweisung "MOVE_BLK_VARIANT: Bereich kopieren" verwendet werden.

Tabelle 8-92 Parameter für die VariantGet-Anweisung

Parameter	Datentyp	Beschreibung
SRC	Variant	Pointer auf Quelldaten
DST	Datentypen Bitzeichenfolgen, Ganzzahlen, Gleitpunktzahlen, Zeiten, Datum und Uhrzeit, Zeichenfolgen, ARRAY-Elemente, PLC	Ziel, in das die Daten zu schreiben sind

Tabelle 8-93 ENO-Zustand

ENO	Bedingung	Ergebnis
1	Kein Fehler	Die Anweisung hat die Variablendaten, auf die SRC zeigt, in Variable DST kopiert.
0	Freigabeeingang EN hat Signalzustand "0" oder Datentypen passen nicht.	Die Anweisung hat keine Daten kopiert.

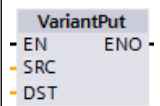
8.6.9.2 VariantPut (VARIANT Variablenwert schreiben)

Mit Anweisung "VARIANT Variablenwert schreiben" kann der Wert der Variablen in Parameter SRC in die Variable in Parameter DST geschrieben werden, auf die VARIANT zeigt.

Parameter DST hat Datentyp VARIANT. Für Parameter SRC kann jeder Datentyp außer VARIANT programmiert werden.

Der Datentyp der Variablen in Parameter SRC muss mit dem Datentyp übereinstimmen, auf den VARIANT zeigt.

Tabelle 8-94 Anweisung VariantPut

KOP/FUP	SCL	Beschreibung
	<pre>VariantPut(SRC:=_variant_in_, DST=>_variant_in_);</pre>	Schreibt die mit Parameter SRC referenzierte Variable in das Variant-Element, auf das Parameter DST zeigt.

Hinweis

Um Strukturen und Felder zu kopieren, kann Anweisung "MOVE_BLK_VARIANT: Bereich kopieren" verwendet werden.

Tabelle 8-95 Parameter für die VariantPut-Anweisung

Parameter	Datentyp	Beschreibung
SRC	Bit strings, integers, floating-point numbers, timers, date and time, character strings, ARRAY elements, PLC data types	Pointer auf Quelldaten
DST	Variant	Ziel, in das die Daten zu schreiben sind

Tabelle 8-96 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Die Anweisung hat die SRC-Variablendaten in Variable DST kopiert.
0	Freigabeeingang EN hat Signalzustand "0" oder Datentypen passen nicht.	Die Anweisung hat keine Daten kopiert.

8.6.9.3 CountOfElements (Anzahl ARRAY-Elemente abfragen)

Mit Anweisung "Anzahl ARRAY-Elemente abfragen" kann abgefragt werden, wie viele Array-Elemente in einer Variablen vorhanden sind, auf die ein Variant-Element zeigt.

Bei einem eindimensionalen ARRAY gibt die Anweisung die Differenz zwischen dem oberen und unteren Grenzwert +1 aus. Bei einem multidimensionalen ARRAY gibt die Anweisung das Produkt aller Dimensionen zurück.

Tabelle 8-97 Anweisung CountOfElements

KOP/FUP	SCL	Beschreibung
CountOfElements - EN - IN	<pre>Result := CountOfElements(ENvariant_in_); RET_VAL</pre>	Zählt die Anzahl der Array-Elemente im Array, auf das Parameter IN zeigt.

Hinweis

Zeigt Variant auf ein Array of Bool, zählt die Anweisung die Füllelemente bis zur nächsten Bytegrenze. Für Array[0..1] of Bool wird beispielsweise ein Wert 8 zurückgegeben.

Tabelle 8-98 Parameter für die CountOfElements-Anweisung

Parameter	Datentyp	Beschreibung
IN	Variant	Variable mit den Array-Elementen, die gezählt werden sollen
RET_VAL	UDint	Anweisungsergebnis

Tabelle 8-99 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Die Anweisung gibt die Anzahl der Array-Elemente zurück.
0	Freigabeeingang EN hat Signalzustand "0" oder Variant zeigt nicht auf ein Array.	Die Anweisung gibt 0 zurück.

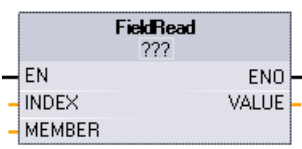

8.6.10 Anweisungen in älteren Systemen

8.6.10.1 Anweisungen FieldRead (Feld lesen) und FieldWrite (Feld schreiben)

Hinweis

In STEP 7 V10.5 wurden eine Variablenreferenz als Array-Index oder multidimensionale Arrays **nicht unterstützt**. Mit den Anweisungen FeldLesen und FeldSchreiben wurden variable Array-Index-Anweisungen für ein eindimensionales Array bereitgestellt. Ab STEP 7 V11 **werden** eine Variablenreferenz als Array-Index und multidimensionale Arrays **unterstützt**. FeldLesen und FeldSchreiben sind ab STEP 7 V11 enthalten und sind abwärts kompatibel für Programme, die diese Anweisungen umfassen.

Tabelle 8-100 Anweisungen FeldLesen und FeldSchreiben

KOP/FUP	SCL	Beschreibung
	<code>value := member[index];</code>	FeldLesen liest das Array-Element mit dem Indexwert INDEX aus dem Array, dessen erstes Element vom Parameter MEMBER angegeben wird. Der Wert des Array-Elements wird an die vom Parameter VALUE angegebene Adresse übertragen.
	<code>member[index] := value;</code>	FeldSchreiben überträgt den Wert an der vom Parameter VALUE angegebenen Adresse zu dem Array, dessen erstes Element vom Parameter MEMBER angegeben wird. Der Wert wird zu dem Array-Element übertragen, dessen Array-Index vom Parameter INDEX angegeben wird.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-101 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
Index	Eingang	DInt	Die Indexnummer des Array-Elements, das gelesen oder in das geschrieben werden soll.
Member ¹	Eingang	Binärzahlen, Ganzzahlen, Gleitpunktzahlen, Zeiten, DATE, TOD, CHAR und WCHAR als Komponenten einer ARRAY-Variablen	Adresse des ersten Elements in einem eindimensionalen Array, definiert in einem globalen Datenbaustein oder in einer Bausteinschnittstelle. Beispiel: Wird der Array-Index angegeben als [-2..4], dann ist der Index des ersten Elements -2 und nicht 0.
Value ¹	Ausgang	Binärzahlen, Ganzzahlen, Gleitpunktzahlen, Zeiten, DATE, TOD, CHAR, WCHAR	Adresse, an die das angegebene Array-Element kopiert wird (FeldLesen). Adresse des Werts, der in das angegebene Array-Element kopiert wird (FeldSchreiben).

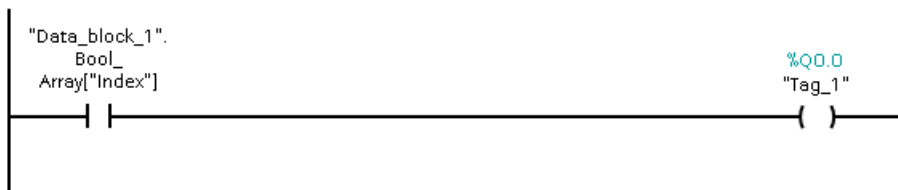
¹ Der Datentyp des vom Parameter MEMBER angegebenen Array-Elements und der Parameter VALUE müssen denselben Datentyp haben.

Der Freigabeausgang ENO ist gleich 0, wenn eine der folgenden Bedingungen zutrifft:

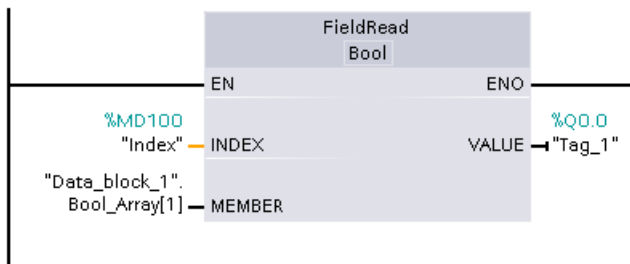
- Der Eingang EN ist im Signalzustand 0.
- Das vom Parameter INDEX angegebene Array-Element ist im vom Parameter MEMBER angegebenen Array nicht definiert.
- Während der Verarbeitung treten Fehler auf, z. B. Überlauf.

Beispiel: Zugriff auf Daten über Array-Indexierung

Um mit einer Variable auf Elemente eines Arrays zuzugreifen, verwenden Sie einfach die Variable als Array-Index in Ihrer Programmlogik. Im folgenden Netzwerk beispielsweise wird basierend auf dem Booleschen Wert eines Arrays aus Booleschen Werten ein Ausgang in "Datenbaustein_1", der von der PLC-Variablen "Index" angegeben wird, gesetzt.



Die Logik mit dem Variablen-Array-Index entspricht dem bisherigen Verfahren über die Anweisung FeldLesen:



Die Anweisungen FeldLesen und FeldSchreiben können durch Indexierungslogik mit Variablen-Arrays ersetzt werden.

In SCL gibt es keine Anweisungen FeldLesen oder FeldSchreiben, es wird jedoch die indirekte Adressierung eines Arrays mit einer Variablen unterstützt:

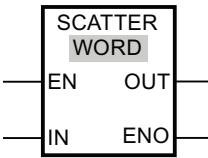
```
#Tag_1 := "Data_block_1".Bool_Array[#Index];
```

8.6.11 SCATTER

SCATTER: Bitfolge in einzelne Bits auflösen

Die Anweisung zum Zerlegen der Bitfolge in einzelne Bits unterteilt eine Variable vom Datentyp BYTE, WORD oder DWORD in einzelne Bits und speichert diese in einem Datentyp ARRAY aus BOOL, Datentyp anonyme STRUCT oder Datentyp PLC ausschließlich mit Booleschen Elementen.

Tabelle 8-102 SCATTER

KOP/FUP	SCL	Beschreibung
	<pre>SCATTER (IN := #SourceWord, OUT => #DestinationArray);</pre>	<p>SCATTER: Die Anweisung zum Zerlegen der Bitfolge in einzelne Bits unterteilt eine Variable vom Datentyp BYTE, WORD oder DWORD in einzelne Bits und speichert diese in einem Datentyp ARRAY aus BOOL, Datentyp anonyme STRUCT oder Datentyp PLC ausschließlich mit Booleschen Elementen.</p>

Hinweis

Multidimensionales ARRAY aus BOOL

Bei der Anweisung zum Zerlegen einer Bitfolge in einzelne Bits ist die Verwendung eines multidimensionalen ARRAY aus BOOL nicht zulässig.

Hinweis

Länge des Datentyps ARRAY, STRUCT oder PLC

Der Datentyp ARRAY, anonyme STRUCT oder PLC muss genau die Anzahl von Elementen haben, die in der Bitfolge angegeben ist. Das bedeutet beim Datentyp BYTE beispielsweise, dass der Datentyp ARRAY, STRUCT oder PLC genau 8 Elemente haben muss (WORD = 16 und DWORD = 32).

Hinweis

Hinweis zur Verfügbarkeit der Anweisung

Die Anweisung kann mit einer CPU der Reihe S7-1200 ab Firmwareversion 4.2 und mit einer CPU der Reihe S7-1500 mit Firmwareversion 2.1 verwendet werden.

Damit können Sie beispielsweise ein Statuswort analysieren und den Status der einzelnen Bits mithilfe des Index lesen und ändern. Mit GATHER können Sie die Bits erneut zu einer Bitfolge zusammenfügen.

Der Freigabeausgang ENO gibt den Signalzustand "0" zurück, wenn eine der folgenden Bedingungen zutrifft:

- Der Freigabeeingang EN hat den Signalzustand "0".
- Der Datentyp ARRAY, STRUCT oder PLC stellt nicht genügend BOOL-Elemente bereit.

Datentypen für die Anweisung SCATTER

Die folgende Tabelle zeigt die Parameter der Anweisung:

Parameter	Deklaration	Datentyp		Speicherbereich	Beschreibung
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	E, A, M, D, L oder Konstante	Freigabeeingang
ENO	Output	BOOL	BOOL	E, A, M, D, L	Freigabeausgang
IN	Input	BYTE, WORD, DWORD	BYTE, WORD, DWORD	E, A, M, D, L	Bitfolge, die analysiert wird. Die Werte dürfen sich nicht im E/A-Bereich oder im DB eines Technologieobjekts befinden.
OUT	Output	Datentyp ARRAY[*] aus BOOL, STRUCT oder PLC *: 8, 16, 32 oder 64 Elemente	Datentyp ARRAY[*] aus BOOL, STRUCT oder PLC *: 8, 16, 32 oder 64 Elemente	E, A, M, D, L	Datentyp ARRAY, STRUCT oder PLC, in dem die einzelnen Bits gespeichert werden

Weitere Informationen zu gültigen Datentypen finden Sie unter "Siehe auch".

Beispiel mit einem ARRAY

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceWord		WORD
EnableOut	Ausgang	BOOL
DestinationArray		ARRAY[0..15] aus BOOL

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:

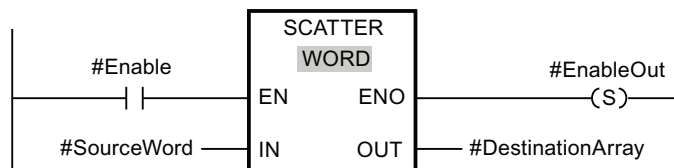


Bild 8-1 SCATTER - Beispiel 2

Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceWord	WORD (16 Bits)
OUT	DestinationUDT	Der Operand "DestinationUDT" hat den Datentyp PLC (UDT). Er besteht aus 16 Elementen und ist somit genauso groß wie der Datentyp WORD, der analysiert werden soll.

Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Der Operand #SourceWord des Datentyps WORD wird in einzelne Bits (16) aufgelöst und den einzelnen Elementen des Operanden #DestinationArray zugewiesen. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Weitere Informationen und den Programmcode für das oben aufgeführte Beispiel finden Sie hier: [Beispielbibliothek für Anweisungen](#).

Beispiel mit einem Datentyp PLC (UDT)

Erstellen Sie den folgenden PLC-Datentyp "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

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceWord		WORD
EnableOut	Ausgang	BOOL
DestinationUDT		"myBits"

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:

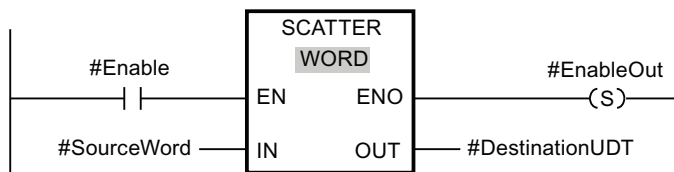


Bild 8-2 SCATTER - Beispiel

Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceWord	WORD (16 Bits)
OUT	DestinationUDT	Der Operand "DestinationUDT" hat den Datentyp PLC (UDT). Er besteht aus 16 Elementen und ist somit genauso groß wie der Datentyp WORD, der analysiert werden soll.

Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Der Operand #SourceWord des Datentyps WORD wird in einzelne Bits (16) aufgelöst und den einzelnen Elementen des Operanden #DestinationArray zugewiesen. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Siehe auch

Neue Funktionen (Seite 35)

8.6.12 SCATTER_BLK

SCATTER_BLK: Elemente eines ARRAY aus Bitfolgen in einzelne Bits auflösen

Die Anweisung Elemente eines ARRAY einer Bitfolge in einzelne Bits zerlegen unterteilt ein oder mehrere Elemente eines ARRAY aus BYTE, WORD oder DWORD in einzelne Bits und speichert diese in einem Datentyp ARRAY aus BOOL, Datentyp anonyme STRUCT oder Datentyp PLC ausschließlich mit Booleschen Elementen. Am Parameter COUNT_IN geben Sie an, wie viele Elemente des Quell-ARRAY analysiert werden sollen. Das Quell-ARRAY am Parameter IN kann mehr Elemente haben, als am Parameter COUNT_IN angegeben sind. Der Datentyp ARRAY aus BOOL, anonyme STRUCT oder PLC muss genügend Elemente haben, um die Bits der analysierten Bitfolgen zu speichern. Der Zielspeicherbereich kann jedoch auch größer sein.

Tabelle 8-103 SCATTER_BLK

KOP/FUP	SCL	Beschreibung
	<pre>SCATTER_BLK(IN:=_ byte_in_, COUNT_IN:=_uin t_in_, OUT=>_bool_out); IN:=_uint_in_ ,</pre>	<p>Die Anweisung Elemente eines ARRAY einer Bitfolge in einzelne Bits zerlegen unterteilt ein oder mehrere Elemente eines ARRAY aus BYTE, WORD oder DWORD in einzelne Bits und speichert diese in einem Datentyp ARRAY aus BOOL, Datentyp anonyme STRUCT oder Datentyp PLC ausschließlich mit Booleschen Elementen. Am Parameter COUNT_IN geben Sie an, wie viele Elemente des Quell-ARRAY analysiert werden sollen. Das Quell-ARRAY am Parameter IN kann mehr Elemente haben, als am Parameter COUNT_IN angegeben sind. Der Datentyp ARRAY aus BOOL, anonyme STRUCT oder PLC muss genügend Elemente haben, um die Bits der analysierten Bitfolgen zu speichern. Der Zielspeicherbereich kann jedoch auch größer sein.</p>

Hinweis

Es werden KEINE Daten geschrieben, wenn ENO falsch ist

Bei der S7-1200 CPU gilt nur bei der Anweisung SCATTER_BLK, dass keine Daten in den Ausgang geschrieben werden, wenn ENO = FALSE ist.

Hinweis

Multidimensionales ARRAY aus BOOL

Wenn es sich bei dem ARRAY um ein multidimensionales ARRAY aus BOOL handelt, werden bei der Zählung auch die Füllbits der Dimensionen berücksichtigt, auch wenn sie nicht explizit deklariert wurden.

Beispiel 1: Ein ARRAY[1..10,0..4,1..2] aus BOOL wird wie ein ARRAY[1..10,0..4,1..8] aus BOOL oder wie ein ARRAY[0..399] aus BOOL gehandhabt.

Beispiel 2: Am Parameter IN wird ein ARRAY[0..5] aus WORD (sourceArrayWord[2]) verschaltet. Der Parameter COUNT_IN hat den Wert "3". Am Parameter OUT wird ein ARRAY[0..1,0..5,0..7] aus BOOL (destinationArrayBool[0,0,0]) verschaltet. Sowohl das Array am Parameter IN als auch das am Parameter OUT haben eine Größe von 96 Bit. Das ARRAY aus WORD wird in 48 einzelne Bits aufgelöst.

Hinweis

Wenn die untere ARRAY-Grenze des Ziel-ARRAY nicht "0" ist, beachten Sie Folgendes:

Aus Gründen der Leistungsfähigkeit muss der Index immer an einer BYTE-, WORD- oder DWORD-Grenze beginnen. Das bedeutet, dass die Berechnung des Index an der unteren Grenze des ARRAY beginnen muss. Als Grundlage für diese Berechnung wird die folgende Formel verwendet:

Gültige Indizes = untere ARRAY-Grenze + $n(\text{Anzahl von Bitfolgen}) \times \text{Anzahl von Bits der gewünschten Bitfolge}$

Bei einem ARRAY[-2..45] aus BOOL und der Bitfolge WORD sieht die Berechnung wie folgt aus:

- Gültiger Index (-2) = $-2 + 0 \times 16$
- Gültiger Index (14) = $-2 + 1 \times 16$
- Gültiger Index (30) = $-2 + 2 \times 16$

Ein Beispiel ist unten beschrieben.

Hinweis

Verfügbarkeit der Anweisung

Die Anweisung kann mit einer CPU der Reihe S7-1200 ab Firmwareversion 4.2 und mit einer CPU der Reihe S7-1500 mit Firmwareversion 2.1 verwendet werden.

Dann können Sie beispielsweise Statuswörter analysieren und den Status der einzelnen Bits mithilfe des Index lesen und ändern. Mit GATHER können Sie die Bits erneut zu einer Bitfolge zusammenfügen.

Der Freigabeausgang ENO gibt den Signalzustand "0" zurück, wenn eine der folgenden Bedingungen zutrifft:

- Der Freigabeeingang EN hat den Signalzustand "0".
- Das Quell-ARRAY hat weniger Elemente als am Parameter COUNT_IN angegeben sind.
- Der Index des Ziel-ARRAY beginnt nicht an einer BYTE-, WORD- oder DWORD-Grenze. In diesem Fall wird kein Ergebnis in das ARRAY aus BOOL geschrieben.
- Der Datentyp ARRAY[*] aus BOOL, STRUCT oder PLC verfügt nicht über die erforderliche Anzahl von Elementen.
 - S7-1500 CPU: In diesem Fall werden so viele Bitfolgen wie möglich analysiert und in den Datentyp ARRAY aus BOOL, anonyme STRUCT oder PLC geschrieben. Die übrigen Bitfolgen werden nicht mehr berücksichtigt.
 - S7-1200 CPU: Es erfolgt kein Kopiervorgang.

Datentypen für die Anweisung SCATTER_BLK

Die folgende Tabelle zeigt die Parameter der Anweisung:

Parameter	Deklaration	Datentyp		Speicherbereich	Beschreibung
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	E, A, M, D, L oder Konstante	Freigabeeingang
ENO	Output	BOOL	BOOL	E, A, M, D, L	Freigabeausgang
IN	Input	Element eines ARRAY[*] aus <Bitfolge>	Element eines ARRAY[*] aus <Bitfolge>	E, A, M, D, L	ARRAY aus <Bitfolge>, die analysiert wird. Die Werte dürfen sich nicht im E/A-Bereich oder im DB eines Technologieobjekts befinden.
COUNT_IN	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	E, A, M, D, L	Zähler für die Anzahl von Elementen des Quell-ARRAY, die analysiert werden. Der Wert darf sich nicht im E/A-Bereich oder in der Datenbank eines Technologieobjekts befinden.
OUT	Output	Element eines Datentyps ARRAY[*] aus BOOL, STRUCT oder PLC	Element eines Datentyps ARRAY[*] aus BOOL, STRUCT oder PLC	E, A, M, D, L	Datentyp ARRAY, STRUCT oder PLC, in dem die einzelnen Bits gespeichert werden

Sie können die erforderliche Bitfolge aus der Klappliste "???" des Anweisungsfelds auswählen.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Siehe auch".

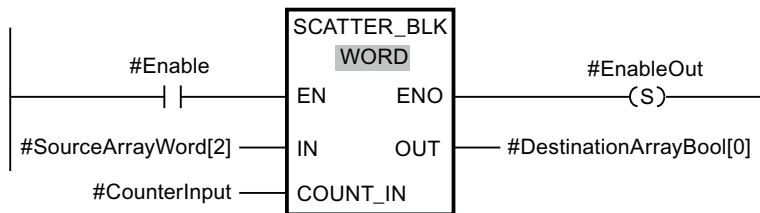
Beispiel für ein Ziel-ARRAY mit der unteren Grenze "0"

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceArrayWord		ARRAY[0..5] aus WORD
CounterInput		UDINT

Variable	Bereich	Datentyp
EnableOut	Ausgang	BOOL
DestinationArrayBool		ARRAY[0..95] aus BOOL

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceArrayWord[2]	ARRAY[0..5] aus WORD (96 Bits können analysiert werden)
COUNT_IN	CounterInput = 3	UDINT3 (3 x WORD oder 48 Bits sollen analysiert werden. Das bedeutet, dass mindestens 48 Bits im Ziel-ARRAY verfügbar sein müssen.)
OUT	DestinationArrayBool[0]	Der Operand "DestinationArrayBool" ist vom Datentyp ARRAY[0..95] aus BOOL. Das bedeutet, dass er 96 BOOL-Elemente bereitstellt.

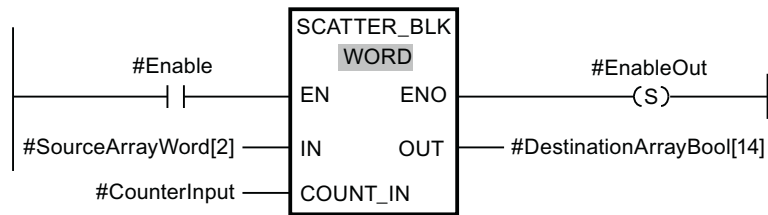
Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Das dritte, vierte und fünfte WORD des Operanden #SourceArrayWord wird in seine einzelnen Bits (48) aufgelöst und ab dem ersten Element den einzelnen Elementen des Operanden #DestinationArrayBool zugewiesen. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Beispiel für ein Ziel-ARRAY mit der unteren Grenze "-2"

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceArrayWord		ARRAY[0..5] aus WORD
CounterInput		UDINT
EnableOut	Ausgang	BOOL
DestinationArrayBool		ARRAY[-2..93] aus BOOL

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceArrayWord[2]	ARRAY[0..5] aus WORD (96 Bits können analysiert werden)
COUNT_IN	CounterInput = 3	UDINT3 (3 x WORD oder 48 Bits sollen analysiert werden. Das bedeutet, dass mindestens 48 Bits im Ziel-ARRAY verfügbar sein müssen.)
OUT	DestinationArrayBool[14]	Der Operand "DestinationArrayBool" ist vom Datentyp ARRAY[-2..93] aus BOOL. Das bedeutet, dass er 96 BOOL-Elemente bereitstellt.

Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Das dritte, vierte und fünfte WORD des Operanden #SourceArrayWord wird in seine einzelnen Bits (48) aufgelöst und ab dem 16. Element den einzelnen Elementen des Operanden #DestinationArrayBool zugewiesen. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück. Die übrigen 32 Bits werden nicht geschrieben.

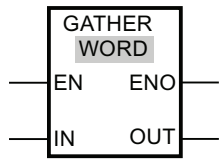
Weitere Informationen und den Programmcode für das oben aufgeführte Beispiel finden Sie hier: [Beispielbibliothek für Anweisungen](#).

8.6.13 GATHER

GATHER

Die Anweisung Einzelne Bits zu mehreren Elementen eines ARRAY einer Bitfolge zusammenfügen verbindet die Bits eines Datentyps ARRAY aus BOOL, Datentyps anonyme STRUCT oder Datentyps PLC ausschließlich mit Booleschen Elementen zu einer Bitfolge. Die Bitfolge wird in einer Variablen vom Datentyp BYTE, WORD, DWORD oder LWORD gespeichert.

Tabelle 8-104 GATHER

KOP/FUP	SCL	Beschreibung
	<pre>GATHER(IN := #SourceArray, OUT => #DestinationArray);</pre>	<p>GATHER: Die Anweisung Einzelne Bits zu mehreren Elementen eines ARRAY einer Bitfolge zusammenfügen verbindet die Bits eines Datentyps ARRAY aus BOOL, Datentyps anonyme STRUCT oder Datentyps PLC ausschließlich mit Booleschen Elementen zu einer Bitfolge. Die Bitfolge wird in einer Variablen vom Datentyp BYTE, WORD oder DWORD gespeichert.</p>

Hinweis

Multidimensionales ARRAY aus BOOL

Bei der Anweisung Einzelne Bits zu mehreren Elementen eines ARRAY einer Bitfolge zusammenfügen ist die Verwendung eines multidimensionalen ARRAY aus BOOL nicht zulässig.

Hinweis

Länge des Datentyps ARRAY, STRUCT oder PLC

Der Datentyp ARRAY, STRUCT oder PLC muss genau die Anzahl von Elementen haben, die in der Bitfolge angegeben ist.

Das bedeutet beim Datentyp BYTE beispielsweise, dass der Datentyp ARRAY, anonyme STRUCT oder PLC genau 8 Elemente haben muss (WORD = 16 und DWORD = 32).

Hinweis

Verfügbarkeit der Anweisung

Die Anweisung kann mit einer CPU der Reihe S7-1200 ab Firmwareversion 4.2 und mit einer CPU der Reihe S7-1500 mit Firmwareversion 2.1 verwendet werden.

Der Freigabeausgang ENO gibt den Signalzustand "0" zurück, wenn eine der folgenden Bedingungen zutrifft:

- Der Freigabeeingang EN hat den Signalzustand "0".
- Der Datentyp ARRAY, anonyme STRUCT oder PLC (UDT) hat weniger oder mehr BOOL-Elemente als von der Bitfolge angegeben. In diesem Fall werden die BOOL-Elemente nicht übertragen.
- Weniger als die benötigte Anzahl von Bits sind verfügbar.

Datentypen für die Anweisung GATHER

Die folgende Tabelle zeigt die Parameter der Anweisung:

Parameter	Deklaration	Datentyp		Speicherbereich	Beschreibung
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	E, A, M, D, L oder Konstante	Freigabeeingang
ENO	Output	BOOL	BOOL	E, A, M, D, L	Freigabeausgang
IN	Input	Datentyp ARRAY[*] aus BOOL, STRUCT oder PLC *: 8, 16, 32 oder 64 Elemente	Datentyp ARRAY[*] aus BOOL, STRUCT oder PLC *: 8, 16, 32 oder 64 Elemente	E, A, M, D, L	Datentyp ARRAY, STRUCT oder PLC, dessen Bits zu einer Bitfolge zusammengeführt werden. Die Werte dürfen sich nicht im E/A-Bereich oder im DB eines Technologieobjekts befinden.
OUT	Output	BYTE, WORD, DWORD	BYTE, WORD, DWORD	E, A, M, D, L	Zusammengeführte Bitfolge, gespeichert in einer Variablen

Sie können die erforderliche Bitfolge aus der Klappliste "???" des Anweisungsfelds auswählen.

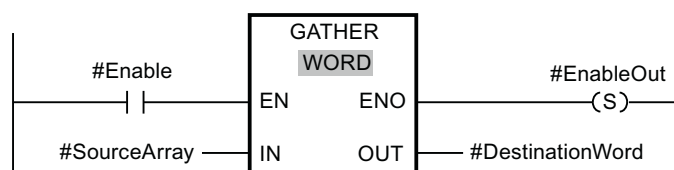
Weitere Informationen zu gültigen Datentypen finden Sie unter "Siehe auch".

Beispiel mit einem ARRAY

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceArray		ARRAY[0..15] aus BOOL
EnableOut	Ausgang	BOOL
DestinationWord		WORD

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceArray	Der Operand "SourceArray" ist vom Datentyp ARRAY[0..15] aus BOOL. Er besteht aus 16 Elementen und ist somit genauso groß wie der Datentyp WORD, zu dem die Bits zusammengeführt werden sollen.
OUT	DestinationWord	WORD (16 Bits)

Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Die Bits des Operanden #SourceArray werden zu einem WORD zusammengeführt. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Weitere Informationen und den Programmcode für das oben aufgeführte Beispiel finden Sie hier: Beispielbibliothek für Anweisungen.

Beispiel mit einem Datentyp PLC (UDT)

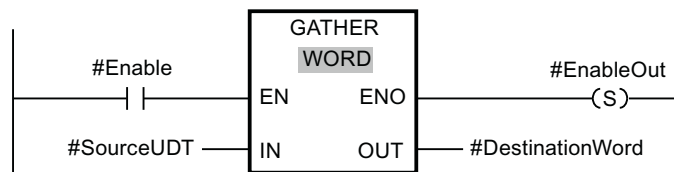
Erstellen Sie den folgenden PLC-Datentyp "myBits":

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

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceUDT		"myBits"
EnableOut	Ausgang	BOOL
DestinationWord		WORD

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceUDT	Der Operand "SourceUDT" hat den Datentyp PLC (UDT). Er besteht aus 16 Elementen und ist somit genauso groß wie der Datentyp WORD, zu dem die Bits zusammengeführt werden sollen.
OUT	DestinationWord	WORD (16 Bits)

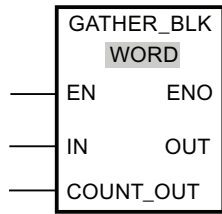
Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Die Bits des Operanden #SourceUDT werden zu einem WORD zusammengeführt. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

8.6.14 GATHER_BLK

Beschreibung

Die Anweisung Einzelne Bits zu mehreren Elementen eines ARRAY einer Bitfolge zusammenfügen verbindet die Bits eines Datentyps ARRAY aus BOOL, Datentyps anonyme STRUCT oder Datentyps PLC ausschließlich mit Booleschen Elementen in einem oder mehreren Elementen eines ARRAY aus <Bitfolge>. Am Parameter COUNT_OUT geben Sie an, wie viele Elemente des Ziel-ARRAY geschrieben werden sollen. Mit diesem Schritt geben Sie implizit auch an, wie viele Elemente des Datentyps ARRAY aus BOOL, anonyme STRUCT oder PLC erforderlich sind. Das Ziel-ARRAY am Parameter OUT kann mehr Elemente haben als am Parameter COUNT_OUT angegeben sind. Das ARRAY aus <Bitfolge> muss genügend Elemente haben, um die Bits, die zusammengeführt werden sollen, zu speichern. Das Ziel-ARRAY kann jedoch auch größer sein.

Tabelle 8-105 GATHER

KOP/FUP	SCL	Beschreibung
	<pre>GATHER_BLK (IN := #SourceArrayBool[0], COUNT_OUT := #CounterOutput, OUT => #DestinationArrayWord[2]);</pre>	<p>Die Anweisung Einzelne Bits zu mehreren Elementen eines ARRAY einer Bitfolge zusammenfügen verbindet die Bits eines Datentyps ARRAY aus BOOL, Datentyps anonyme STRUCT oder Datentyps PLC ausschließlich mit Booleschen Elementen in einem oder mehreren Elementen eines ARRAY aus <Bitfolge>. Am Parameter COUNT_OUT geben Sie an, wie viele Elemente des Ziel-ARRAY geschrieben werden sollen. Mit diesem Schritt geben Sie implizit auch an, wie viele Elemente des Datentyps ARRAY aus BOOL, anonyme STRUCT oder PLC erforderlich sind. Das Ziel-ARRAY am Parameter OUT kann mehr Elemente haben als am Parameter COUNT_OUT angegeben sind. Das ARRAY aus <Bitfolge> muss genügend Elemente haben, um die Bits, die zusammengeführt werden sollen, zu speichern. Das Ziel-ARRAY kann jedoch auch größer sein.</p>

Hinweis

Es werden KEINE Daten geschrieben, wenn ENO falsch ist

Bei der S7-1200 CPU gilt nur bei der Anweisung GATHER_BLK, dass keine Daten in den Ausgang geschrieben werden, wenn ENO = FALSE ist.

Hinweis**Multidimensionales ARRAY aus BOOL**

Wenn es sich bei dem ARRAY um ein multidimensionales ARRAY aus BOOL handelt, werden bei der Zählung auch die Füllbits der Dimensionen berücksichtigt, auch wenn sie nicht explizit deklariert wurden.

Beispiel 1: Ein ARRAY[1..10,0..4,1..2] aus BOOL wird wie ein ARRAY[1..10,0..4,1..8] aus BOOL oder wie ein ARRAY[0..399] aus BOOL gehandhabt.

Beispiel 2: Am Parameter OUT wird ein ARRAY[0..5] aus WORD (sourceArrayWord[2]) verschaltet. Der Parameter COUNT_IN hat den Wert "3". Am Parameter IN wird ein ARRAY[0..1,0..5,0..7] aus BOOL (destinationArrayBool[0,0,0]) verschaltet. Sowohl das Array am Parameter IN als auch das am Parameter OUT haben eine Größe von 96 Bit. 48 einzelne Bits werden aus dem ARRAY aus BOOL zusammengeführt.

Hinweis**Wenn die untere ARRAY-Grenze des Quell-ARRAY nicht "0" ist, beachten Sie Folgendes:**

Aus Gründen der Leistungsfähigkeit muss der Index immer an einer BYTE-, WORD- oder DWORD-Grenze beginnen. Das bedeutet, dass die Berechnung des Index an der unteren Grenze des ARRAY beginnen muss. Als Grundlage für diese Berechnung wird die folgende Formel verwendet:

Gültige Indizes = untere ARRAY-Grenze + n(Anzahl von Bitfolgen) × Anzahl von Bits der gewünschten Bitfolge

Bei einem ARRAY[-2..45] aus BOOL und der Bitfolge WORD sieht die Berechnung wie folgt aus:

- Gültiger Index (-2) = -2 + 0 × 16
- Gültiger Index (14) = -2 + 1 × 16
- Gültiger Index (30) = -2 + 2 × 16

Ein Beispiel ist unten beschrieben.

Hinweis**Verfügbarkeit der Anweisung**

Die Anweisung kann mit einer CPU der Reihe S7-1200 ab Firmwareversion 4.2 und mit einer CPU der Reihe S7-1500 mit Firmwareversion 2.1 verwendet werden.

Der Freigabeausgang ENO gibt den Signalzustand "0" zurück, wenn eine der folgenden Bedingungen zutrifft:

- Der Freigabeeingang EN hat den Signalzustand "0".
- Der Index des Quell-ARRAY beginnt nicht an einer BYTE-, WORD- oder DWORD-Grenze. In diesem Fall wird kein Ergebnis in das ARRAY aus <Bitfolge> geschrieben.
- Der Datentyp ARRAY[*] aus <Bitfolge> verfügt nicht über die erforderliche Anzahl von Elementen.
 - S7-1500 CPU: In diesem Fall werden so viele Bitfolgen wie möglich zusammengeführt und in den Datentyp ARRAY aus <Bitfolge> geschrieben. Die übrigen Bits werden nicht mehr berücksichtigt.
 - S7-1200 CPU: Es erfolgt kein Kopiervorgang.

Datentypen für die Anweisung GATHER_BLK

Die folgende Tabelle zeigt die Parameter der Anweisung:

Parameter	Deklaration	Datentyp		Speicherbereich	Beschreibung
		S7-1200	S7-1500		
EN	Input	BOOL	BOOL	E, A, M, D, L oder Konstante	Freigabeeingang
ENO	Output	BOOL	BOOL	E, A, M, D, L	Freigabeausgang
IN	Input	Element eines Datentyps ARRAY[*] aus BOOL, STRUCT oder PLC	Element eines Datentyps ARRAY[*] aus BOOL, STRUCT oder PLC	E, A, M, D, L	Datentyp ARRAY aus BOOL, STRUCT oder PLC, dessen Bits zusammengeführt werden (Quell-ARRAY) Die Werte dürfen sich nicht im E/A-Bereich oder im DB eines Technologieobjekts befinden.
COUNT_OUT	Input	USINT, UINT, UDINT	USINT, UINT, UDINT	E, A, M, D, L	Zähler, wie viele Elemente des Ziel-ARRAY beschrieben werden sollen. Der Wert darf sich nicht im E/A-Bereich oder in der Datenbank eines Technologieobjekts befinden.
OUT	Output	Element eines ARRAY[*] aus <Bitfolge>	Element eines ARRAY[*] aus <Bitfolge>	E, A, M, D, L	ARRAY aus <Bitfolge>, in dem die Bits gespeichert werden (Ziel-ARRAY)

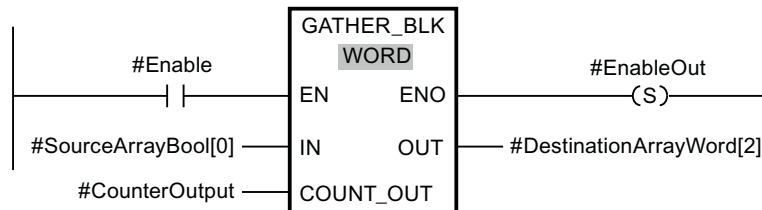
Sie können die erforderliche Bitfolge aus der Klappliste "???" des Anweisungsfelds auswählen. Weitere Informationen zu gültigen Datentypen finden Sie unter "Siehe auch".

Beispiel für ein Quell-ARRAY mit der unteren Grenze "0"

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceArrayBool		ARRAY[0..95] aus BOOL
CounterOutput		UDINT
EnableOut	Ausgang	BOOL
DestinationArrayWord		ARRAY[0..5] aus WORD

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceArrayBool[0]	Der Operand "SourceArrayBool" gehört zum Datentyp ARRAY[0..95] aus BOOL. Das bedeutet, dass er 96 BOOL-Elemente bereitstellt, die wieder zu Wörtern zusammengeführt werden können.
COUNT_OUT	CounterOutput = 3	UDINT3 (3 Wörter sollen geschrieben werden. Das bedeutet, dass 48 Bits im Quell-ARRAY verfügbar sein müssen.)
OUT	DestinationArrayWord[2]	Der Operand "DestinationArrayWord" gehört zum Datentyp ARRAY[0..5] aus WORD. Das bedeutet, dass 6 WORD-Elemente verfügbar sind.

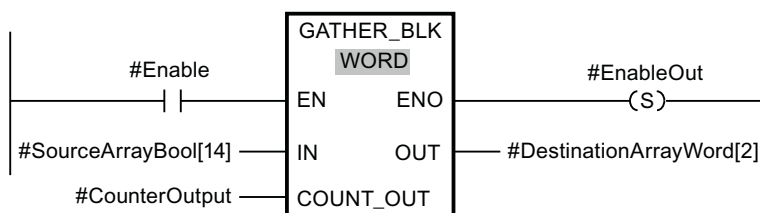
Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Ab dem ersten Element des Operanden #SourceArrayBool werden 48 Bits im Operand #DestinationArrayWord zusammengeführt. Der Anfangspunkt im Ziel-ARRAY ist das dritte Element. Das bedeutet, dass die ersten 16 Bits in das dritte Wort, die zweiten 16 Bits in das vierte Wort und die dritten 16 Bits in das fünfte Wort im Ziel-ARRAY geschrieben werden. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Beispiel für ein Quell-ARRAY mit der unteren Grenze "-2"

Erstellen Sie die folgenden Variablen in der Bausteinschnittstelle:

Variable	Bereich	Datentyp
Enable	Eingang	BOOL
SourceArrayBool		ARRAY[-2..93] aus BOOL
CounterOutput		UDINT
EnableOut	Ausgang	BOOL
DestinationArrayWord		ARRAY[0..5] aus WORD

Das folgende Beispiel zeigt, wie die Anweisung funktioniert:



Die folgende Tabelle zeigt, wie die Anweisung mit spezifischen Operandenwerten funktioniert:

Parameter	Operand	Datentyp
IN	SourceArrayBool[14]	Der Operand "SourceArrayBool" ist vom Datentyp ARRAY[-2..93] aus BOOL. Da der Anfangspunkt das 16. Element ist, sind nur 80 BOOL-Elemente verfügbar, die wieder zu Wörtern zusammengeführt werden können.
COUNT_OUT	CounterOutput = 3	UDINT3 (3 Wörter sollen geschrieben werden. Das bedeutet, dass 48 Bits im Quell-ARRAY verfügbar sein müssen.)
OUT	DestinationArrayWord[2]	Der Operand "DestinationArrayWord" gehört zum Datentyp ARRAY[0..5] aus WORD. Das bedeutet, dass 6 WORD-Elemente verfügbar sind.

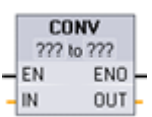
Wenn der Operand #Enable am Freigabeeingang EN den Signalzustand "1" zurückgibt, wird die Anweisung ausgeführt. Ab dem 16. Element des Operanden #SourceArrayBool werden 48 Bits zum Operand #DestinationArrayWord zusammengeführt. Der Anfangspunkt im Ziel-ARRAY ist das dritte Element. Das bedeutet, dass die ersten 16 Bits des Quell-ARRAY ignoriert werden. Die zweiten 16 Bits werden in das dritte Wort, die dritten 16 Bits in das vierte Wort und die vierten 16 Bits in das fünfte Wort im Ziel-ARRAY geschrieben. Die übrigen 64 Bits des Quell-ARRAY werden ebenfalls nicht berücksichtigt. Tritt während der Ausführung der Anweisung ein Fehler auf, gibt der Operand #EnableOut am Freigabeausgang ENO den Signalzustand "0" zurück.

Weitere Informationen und den Programmcode für das oben aufgeführte Beispiel finden Sie hier: Beispielbibliothek für Anweisungen.

8.7 Umwandlungsoperationen

8.7.1 CONV (Wert umwandeln)

Tabelle 8-106 Anweisung Umwandeln (CONV)

KOP/FUP	SCL	Beschreibung
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	Konvertiert ein Datenelement von einem Datentyp in einen anderen Datentyp.

- ¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.
- ² In SCL: Konstruieren Sie die Umwandlungsanweisung durch Angeben des Datentyps für den Eingangsparameter (in) und den Ausgangsparameter (out). Beispiel: DWORD_TO_REAL konvertiert einen Doppelwortwert in einen Realzahlenwert.

Tabelle 8-107 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Bitzeichenkette ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Eingangswert
OUT	Bitzeichenkette ¹ , SInt, USInt, Int, UInt, DInt, UDInt, Real, LReal, BCD16, BCD32, Char, WChar	Eingangswert, umgewandelt in einen neuen Datentyp

- ¹ Für die Anweisung dürfen keine Bitzeichenketten (Byte, Word, DWord) ausgewählt werden. Um für einen Parameter der Anweisung einen Operanden vom Datentyp Byte, Word oder DWord einzugeben, wählen Sie eine vorzeichenlose Ganzzahl mit der gleichen Bitlänge aus. Beispiel: Für ein Byte wählen Sie USInt, für ein Word wählen Sie UInt und für DWord wählen Sie UDInt.

Nach Auswahl des Quelldatentyps (Konvertieren aus) wird in der Klappliste eine Reihe möglicher Umwandlungen (Konvertieren in) angezeigt. Umwandlungen von und in BCD16 sind auf den Datentyp Int begrenzt. Umwandlungen von und in BCD32 sind auf den Datentyp DInt begrenzt.

Tabelle 8-108 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN
0	Ergebnis überschreitet den gültigen Bereich für den Datentyp von OUT	OUT wird auf den Wert von IN gesetzt

8.7.2 Umwandlungsanweisungen in SCL

Umwandlungsanweisungen in SCL

Tabelle 8-109 Umwandlung von Bool, Byte, Word oder DWord

Datentyp	Anweisung	Ergebnis
Bool	BOOL_TO_BYTE, BOOL_TO_WORD, BOOL_TO_DWORD, BOOL_TO_INT, BOOL_TO_DINT	Der Wert wird in das niederwertigste Bit des Zieldatentyps übertragen.
Byte	BYTE_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	BYTE_TO_WORD, BYTE_TO_DWORD	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
	BYTE_TO_SINT, BYTE_TO_USINT	Der Wert wird in den Zieldatentyp übertragen.
	BYTE_TO_INT, BYTE_TO_UINT, BYTE_TO_DINT, BYTE_TO_UDINT	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
Word	WORD_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	WORD_TO_BYTE	Das niederwertigste Byte des Quellwerts wird in den Zieldatentyp übertragen.
	WORD_TO_DWORD	Der Wert wird in das niederwertigste Wort des Zieldatentyps übertragen.
	WORD_TO_SINT, WORD_TO_USINT	Das niederwertigste Byte des Quellwerts wird in den Zieldatentyp übertragen.
	WORD_TO_INT, WORD_TO_UINT	Der Wert wird in den Zieldatentyp übertragen.
	WORD_TO_DINT, WORD_TO_UDINT	Der Wert wird in das niederwertigste Wort des Zieldatentyps übertragen.
DWord	DWORD_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	DWORD_TO_BYTE, DWORD_TO_WORD, DWORD_TO_SINT	Das niederwertigste Byte des Quellwerts wird in den Zieldatentyp übertragen.
	DWORD_TO_USINT, DWORD_TO_INT, DWORD_TO_UINT	Das niederwertigste Wort des Quellwerts wird in den Zieldatentyp übertragen.
	DWORD_TO_DINT, DWORD_TO_UDINT, DWORD_TO_REAL	Der Wert wird in den Zieldatentyp übertragen.

Tabelle 8-110 Umwandlung einer kurzen Ganzzahl (SInt oder USInt)

Datentyp	Anweisung	Ergebnis
SInt	SINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	SINT_TO_BYTE	Der Wert wird in den Zieldatentyp übertragen.
	SINT_TO_WORD, SINT_TO_DWORD	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
	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	Der Wert wird umgewandelt.
USInt	USINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	USINT_TO_BYTE	Der Wert wird in den Zieldatentyp übertragen.
	USINT_TO_WORD, USINT_TO_DWORD, USINT_TO_INT, USINT_TO_UINT, USINT_TO_DINT, USINT_TO_UDINT	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
	USINT_TO_SINT, USINT_TO_REAL, USINT_TO_LREAL, USINT_TO_CHAR, USINT_TO_STRING	Der Wert wird umgewandelt.

Tabelle 8-111 Umwandlung einer Ganzzahl (Int oder UInt)

Datentyp	Anweisung	Ergebnis
Int	INT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	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	Der Wert wird umgewandelt.
	INT_TO_WORD	Der Wert wird in den Zieldatentyp übertragen.
	INT_TO_DINT	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.
UInt	UINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	UINT_TO_BYTE, UINT_TO_SINT, UINT_TO_USINT, UINT_TO_INT, UINT_TO_REAL, UINT_TO_LREAL, UINT_TO_CHAR, UINT_TO_STRING	Der Wert wird umgewandelt.
	UINT_TO_WORD, UINT_TO_DATE	Der Wert wird in den Zieldatentyp übertragen.
	UINT_TO_DWORD, UINT_TO_DINT, UINT_TO_UDINT	Der Wert wird in das niederwertigste Byte des Zieldatentyps übertragen.

8.7 Umwandlungsoperationen

Tabelle 8-112 Umwandlung einer doppelten Ganzzahl (Dint oder UDInt)

Datentyp	Anweisung	Ergebnis
Dint	DINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	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	Der Wert wird umgewandelt.
	DINT_TO_DWORD, DINT_TO_TIME	Der Wert wird in den Zieldatentyp übertragen.
UDInt	UDINT_TO_BOOL	Das niederwertigste Bit wird in den Zieldatentyp übertragen.
	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	Der Wert wird umgewandelt.
	UDINT_TO_DWORD, UDINT_TO_TOD	Der Wert wird in den Zieldatentyp übertragen.

Tabelle 8-113 Umwandlung einer Realzahl (Real oder LReal)

Datentyp	Anweisung	Ergebnis
Real	REAL_TO_DWORD, REAL_TO_LREAL	Der Wert wird in den Zieldatentyp übertragen.
	REAL_TO_SINT, REAL_TO_USINT, REAL_TO_INT, REAL_TO_UINT, REAL_TO_DINT, REAL_TO_UDINT, REAL_TO_STRING	Der Wert wird umgewandelt.
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	Der Wert wird umgewandelt.

Tabelle 8-114 Umwandlung von Time, DTL, TOD oder Date

Datentyp	Anweisung	Ergebnis
Time	TIME_TO_DINT	Der Wert wird in den Zieldatentyp übertragen.
DTL	DTL_TO_DATE, DTL_TO_TOD	Der Wert wird umgewandelt.
TOD	TOD_TO_UDINT	Der Wert wird umgewandelt.
Date	DATE_TO_UINT	Der Wert wird umgewandelt.



Tabelle 8-115 Umwandlung von Char oder String

Datentyp	Anweisung	Ergebnis
Char	CHAR_TO_SINT, CHAR_TO_USINT, CHAR_TO_INT, CHAR_TO_UINT, CHAR_TO_DINT, CHAR_TO_UDINT	Der Wert wird umgewandelt.
	CHAR_TO_STRING	Der Wert wird zum ersten Zeichen der Zeichenkette übertragen.

Datentyp	Anweisung	Ergebnis
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	Der Wert wird umgewandelt.
	STRING_TO_CHAR	Das erste Zeichen der Zeichenkette wird in Char kopiert.

8.7.3 ROUND (Zahl runden) und TRUNC (Ganzzahl erzeugen)

Tabelle 8-116 Anweisungen ROUND und TRUNC

KOP/FUP	SCL	Beschreibung
	<code>out := ROUND (in);</code>	<p>konvertiert eine Realzahl in eine Ganzzahl. Klicken Sie für KOP/FUP auf das "???" im Anweisungsfeld, um den Datentyp für den Ausgang auszuwählen, zum Beispiel "DInt".</p> <p>Für SCL ist der Datentyp für den Ausgang der Anweisung ROUND standardmäßig DINT. Für die Rundung auf einen anderen Ausgangsdantentyp ist der Name der Anweisung mit dem expliziten Namen des Datentyps einzugeben, zum Beispiel ROUND_REAL oder ROUND_LREAL.</p> <p>Die Nachkommastellen der Realzahl werden auf den nächsten ganzzahligen Wert gerundet (IEEE - runden). Wenn die Zahl genau die Hälfte der Spanne zwischen zwei Ganzzahlen ist (z. B. 10,5), wird die Zahl auf die gerade Ganzzahl gerundet. Beispiel:</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12
	<code>out := TRUNC (in);</code>	TRUNC konvertiert eine Realzahl in eine Ganzzahl. Die Nachkommastellen der Realzahl werden auf Null verkürzt (IEEE - runden auf Null).

¹ In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-117 Datentypen für die Parameter

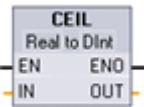

Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingang Gleitpunktzahl
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Gerundeter oder ganzzahliger Ausgang

Tabelle 8-118 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN

8.7.4 CEIL und FLOOR (Aus Gleitpunktzahl nächsthöhere und nächstniedere Ganzzahl erzeugen)

Tabelle 8-119 Anweisungen CEIL und FLOOR

KOP/FUP	SCL	Beschreibung
	<code>out := CEIL(in);</code>	Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die größer oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf +unendlich).
	<code>out := FLOOR(in);</code>	Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die kleiner oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf -unendlich).

¹ In KOP und FUP: Klicken Sie auf die "???" (beim Anweisungsnamen) und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-120 Datentypen für die Parameter

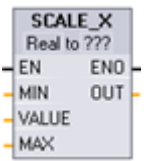
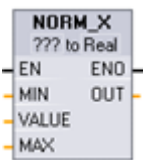
Parameter	Datentyp	Beschreibung
IN	Real, LReal	Eingang Gleitpunktzahl
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Umgewandelter Ausgang

Tabelle 8-121 ENO-Status

ENO	Beschreibung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	IN ist +/- INF oder +/- NaN	+/- INF oder +/- NaN

8.7.5 SCALE_X (Skalieren) und NORM_X (Normieren)

Tabelle 8-122 Anweisungen SCALE_X und NORM_X

KOP/FUP	SCL	Beschreibung
	<pre>out :=SCALE_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>Skaliert den normierten Realparameter VALUE ($0,0 \leq \text{VALUE} \leq 1,0$) in den mit den Parametern MIN und MAX vorgegebenen Datentyp und Wertebereich:</p> $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$
	<pre>out :=NORM_X(min:=_in_, value:=_in_, max:=_in_);</pre>	<p>Normiert den Parameter VALUE innerhalb des von den Parametern MIN und MAX angegebenen Wertebereichs:</p> $\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN}),$ <p>dabei ist ($0,0 \leq \text{OUT} \leq 1,0$)</p>

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-123 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
MIN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingang Mindestwert des Bereichs
VALUE	SCALE_X: Real, LReal NORM_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingangswert für Skalierung oder Normierung
MAX	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal	Eingang Höchstwert des Bereichs
OUT	SCALE_X: SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal NORM_X: Real, LReal	Skalierter oder normierter Ausgangswert

¹ Bei SCALE_X: Die Parameter MIN, MAX und OUT müssen denselben Datentyp haben.
Bei NORM_X: Die Parameter MIN, VALUE und MAX müssen denselben Datentyp haben.

Hinweis

Der Parameter VALUE von SCALE_X muss im Bereich (0,0 <= VALUE <= 1,0) liegen.

Falls der Parameter VALUE kleiner als 0,0 oder größer als 1,0 ist:

- Die lineare Skalierungsoperation kann Ausgabewerte OUT erzeugen, die kleiner als der Wert des Parameters MIN oder größer als der Wert des Parameters MAX sind, sofern es sich um OUT-Werte handelt, die im Wertebereich des Datentyps von OUT liegen. Bei der Ausführung von SCALE_X wird in diesen Fällen ENO = WAHR gesetzt.
- Es ist möglich, skalierte Zahlen zu erzeugen, die nicht im Bereich des Datentyps von OUT liegen. In diesem Fall wird der Wert des Parameters OUT auf einen Zwischenwert gesetzt, der dem niederwertigsten Anteil der skalierten Realzahl vor der endgültigen Umwandlung in den Datentyp OUT entspricht. Bei der Ausführung von SCALE_X wird in diesen Fällen ENO = FALSCH gesetzt.

Der Parameter VALUE von NORM_X muss im Bereich (MIN <= VALUE <= MAX) liegen.

Wenn der Parameter VALUE kleiner als MIN oder größer als MAX ist, kann die lineare Skalierungsanweisung normierte Ausgabewerte OUT erzeugen, die kleiner als 0,0 oder größer als 1,0 sind. Bei der Ausführung von NORM_X wird dann ENO = WAHR gesetzt.

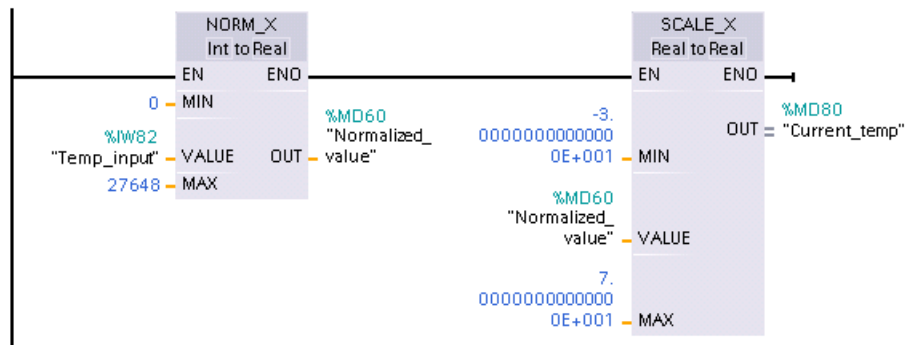
Tabelle 8-124 ENO-Status

ENO	Bedingung	Ergebnis OUT
1	Kein Fehler	Gültiges Ergebnis
0	Das Ergebnis überschreitet den gültigen Bereich für den Datentyp von OUT.	Zwischenergebnis: Der niederwertigste Anteil einer Realzahl vor der endgültigen Umwandlung in den Datentyp von OUT.
0	ParameterMAX <= MIN	SCALE_X: Der niederwertigste Anteil der Realzahl VALUE, mit dem die Größe von OUT belegt wird. NORM_X: VALUE im Datentyp VALUE, zur Belegung der Doppelwortgröße erweitert.
0	Parameter VALUE = +/- INF oder +/- NaN	VALUE wird geschrieben in OUT

Beispiel (KOP): Normieren und Skalieren eines Analogeingangswerts

Ein Analogeingang eines analogen Signalmoduls oder Signalboards mit Stromeingang liegt im Bereich von 0 bis 27648 der gültigen Werte. Angenommen, ein Analogeingang stellt eine Temperatur dar, wobei der Wert 0 des Analogeingangs -30,0 Grad C und der Wert 27648 die Temperatur 70,0 Grad C darstellt.

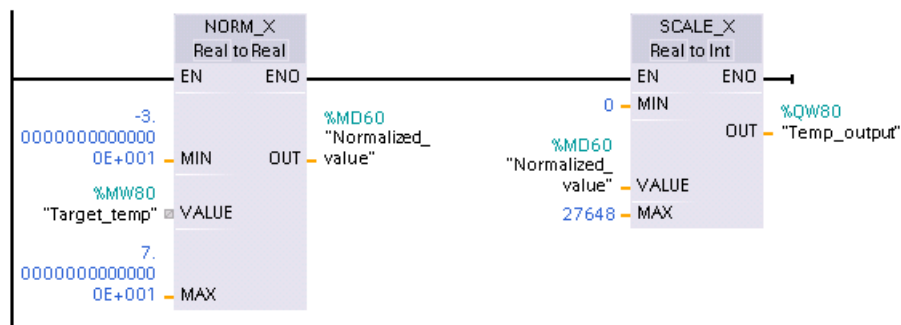
Um den Analogwert in die entsprechenden physikalischen Einheiten umzuwandeln, normieren Sie den Eingang in einen Wert zwischen 0,0 und 1,0 und skalieren ihn dann zwischen -30,0 und 70,0. Der resultierende Wert ist die vom Analogeingang dargestellte Temperatur in Grad C:



Beachten Sie, dass, wenn der Analogeingang von einem analogen Signalmodul oder Signalboard mit Spannungseingang kommen würde, der MIN-Wert für die Anweisung NORM_X -27648 und nicht 0 wäre.

Beispiel (KOP): Normieren und Skalieren eines Analogausgangswerts

Ein in einem analogen Signalmodul oder Signalboard mit Stromausgang zu setzender Analogausgang muss im Bereich von 0 bis 27648 der gültigen Werte liegen. Angenommen, ein Analogausgang stellt eine Temperatureinstellung dar, wobei der Wert 0 des Analogeingangs -30,0 Grad C und der Wert 27648 die Temperatur 70,0 Grad C darstellt. Um im Speicher einen Temperaturwert, der zwischen -30,0 und 70,0 liegt, in einen Wert für den Analogausgang im Bereich zwischen 0 und 27648 umzuwandeln, müssen Sie den Wert in physikalischen Einheiten in einen Wert zwischen 0,0 und 1,0 normieren und ihn dann in den Bereich des Analogausgangs zwischen 0 und 27648 skalieren:



Beachten Sie, dass, wenn der Analogausgang für ein analoges Signalmodul oder Signalboard mit Spannungseingang vorgesehen wäre, der MIN-Wert für die Anweisung SCALE_X -27648 und nicht 0 wäre.

Weitere Informationen zur Darstellung von Analogeingängen (Seite 1358) und zur Darstellung von Analogausgängen (Seite 1359) für Spannung und Strom finden Sie in den technischen Daten.

8.7.6 Variant-Umwandlungsanweisungen

8.7.6.1 VARIANT_TO_DB_ANY (VARIANT in DB_ANY konvertieren)

Mit der Anweisung "VARIANT to DB_ANY" wird der Operand von Parameter IN gelesen und in den Datentyp DB_ANY umgewandelt. Der Parameter IN hat den Datentyp Variant und stellt entweder einen Instanzdatenbaustein oder einen ARRAY-Datenbaustein dar. Beim Anlegen des Programms muss nicht bekannt sein, welcher Datenbaustein dem IN-Parameter entspricht. Die Anweisung liest die Datenbausteinnummer während der Laufzeit und schreibt sie in den Operanden von Parameter RET_VAL .

Tabelle 8-125 Anweisung VARIANT_TO_DB_ANY

KOP/FUP	SCL	Beschreibung
Nicht verfügbar	<pre>RET_VAL := VARIANT_TO_DB_ANY(in := _variant_in_, err => _int_out_);</pre>	Liest den Operanden von Parameter Variant IN und speichert ihn im Funktionsergebnis des Typs DB_ANY

Tabelle 8-126 Parameter für die VARIANT_TO_DB_ANY-Anweisung

Parameter	Datentyp	Beschreibung
IN	Variant	Variant-Element, das einen Instanz- oder Array-Datenbaustein darstellt.
RET_VAL	DB_ANY	Ausgegebener Datentyp DB_ANY, der die umgewandelte Datenbausteinnummer enthält.
ERR	Int	Fehlerinformationen

Tabelle 8-127 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Die Anweisung wandelt den Variant-Eingang um und speichert ihn im Funktionsausgang DB_ANY.
0	Freigabeeingang EN hat Signalzustand "0" oder IN-Parameter ist ungültig.	Die Anweisung tut nichts.

Tabelle 8-128 Fehlercodes für Anweisung VARIANT_TO_DB_ANY

Err (W#16#...)	Beschreibung
0000	Kein Fehler
252C	Der Datentyp Variant von Parameter IN hat den Wert 0. Die CPU wechselt in Betriebszustand STOP.
8131	Der Datenbaustein existiert nicht oder ist zu kurz (erster Zugriff).

Err (W#16#...)	Beschreibung
8132	Der Datenbaustein ist zu kurz und kein Array-Datenbaustein (zweiter Zugriff).
8134	Der Datenbaustein ist schreibgeschützt
8150	Datentyp Variant von Parameter IN liefert den Wert "0". Für diese Fehlermeldung muss die Bausteineigenschaft "Fehler in Baustein bearbeiten" aktiviert sein. Sonst wechselt die CPU in Betriebszustand STOP und sendet Fehlercode 16#252C
8154	Der Datenbaustein hat den falschen Datentyp.
*Feldercode können im Programmeditor als Ganzzahlen oder Hexadezimalwerte angezeigt werden.	

8.7.6.2 DB_ANY_TO_VARIANT (DB_ANY in VARIANT konvertieren)

Mit der Anweisung "DB_ANY to VARIANT" kann die Nummer des Datenbausteins, der die nachstehenden Voraussetzungen erfüllt, gelesen werden. Der Operand von Parameter IN hat den Datentyp DB_ANY, weshalb beim Anlegen des Programms nicht bekannt sein muss, welcher Datenbaustein gelesen werden soll. Die Anweisung liest die Datenbausteinnummer während der Laufzeit und schreibt sie mit Hilfe eines VARIANT-Pointer in das Funktionsergebnis RET_VAL.

Tabelle 8-129 Anweisung DB_ANY_TO_VARIANT

KOP/FUP	SCL	Beschreibung
Nicht verfügbar	<pre>RET_VAL := DB_ANY_TO_VARIANT (in := _db_any_in_, err => _int_out_);</pre>	Liest die Datenbausteinnummer von Parameter Variant IN und speichert sie im Funktionsergebnis des Typs VARIANT.

Tabelle 8-130 Parameter für die Anweisung DB_ANY_TO_VARIANT

Parameter	Datentyp	Beschreibung
IN	DB_ANY	Variant-Element, das die Datenbausteinnummer enthält
RET_VAL	Variant	Ausgegebener Datentyp DB_ANY, der die umgewandelte Datenbausteinnummer enthält.
ERR	Int	Fehlerinformationen

Tabelle 8-131 ENO-Status

ENO	Bedingung	Ergebnis
1	Kein Fehler	Die Anweisung wandelt die Datenbausteinnummer in das Variant-Format und speichert sie im Funktionsausgang DB_ANY
0	Freigabeeingang EN hat Signalzustand "0" oder IN-Parameter ist ungültig.	Die Anweisung tut nichts.

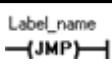

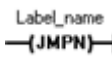



Tabelle 8-132 Fehlercodes für Anweisung DB_ANY_TO_VARIANT

Err (W#16#...)	Beschreibung
0000	Kein Fehler
8130	Nummer des Datenbausteins ist 0.
8131	Der Datenbaustein existiert nicht oder ist zu kurz.
8132	Der Datenbaustein ist zu kurz und kein Array-Datenbaustein.
8134	Der Datenbaustein ist schreibgeschützt.
8154	Der Datenbaustein hat den falschen Datentyp.
8155	Unbekannter Typcode
*Feldercodes können im Programmeditor als Ganzzahlen oder Hexadezimalwerte angezeigt werden.	

8.8 Programmsteuerungsoperationen

8.8.1 Anweisungen JMP (Springen bei VKE = 1), JMPN (Springen bei VKE = 0) und Label (Sprungmarke)

Tabelle 8-133 Anweisungen JMP, JMPN und LABEL

KOP	FUP	SCL	Beschreibung
		Siehe GOTO (Seite 317)-Anweisung.	Springen bei VKE (Verknüpfungsergebnis) = 1: Bei Signalfluss zu einer JMP-Spule (KOP) oder wenn der JMP-Boxeingang wahr ist (FUP), wird die Programmausführung mit der ersten Anweisung nach der angegebenen Sprungmarke fortgesetzt.
			Springen bei VKE = 0: Ohne Signalfluss zu einer JMPN-Spule (KOP) oder wenn der JMPN-Boxeingang falsch ist (FUP), wird die Programmausführung mit der ersten Anweisung nach der angegebenen Sprungmarke fortgesetzt.
			Sprungziel für eine Sprunganweisung JMP oder JMPN

- ¹ Sie erstellen die Namen für die Sprungmarken, indem Sie die Anweisung LABEL direkt eingeben. Über das Parametersymbol können Sie die verfügbaren Sprungmarken für die Anweisung JMP oder JMPN auswählen. Sie können den Namen der Sprungmarke auch direkt in die Anweisung JMP oder JMPN eingeben.

Tabelle 8-134 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Label_name	Kennung der Sprungmarke	Kennzeichnung für Sprunganweisungen und die entsprechende Sprungmarke für das Sprungziel

- Jede Sprungmarke muss innerhalb eines Codebausteins eindeutig sein.
- Sie können innerhalb eines Codebausteins springen, aber Sie können nicht von einem Codebaustein in einen anderen Codebaustein springen.
- Sie können vorwärts oder rückwärts springen.
- Sie können von mehreren Stellen eines Codebausteins zu derselben Sprungmarke springen.

8.8.2 JMP_LIST (Sprungliste definieren)

Tabelle 8-135 Anweisung JMP_LIST

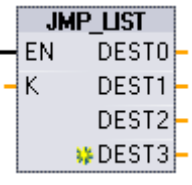
KOP/FUP	SCL	Beschreibung
	<pre> CASE k OF 0: GOTO dest0; 1: GOTO dest1; 2: GOTO dest2; [n: GOTO destn;] END_CASE; </pre>	<p>Die Anweisung JMP_LIST verteilt die Programmsprünge, um die Ausführung von Programmabschnitten zu steuern. Je nach Wert des Eingangs K erfolgt ein Sprung zur entsprechenden Sprungmarke im Programm. Die Programmausführung wird mit den Programmanweisungen fortgesetzt, die auf das Sprungziel folgen. Wenn der Wert des Eingangs K die Anzahl der Sprungmarken - 1 überschreitet, wird kein Sprung durchgeführt und die Verarbeitung wird mit dem nächsten Netzwerk im Programm fortgesetzt.</p>

Tabelle 8-136 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
K	UInt	Steuerwert für die Sprungverteilung
DEST0, DEST1, ..., DESTn.	Sprungmarken im Programm	<p>Sprungmarken der Sprungziele entsprechend spezifischen Werten des Parameters K:</p> <p>Wenn der Wert von K gleich 0 ist, wird ein Sprung zu der Sprungmarke durchgeführt, die dem Ausgang DEST0 zugewiesen ist. Wenn der Wert von K gleich 1 ist, wird ein Sprung zu der Sprungmarke durchgeführt, die dem Ausgang DEST1 zugewiesen ist usw. Wenn der Wert des Eingangs K (die Anzahl der Sprungmarken - 1) überschreitet, wird kein Sprung durchgeführt und die Verarbeitung wird mit dem nächsten Netzwerk im Programm fortgesetzt.</p>

In KOP und FUP: Beim Einfügen der Box JMP_LIST in Ihr Programm sind zwei Ausgänge für Sprungmarken vorhanden. Sie können Sprungziele hinzufügen oder löschen.



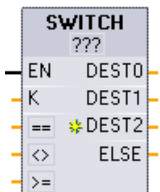
Klicken Sie in der Box auf das Symbol zum Erstellen (links vom letzten Parameter DEST), um neue Ausgänge für Sprungmarken hinzuzufügen.



- Klicken Sie mit der rechten Maustaste auf einen Ausgangsanschluss und wählen Sie "Ausgang einfügen".
- Klicken Sie mit der rechten Maustaste auf einen Ausgangsanschluss und wählen Sie "Löschen".

8.8.3 SWITCH (Sprungverteilung)

Tabelle 8-137 Anweisung SWITCH

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	Die Anweisung SWITCH verteilt die Programmsprünge, um die Ausführung von Programmabschnitten zu steuern. Je nach Ergebnis des Vergleichs zwischen dem Wert des Eingangs K und den Werten der angegebenen Vergleichseingänge wird ein Sprung zu der Sprungmarke durchgeführt, die dem ersten Vergleichstest mit dem Ergebnis WAHR entspricht. Ist keiner der Vergleiche wahr, erfolgt ein Sprung zu der Sprungmarke, die ELSE zugewiesen ist. Die Programmausführung wird mit den Programmanweisungen fortgesetzt, die auf das Sprungziel folgen.

¹ In KOP und FUP: Klicken Sie unterhalb des Box-Namens und wählen Sie in der Klappliste einen Datentyp aus.

² In SCL: Verwenden Sie Vergleiche über IF-THEN.

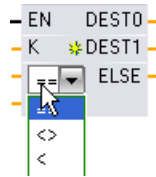
Tabelle 8-138 Datentypen für die Parameter

Parameter	Datentyp ¹	Beschreibung
K	UInt	Eingang für den allgemeinen Vergleichswert
==, <>, <, <=, >, >=	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, TOD, Date	Eingänge für einzelne Vergleichswerte für spezifische Vergleichsarten
DEST0, DEST1, ..., DESTn, ELSE	Sprungmarken im Programm	Sprungmarken der Sprungziele entsprechend spezifischen Vergleichen: Der Vergleichseingang unter und neben dem Eingang K wird zuerst verarbeitet und bewirkt einen Sprung zu der Sprungmarke, die DEST0 zugewiesen ist, wenn der Vergleich zwischen dem Wert K und diesem Eingang wahr ist. Der nächste Vergleichstest verwendet den nächsten Eingang darunter und bewirkt einen Sprung zu der Sprungmarke, die DEST1 zugewiesen ist, wenn der Vergleich wahr ist. Die übrigen Vergleiche werden auf ähnliche Weise verarbeitet und wenn keiner der Vergleiche wahr ist, erfolgt ein Sprung zu der Sprungmarke, die dem Ausgang ELSE zugewiesen ist.

¹ Der Eingang K und die Vergleichseingänge (==, <>, <, <=, >, >=) müssen denselben Datentyp haben.

Eingänge hinzufügen, Eingänge löschen und Vergleichsarten angeben

Beim Einfügen der Box SWITCH in KOP oder FUP in Ihrem Programm sind zwei Vergleichseingänge vorhanden. Sie können Vergleichsarten zuweisen und Eingänge bzw. Sprungziele hinzufügen oder löschen (siehe unten).



Klicken Sie in der Box auf einen Vergleichsoperator und wählen Sie in der Klappliste einen neuen Operator aus.



Klicken Sie in der Box auf das Symbol zum Erstellen (links vom letzten Parameter DEST), um neue Parameter für Vergleichsziele hinzuzufügen.



- Klicken Sie mit der rechten Maustaste auf einen Eingangsanschluss und wählen Sie "Eingang einfügen".
- Klicken Sie mit der rechten Maustaste auf einen Eingangsanschluss und wählen Sie "Löschen".

Tabelle 8-139 Auswahl des Datentyps für die Box SWITCH und zulässige Vergleichsanweisungen

Datentyp	Vergleich	Syntax des Operators
Byte, Word, DWord	Gleich	==
	Ungleich	<>
SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal, Time, TOD, Date	Gleich	==
	Ungleich	<>
	Größer oder gleich	>=
	Kleiner oder gleich	<=
	Größer als	>
	Kleiner als	<

Regeln für die Platzierung der Box SWITCH

- Vor dem Vergleichseingang darf keine KOP/FUP-Anweisung angeschlossen werden.
- Es gibt keinen ENO-Ausgang, deshalb ist nur eine SWITCH-Anweisung in einem Netzwerk zulässig, und die SWITCH-Anweisung muss die letzte Anweisung im Netzwerk sein.

8.8.4 RET (Rückgabewert)

Mit der optionalen Anweisung RET wird die Ausführung des aktuellen Bausteins beendet. Nur bei Signalfluss zur Spule RET (KOP) oder wenn die Box RET wahr ist (FUP), wird die Programmausführung des aktuellen Bausteins an diesem Punkt beendet; nach der Anweisung RET werden keine weiteren Anweisungen ausgeführt. Wenn es sich bei dem aktuellen Baustein um einen OB handelt, wird der Parameter "Return_Value" ignoriert. Wenn es sich bei dem aktuellen Baustein um eine FC oder einen FB handelt, wird der Wert des Parameters "Return_Value" als ENO-Wert der aufgerufenen Box wieder an die aufrufende Routine übergeben.

Sie müssen die Anweisung RET nicht als letzte Anweisung in einen Baustein einfügen, dies geschieht automatisch. Sie können mehrere Anweisungen RET in einen einzigen Baustein einfügen.

Für SCL siehe RETURN (Seite 318)-Anweisung.

Tabelle 8-140 Programmsteuerungsanweisung Rückgabewert (RET)

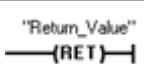

KOP	FUP	SCL	Beschreibung
		RETURN ;	Beendet die Ausführung des aktuellen Bausteins.

Tabelle 8-141 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
Return_Value	Bool	Der Parameter "Return_value" der Anweisung RET ist dem Ausgang ENO der Baustein-aufruf-Box im aufrufenden Baustein zugewiesen.

Beispiel für die Verwendung einer Anweisung RET in einem Codebaustein FC:

1. Erstellen Sie ein neues Projekt und fügen Sie eine FC ein:
2. Bearbeiten Sie die FC:
 - Fügen Sie Anweisungen aus dem Anweisungsverzeichnis ein.
 - Fügen Sie eine Anweisung RET einschließlich einem der folgenden Elemente für den Parameter "Return_Value" ein:
WAHR, FALSCH oder eine Adresse, die den erforderlichen Rückgabewert angibt.
 - Fügen Sie weitere Anweisungen ein.
3. Rufen Sie die FC aus dem Hauptprogramm MAIN [OB1] auf.

Der Eingang EN der FC-Box im Codebaustein MAIN muss wahr sein, damit die FC ausgeführt wird.

Der von der Anweisung RET in der FC angegebene Wert liegt nach der Ausführung der FC, an deren Anweisung RET Signalfluss anliegt, am Ausgang ENO der FC-Box im Codebaustein MAIN an.

8.8.5 ENDIS_PW (CPU-Passwort aktivieren/deaktivieren)

Tabelle 8-142 Anweisung ENDIS_PW

KOP/FUP	SCL	Beschreibung
	<pre> 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_); </pre>	<p>Mit der Anweisung ENDIS_PW kann die Verbindung eines Clients mit einer S7-1200 CPU freigegeben oder gesperrt werden, auch wenn der Client das richtige Passwort bereitstellt.</p> <p>Diese Anweisung sperrt keine Webserver-Passwörter.</p>

Tabelle 8-143 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Funktion ausführen, wenn REQ=1
F_PWD	IN	Bool	Passwort für fehlersicheren Betrieb: Freigeben (=1) oder sperren (=0)
FULL_PWD	IN	Bool	Passwort für Vollzugriff: Passwort für Vollzugriff freigeben (=1) oder sperren (=0)
R_PWD	IN	Bool	Passwort für Lesezugriff: Freigeben (=1) oder sperren (=0)
HMI_PWD	IN	Bool	HMI-Passwort: Freigeben (=1) oder sperren (=0)
F_PWD_ON	OUT	Bool	Status des Passworts für fehlersicheren Betrieb: Freigegeben (=1) oder gesperrt (=0)
FULL_PWD_ON	OUT	Bool	Status des Passworts für Vollzugriff: Freigegeben (=1) oder gesperrt (=0)
R_PWD_ON	OUT	Bool	Status des Passworts für Lesezugriff: Freigegeben (=1) oder gesperrt (=0)
HMI_PWD_ON	OUT	Bool	Status des HMI-Passworts: Freigegeben (=1) oder gesperrt (=0)
Ret_Val	OUT	Word	Funktionsergebnis

Wird ENDIS_PW mit REQ=1 aufgerufen, so werden die Passworttypen gesperrt, für die der entsprechende Passwort-Eingangsparameter FALSCH ist. Jeder Passworttyp kann einzeln freigegeben oder gesperrt werden. Wird zum Beispiel nur das Passwort für den fehlersicheren Betrieb freigegeben und werden alle anderen Passwörter gesperrt, kann damit der Zugriff auf die CPU auf eine kleine Benutzergruppe beschränkt werden.

ENDIS_PW wird synchron in einem Programmzyklus ausgeführt und die Passwort-Ausgangsparameter zeigen immer den aktuellen Stand der Passwortfreigabe, unabhängig vom Eingangsparameter REQ. Passwörter, die Sie freigeben, müssen grundsätzlich auch gesperrt werden können. Andernfalls wird ein Fehler ausgegeben und es werden alle Passwörter freigegeben, die vor der Ausführung von ENDIS_PW freigegeben waren. Dies bedeutet, dass bei einer Standard-CPU (bei der das Passwort für fehlersicheren Betrieb nicht konfiguriert ist) F_PWD

immer auf 1 gesetzt sein muss, um einen Rückgabewert von 0 zu erreichen. In diesem Fall ist F_PWD_ON immer 1.

Hinweis

- Die Ausführung von ENDIS_PW kann den Zugriff auf HMI-Geräte blockieren, wenn das HMI-Passwort gesperrt ist.
 - Client-Sitzungen, die vor der Ausführung von ENDIS_PW freigegeben waren, können durch die Ausführung von ENDIS_PW abhängig von der vorhandenen Legitimierungsstufe beendet werden. So wird beispielsweise eine mit Passwortschutz LESEN legitimierte Verbindung von ENDIS_PW (REQ=1, R_PWD=0) abgebrochen. Andere Verbindungen mit niedrigerer Schutzstufe werden in diesem Szenario ebenfalls beendet. Verbindungen mit VOLLZUGRIFF bleiben hingegen erhalten.
-

Nach dem Anlauf ist der CPU-Zugriff durch die Passwörter eingeschränkt, die zuvor in der regulären Konfiguration des CPU-Schutzes definiert wurden. Die Fähigkeit, ein gültiges Passwort zu sperren, muss mit der Ausführung von ENDIS_PW erneut eingerichtet werden. Wird ENDIS_PW jedoch sofort ausgeführt und werden dadurch erforderliche Passwörter gesperrt, kann der Zugriff auf das TIA Portal gesperrt werden. Mit Hilfe einer Zeitoperation können Sie die Ausführung von ENDIS_PW verzögern und einen Zeitraum zur Eingabe von Passwörtern gestatten, bevor die Passwörter gesperrt werden.

Hinweis

Wiederherstellen einer CPU, deren Kommunikation mit dem TIA Portal gesperrt wurde

Weitere Informationen zum Löschen des internen Ladespeichers eines PLC mit Hilfe einer Memory Card finden Sie unter "Wiederherstellung eines verlorenen Passworts (Seite 134)".

Ein Wechsel des Betriebszustands nach STOP auf Grund von Fehlern, der STP-Ausführung oder von STEP 7 beeinflusst den Schutz nicht. Der Schutz gilt, bis die CPU aus- und wieder eingeschaltet wird. In der folgenden Tabelle finden Sie weitere Details.

Aktion	Betriebszustand	ENDIS_PW-Passwortschutz
Nach dem Urlöschen durch STEP 7	STOP	Aktiv: Gesperrte Passwörter bleiben gesperrt.
Nach dem Einschalten oder Wechseln einer Memory Card	STOP	Aus: Keine Passwörter sind gesperrt.
Nach Ausführung von ENDIS_PW in einem Zyklus- oder Anlauf-OB	STARTUP, RUN	Aktiv: Passwörter sind entsprechend den ENDIS_PW-Parametern gesperrt
Nach einem Wechsel des Betriebszustands von RUN oder STARTUP nach STOP durch STP-Anweisung, Fehler oder STEP 7	STOP	Aktiv: Gesperrte Passwörter bleiben gesperrt


	WARNUNG
Nicht berechtigter Zugriff auf eine geschützte CPU	
<p>Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.</p> <p>Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:</p> <ul style="list-style-type: none"> • Schützen Sie die CPU-Zugriffsstufen (Seite 163) und Webserver-Benutzer-IDs (Seite 852) durch starke Passwörter. • Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden: <ul style="list-style-type: none"> – Großbuchstaben – Kleinbuchstaben – Ziffern – Sonderzeichen • Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234. • Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d • Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll. • Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht. • Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil die Benutzer von Webseiten für PLC-Variablen ungültige Werte festlegen können. • Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN). 	


Tabelle 8-144 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Die Anweisung wird nicht unterstützt.
80D0	Das Passwort für fehlersicheren Betrieb ist nicht konfiguriert.
80D1	Das Passwort für Lese-/Schreibzugriff ist nicht konfiguriert.

RET_VAL (W#16#...)	Beschreibung
80D2	Das Passwort für Lesezugriff ist nicht konfiguriert.
80D3	Das Passwort für HMI-Zugriff ist nicht konfiguriert.

8.8.6 RE_TRIGR (Zyklusüberwachungszeit neu starten)

Tabelle 8-145 Anweisung RE_TRIGR

KOP/FUP	SCL	Beschreibung
	<code>RE_TRIGR () ;</code>	Mit RE_TRIGR (Zykluszeitüberwachung neu starten) kann die maximal zulässige Zeit bis zur Erzeugung einer Fehlermeldung durch die Zykluszeitüberwachung verlängert werden.

Mit der Anweisung RE_TRIGR starten Sie die Zykluszeitüberwachung während eines Zyklusablaufs neu. Dadurch wird die maximal zulässige Zykluszeit mit der letzten Ausführung der Funktion RE_TRIGR um eine maximale Zykluszeit verlängert.

Hinweis

Vor der Firmware-Version 2.2 der S7-1200 CPU musste RE_TRIGR von einem Programmzyklus-OB ausgeführt werden und konnte dazu verwendet werden, die Zykluszeit des Zielsystems unbegrenzt zu verlängern. ENO = FALSCH und die Zykluszeitüberwachung wird nicht zurückgesetzt, wenn RE_TRIGR aus einem Anlauf-OB, einem Alarm-OB oder einem Fehler-OB aufgerufen wird.

Ab Firmware-Version 2.2 kann RE_TRIGR von jedem OB ausgeführt werden (auch Anlauf-, Alarm- und Fehler-OBs). Die Zykluszeit des Zielsystems kann jedoch nur um das 10-Fache der konfigurierten maximalen Zykluszeit verlängert werden.

Maximale Zykluszeit des PLC-Geräts einstellen

Konfigurieren Sie den Wert für die maximale Zykluszeit in der Gerätekonfiguration unter "Zykluszeit".

Tabelle 8-146 Zykluszeitwerte

Zykluszeitüberwachung	Mindestwert	Höchstwert	Voreinstellung
Maximale Zykluszeit	1 ms	6000 ms	150 ms

Laufzeitüberwachung

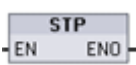
Endet die maximale Zykluszeit, bevor der Zyklus vollständig ist, wird ein Fehler erzeugt. Wenn das Anwenderprogramm einen Zeitfehleralarm-OB (OB 80) enthält, führt die CPU den Zeitfehleralarm-OB aus. Dieser kann Logik enthalten, um eine besondere Reaktion zu bewirken.

Wenn das Anwenderprogramm keinen Zeitfehleralarm-OB enthält, wird die erste Timeout-Bedingung ignoriert und die CPU bleibt im Betriebszustand RUN. Wenn in demselben Programmzyklus eine zweite Überschreitung der maximalen Zykluszeit auftritt (2 x maximaler Zykluszeitwert), wird ein Fehler ausgelöst, der bewirkt, dass das Zielsystem in den Betriebszustand STOP geht.

Im Betriebszustand STOP wird die Ausführung des Programms angehalten, die CPU-Systemkommunikation und die Systemdiagnose laufen jedoch weiter.

8.8.7 STP (Programm beenden)

Tabelle 8-147 Anweisung STP

KOP/FUP	SCL	Beschreibung
	<code>STP () ;</code>	STP versetzt die CPU in den Betriebszustand STOP. Ist die CPU im Betriebszustand STOP, so werden die Ausführung des Programms und die Aktualisierung der Ein- und Ausgänge im Prozessbild angehalten.

Weitere Informationen finden Sie unter: Konfigurieren der Ausgänge für den Wechsel von RUN in STOP (Seite 97).

Ist EN = WAHR, so geht die CPU in den Betriebszustand STOP, die Programmausführung endet und der Zustand von ENO ist bedeutungslos. Sonst ist EN = ENO = 0.

8.8.8 Anweisungen GET_ERROR und GET_ERROR_ID (Fehler lokal abrufen und Fehler-ID lokal abrufen)

Die Anweisungen GET_ERROR liefern Informationen über Fehler in der Ausführung von Programmbausteinen. Wenn Sie eine Anweisung GET_ERROR oder GET_ERROR_ID in Ihren Codebaustein aufnehmen, können Sie Programmfehler in Ihrem Programmbaustein behandeln.

GET_ERROR

Tabelle 8-148 Anweisung GET_ERROR

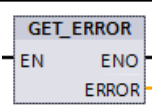
KOP/FUP	SCL	Beschreibung
	<code>GET_ERROR (_out_);</code>	Zeigt an, wenn ein Fehler in der Ausführung eines lokalen Programmbausteins aufgetreten ist und belegt eine vordefinierte Fehlerdatenstruktur mit ausführlichen Fehlerinformationen.

Tabelle 8-149 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
ERROR	ErrorStruct	Fehlerdatenstruktur: Sie können die Struktur, nicht jedoch die Elemente innerhalb der Struktur umbenennen.

Tabelle 8-150 Elemente der ErrorStruct-Datenstruktur

Strukturkomponenten		Datentyp	Beschreibung					
ERROR_ID		Word	Fehler-ID					
FLAGS		Byte	Zeigt an, ob während eines Bausteinaufrufs ein Fehler aufgetreten ist. <ul style="list-style-type: none"> 16#01: Fehler während eines Bausteinaufrufs. 16#00: Kein Fehler während eines Bausteinaufrufs. 					
REACTION		Byte	Standardreaktion: <ul style="list-style-type: none"> 0: Ignorieren (Schreibfehler), 1: Mit Ersatzwert "0" fortfahren (Lesefehler), 2: Anweisung überspringen (Systemfehler) 					
CODE_ADDRESS		CREF	Informationen zur Adresse und zum Bausteintyp					
	BLOCK_TYPE	Byte	Typ des Bausteins, in dem der Fehler aufgetreten ist: <ul style="list-style-type: none"> 1: OB 2: FC 3: FB 					
	CB_NUMBER	UInt	Nummer des Codebausteins					
	OFFSET	UDInt	Verweis auf den internen Speicher					
MODE		Byte	Zugriffsart: Je nach Art des Zugriffs können die folgenden Informationen ausgegeben werden:					
			Betriebsart	(A)	(B)	(C)	(D)	(E)
			0					
			1					Offset
			2			Area		
			3	Adresse	Umfang		Number (Nummer)	
			4			Area		Offset
			5			Area	DB-Nr.	Offset
			6	PtrNr. / Zugr.		Area	DB-Nr.	Offset
7	PtrNr. / Zugr.	Stpl.-Nr./ Umfang	Area	DB-Nr.	Offset			
OPERAND_NUMBER		UInt	Operandennummer des Maschinenbefehls					
POINTER_NUMBER_LOCATION		UInt	(A) Interner Pointer					
SLOT_NUMBER_SCOPE		UInt	(B) Speicherbereich im internen Speicher					
DATA_ADDRESS		NREF	Informationen zur Adresse eines Operanden					
	AREA	Byte	(C) Speicherbereich: <ul style="list-style-type: none"> L: 16#40 – 4E, 86, 87, 8E, 8F, C0 – CE I: 16#81 A: 16#82 M: 16#83 DB: 16#84, 85, 8A, 8B 					

Strukturkomponenten	Datentyp	Beschreibung
DB_NUMBER	UInt	(D) Nummer des Datenbausteins
OFFSET	UDInt	(E) Relative Adresse des Operanden

GET_ERROR_ID

Tabelle 8-151 Anweisung GetErrorID

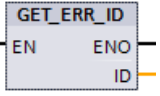
KOP/FUP	SCL	Beschreibung
	<code>GET_ERR_ID () ;</code>	Zeigt an, wenn ein Fehler in der Ausführung eines Programmbausteins aufgetreten ist und meldet die ID (Kennung) des Fehlers.

Tabelle 8-152 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
ID	Word	Fehlerkennwerte für das ErrorStruct-Element ERROR_ID

Tabelle 8-153 Error_ID-Werte

ERROR_ID hexadezimal	ERROR_ID dezimal	Ausführungsfehler Programmbaustein
0	0	Kein Fehler
2520	9504	Zeichenkette beschädigt
2522	9506	Lesefehler Operand außerhalb des Bereichs
2523	9507	Schreibfehler Operand außerhalb des Bereichs
2524	9508	Lesefehler ungültiger Bereich
2525	9509	Schreibfehler ungültiger Bereich
2528	9512	Lesefehler Datenausrichtung (fehlerhafte Bitausrichtung)
2529	9513	Schreibfehler Datenausrichtung (fehlerhafte Bitausrichtung)
252C	9516	Fehler nicht initialisierter Pointer
2530	9520	DB schreibgeschützt
2533	9523	Ungültiger Pointer
2538	9528	Zugriffsfehler: DB ist nicht vorhanden
2539	9529	Zugriffsfehler: Falscher DB
253A	9530	Globaler DB ist nicht vorhanden
253C	9532	Falsche Version oder FC ist nicht vorhanden
253D	9533	Anweisung ist nicht vorhanden
253E	9534	Falsche Version oder FB ist nicht vorhanden
253F	9535	Anweisung ist nicht vorhanden
2550	9552	Zugriffsfehler: DB ist nicht vorhanden
2575	9589	Fehler in der Schachtelungstiefe des Programms
2576	9590	Zuordnungsfehler Lokaldaten

ERROR_ID hexadezimal	ERROR_ID dezimal	Ausführungsfehler Programmbaustein
2942	10562	Physischer Eingang ist nicht vorhanden
2943	10563	Physischer Ausgang ist nicht vorhanden

Funktionsweise

Standardmäßig reagiert die CPU auf einen Bausteinausführungsfehler durch Eintragen des Fehlers in den Diagnosepuffer. Wenn Sie jedoch eine oder mehrere Anweisungen GET_ERROR oder GET_ERROR_ID in einem Codebaustein anordnen, kann dieser Baustein die Fehler im Baustein behandeln. Dann meldet die CPU keinen Fehler im Diagnosepuffer. Stattdessen werden die Fehlerinformationen im Ausgang der Anweisung GET_ERROR oder GET_ERROR_ID gemeldet. Ausführliche Fehlerinformationen können Sie mit der Anweisung GET_ERROR auslesen, mit der Anweisung GET_ERROR_ID wird nur die Fehler-ID gelesen. Normalerweise ist der erste Fehler der wichtigste und die weiteren Fehler lediglich Folgen des ersten Fehlers.

Die erste Ausführung einer Anweisung GET_ERROR oder GET_ERROR_ID innerhalb eines Bausteins gibt den ersten während der Bausteinausführung erkannten Fehler aus. Dieser Fehler kann an einer beliebigen Stelle zwischen dem Start des Bausteins und der Ausführung von GET_ERROR oder GET_ERROR_ID aufgetreten sein. Nachfolgende Ausführungen von GET_ERROR oder GET_ERROR_ID geben den ersten Fehler nach der vorherigen Ausführung von GET_ERROR oder GET_ERROR_ID aus. Der Verlauf der Fehler wird nicht gespeichert, und die Ausführung der Anweisung initiiert das Zielsystem neu, um den nächsten Fehler zu erfassen.

Der Datentyp ErrorStruct der Anweisung GET_ERROR kann im Datenbausteineditor und in den Bausteinschnittstellen eingefügt werden, damit Ihre Programmlogik auf diese Werte zugreifen kann. Wählen Sie den Datentyp ErrorStruct in der Klappliste aus, um diese Struktur zu ergänzen. Mithilfe von eindeutigen Namen können Sie mehrere ErrorStruct-Elemente erstellen. Die Elemente eines ErrorStruct können nicht umbenannt werden.

Von ENO angezeigte Fehlerbedingung

Wenn gilt: EN = WAHR und GET_ERROR oder GET_ERROR_ID wird ausgeführt, dann:

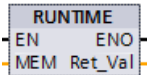
- ENO = WAHR gibt an, dass während der Ausführung eines Codebausteins ein Fehler aufgetreten ist und Fehlerdaten vorliegen
- ENO = FALSCH gibt an, dass während der Ausführung eines Codebausteins kein Fehler aufgetreten ist

Sie können die Logik des Fehlerverarbeitungsprogramms mit dem Eingang ENO verknüpfen, der nach dem Auftreten eines Fehlers gesetzt wird. Bei einem Fehler werden die Fehlerdaten im Ausgangsparameter gespeichert, wo Ihr Programm darauf zugreifen kann.

Mit GET_ERROR und GET_ERROR_ID können Fehlerinformationen des aktuellen Bausteins (des aufgerufenen Bausteins) an einen aufrufenden Baustein gesendet werden. Platzieren Sie die Anweisung im letzten Netzwerk des aufgerufenen Bausteinprogramms, um den endgültigen Ausführungsstatus des aufgerufenen Bausteins zu melden.

8.8.9 RUNTIME (Programmlaufzeit messen)

Tabelle 8-154 Anweisung RUNTIME

KOP/FUP	SCL	Beschreibung
	<pre>Ret_Val := RUNTIME(_lread_inout_);</pre>	Misst die Laufzeit des gesamten Programms, einzelner Bausteine oder von Befehlssequenzen.

Soll die Laufzeit des gesamten Programms gemessen werden, ist die Anweisung "Programmlaufzeit messen" in OB 1 aufzurufen. Die Laufzeitmessung beginnt mit dem ersten Aufruf und Ausgang RET_VAL gibt die gemessene Programmlaufzeit nach dem zweiten Aufruf zurück. Die gemessene Laufzeit umfasst alle CPU-Prozesse, die während der Programmausführung auftreten können, zum Beispiel Unterbrechungen durch übergeordnete Ereignisse oder Kommunikation. Anweisung "Programmlaufzeit messen" liest einen internen Zähler der CPU und schreibt den Wert in den IN-OUT-Parameter MEM. Die Anweisung berechnet die aktuelle Programmlaufzeit anhand der internen Zählerfrequenz und schreibt sie in Ausgang RET_VAL.

Soll die Laufzeit einzelner Bausteine oder Befehlssequenzen gemessen werden, werden drei separate Netzwerke benötigt. Rufen Sie Anweisung "Programmlaufzeit messen" in einem einzelnen Netzwerk Ihres Programms auf. Sie setzen den Startpunkt der Laufzeitmessung mit dem ersten Aufruf der Anweisung. Rufen Sie dann den benötigten Programmbaustein oder die Befehlssequenz im nächsten Netzwerk auf. Rufen Sie Anweisung "Programmlaufzeit messen" in einem anderen Netzwerk ein zweites Mal auf und weisen Sie dem IN-OUT-Parameter MEM den gleichen Speicher wie beim ersten Aufruf der Anweisung zu. Die Anweisung "Programmlaufzeit messen" im dritten Netzwerk liest einen CPU-internen Zähler, berechnet die aktuelle Laufzeit der Programmbausteine oder der Befehlssequenz anhand der internen Zählerfrequenz und schreibt diese in Ausgang RET_VAL.

Anweisung "Programmlaufzeit messen" nutzt einen internen Hochfrequenzzähler für die Berechnung der Zeit. Bei Zählerüberlauf gibt die Anweisung Werte ≤ 0.0 zurück. Diese Laufzeitwerte können ignoriert werden.

Hinweis

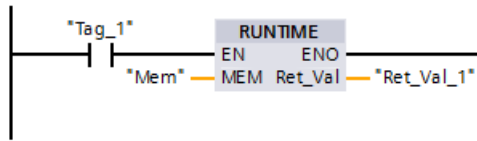
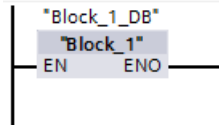
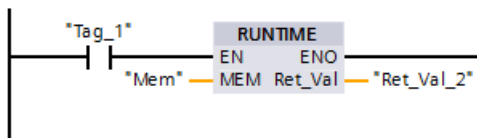
Die CPU kann die Laufzeit einer Befehlssequenz nicht genau bestimmen, weil sich die Anweisungsfolge innerhalb einer Befehlssequenz während der optimierten Programmkompilierung ändert.

Tabelle 8-155 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
MEM	LReal	Startpunkt für die Laufzeitmessung
RET_VAL	LReal	Gemessene Laufzeit in Sekunden

Beispiel: Anweisung RUNTIME

Das folgende Beispiel zeigt, wie Anweisung RUNTIME für die Messung der Ausführungszeit eines Funktionsbausteins verwendet wird:

Netzwerk 1:**Netzwerk 2:****Netzwerk 3:**

Wenn der Operand "Tag_1" in Netzwerk 1 Signalzustand "1" hat, wird Anweisung RUNTIME ausgeführt. Der Startpunkt für die Laufzeitmessung wird mit dem ersten Aufruf der Anweisung gesetzt und als Referenz für den zweiten Aufruf der Anweisung in Operand "Mem" gepuffert.

Funktionsbaustein FB1 wird in Netzwerk 2 ausgeführt.

Wenn Programmbaustein FN1 ausgeführt ist und Operand "Tag_1" Signalzustand "1" hat, wird Anweisung RUNTIME in Netzwerk 3 ausgeführt. Der zweite Aufruf der Anweisung berechnet die Laufzeit des Programmbausteins und schreibt das Ergebnis in Ausgang RET_VAL_2.

8.8.10 Programmsteuerungsanweisungen in SCL

8.8.10.1 Übersicht über die Programmsteuerungsanweisungen in SCL

Structured Control Language (SCL) bietet drei Arten von Programmsteuerungsanweisungen für die Strukturierung Ihres Anwenderprogramms:

- **Auswahanweisungen:** Mit Hilfe einer Auswahanweisung können Sie den Programmfluss in alternative Anweisungsfolgen steuern.
- **Schleifen:** Sie können die Schleifenausführung über Wiederholungsanweisungen steuern. Eine Wiederholungsanweisung gibt an, welche Teile eines Programm abhängig von bestimmten Bedingungen wiederholt werden sollen.
- **Programmsprünge:** Ein Programmsprung bewirkt einen direkten Sprung zu einem angegebenen Sprungziel und damit zu einer anderen Anweisung in demselben Baustein.

Die Programmsteuerungsanweisungen verwenden die Syntax der Programmiersprache PASCAL.

Tabelle 8-156 Arten von Programmsteuerungsanweisungen in SCL

Programmsteuerungsanweisung	Beschreibung	
Auswahl	IF-THEN-Anweisung (Seite 311)	Ermöglicht Ihnen, den Programmfluss in Abhängigkeit von einer Bedingung, die entweder WAHR oder FALSCH ist, in eine von zwei Alternativen zu verzweigen.
	CASE-Anweisung (Seite 312)	Ermöglicht die selektive Ausführung einer von n alternativen Verzweigungen, basierend auf dem Wert einer Variablen.
Schleife	FOR-Anweisung (Seite 313)	Dient zur Wiederholung einer Folge von Anweisungen, solange die Steuervariable innerhalb des angegebenen Wertebereichs liegt.
	WHILE-DO-Anweisung (Seite 314)	Dient zur Wiederholung einer Folge von Anweisungen, solange eine Durchführungsbedingung erfüllt ist.
	REPEAT-UNTIL-Anweisung (Seite 315)	Dient zur Wiederholung einer Folge von Anweisungen, bis eine Abbruchbedingung erfüllt ist.
Programmsprung	CONTINUE-Anweisung (Seite 316)	Dient zum Abbruch der Ausführung des momentanen Schleifendurchlaufs.
	EXIT-Anweisung (Seite 317)	Dient zum Verlassen einer Schleife an beliebiger Stelle und unabhängig vom Erfülltsein der Abbruchbedingung.
	GOTO-Anweisung (Seite 317)	Bewirkt den sofortigen Sprung zu einer angegebenen Sprungmarke.
	RETURN-Anweisung (Seite 318)	Bewirkt das Verlassen des gerade ausgeführten Codebausteins und die Rückkehr zum aufrufenden Codebaustein.

8.8.10.2 IF-THEN-Anweisung

Die IF-THEN-Anweisung ist eine bedingte Anweisung, die den Programmfluss steuert, indem eine Gruppe von Anweisungen basierend auf der Auswertung eines Bool-Werts eines logischen Ausdrucks ausgeführt wird. Die Ausführung mehrerer IF-THEN-Anweisungen können Sie auch mit Hilfe von Klammern verschachteln bzw. strukturieren.

Tabelle 8-157 Elemente der IF-THEN-Anweisung

SCL	Beschreibung
IF "Bedingung" THEN Anweisung_A; Anweisung_B; Anweisung_C; ;	Falls "Bedingung" WAHR oder 1 ist, die folgenden Anweisungen bis zur Anweisung END_IF ausführen. Falls "Bedingung" FALSCH oder 0 ist, bis zur Anweisung END_IF überspringen (sofern das Programm keine optionalen ELSIF- oder ELSE-Anweisungen enthält).
[ELSIF "Bedingung-n" THEN Anweisung_N; ;]	Die optionale ELSEIF ¹ -Anweisung bietet weitere auszuwertende Bedingungen. Beispiel: Falls "Bedingung" in der IF-THEN-Anweisung FALSCH ist, wertet das Programm "Bedingung-n" aus. Falls "Bedingung-n" WAHR ist, "Anweisung_N" ausführen.
[ELSE Anweisung_X; ;]	Die optionale ELSE-Anweisung bietet Anweisungen, die auszuführen sind, wenn die "Bedingung" der IF-THEN-Anweisung FALSCH ist.
END_IF;	Die END_IF-Anweisung beendet die IF-THEN-Anweisung.

¹ Sie können in eine IF-THEN-Anweisung mehrere ELSIF-Anweisungen einfügen.

Tabelle 8-158 Variablen der IF-THEN-Anweisung

Variablen	Beschreibung
"Bedingung"	Erforderlich. Der logische Ausdruck ist entweder WAHR (1) oder FALSCH (0).
"Anweisung_A"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung" WAHR ist.
"Bedingung-n"	Optional. Der logische Ausdruck, der von der optionalen ELSIF-Anweisung auszuwerten ist.
"Anweisung_N"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung-n" der ELSIF-Anweisung WAHR ist.
"Anweisung_X"	Optional. Eine oder mehrere Anweisungen, die auszuführen sind, wenn "Bedingung" der IF-THEN-Anweisung FALSCH ist.

Eine IF-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Die erste Anweisungsfolge, deren logischer Ausdruck = WAHR ist, wird ausgeführt. Die übrigen Anweisungsfolgen werden nicht ausgeführt.
- Wenn kein Boolescher Ausdruck = WAHR ist, wird die von ELSE eingeführte Anweisungsfolge ausgeführt (oder keine Anweisungsfolge, falls keine ELSE-Verzweigung vorhanden ist).
- Die Anzahl der ELSIF-Anweisungen ist nicht begrenzt.

Hinweis

Die Verwendung einer oder mehrerer ELSIF-Verzweigungen hat gegenüber einer Folge von IF-Anweisungen den Vorteil, dass die logischen Ausdrücke, die auf einen gültigen Ausdruck folgen, nicht mehr ausgewertet werden. Die Laufzeit eines Programms kann dadurch verringert werden.

8.8.10.3 CASE-Anweisung

Tabelle 8-159 Elemente der CASE-Anweisung

SCL	Beschreibung
<pre> CASE "Testwert" OF "Werteliste": Anweisung[; Anweisung, ...] "Werteliste": Anweisung[; Anweisung, ...] [ELSE Else-Anweisung[; Else-Anweisung, ...]] END_CASE;</pre>	Die Anweisung CASE führt eine von mehreren Anweisungsgruppen abhängig vom Wert eines Ausdrucks aus.

Tabelle 8-160 Parameter

Parameter	Beschreibung
"Testwert"	Erforderlich. Beliebiger numerischer Ausdruck vom Datentyp Int.
"Werteliste"	Erforderlich. Ein einzelner Wert oder eine Liste mit durch Komma getrennten Werten oder Wertebereichen. (Einen Wertebereich geben Sie mittels zwei Punkten an: 2..8). Das folgende Beispiel zeigt die verschiedenen Varianten von Wertelisten: 1: Anweisung_A; 2, 4: Anweisung_B; 3, 5..7,9: Anweisung_C;

Parameter	Beschreibung
Anweisung	Erforderlich. Eine oder mehrere Anweisungen, die ausgeführt werden, wenn "Testwert" einem Wert in der Werteliste entspricht.
Else-Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, wenn kein Wert der "Werteliste" eine Übereinstimmung ergibt.

Eine CASE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Der Testwert-Ausdruck muss einen Wert des Typs Int zurückgeben.
- Bei der Verarbeitung einer Anweisung CASE prüft das Programm, ob der Wert des Testwert-Ausdrucks in einer angegebenen Werteliste enthalten ist. Wird eine Übereinstimmung gefunden, wird die der Liste zugewiesene Anweisungskomponente ausgeführt.
- Wird keine Übereinstimmung gefunden, wird der auf ELSE folgende Programmabschnitt ausgeführt. Falls keine ELSE-Verzweigung vorhanden ist, wird keine Anweisung ausgeführt.

Beispiel: Verschachtelte CASE-Anweisungen

CASE-Anweisungen können verschachtelt werden. Jede verschachtelte CASE-Anweisung benötigt eine zugehörige END_CASE-Anweisung.

```

CASE "var1" OF
    1 : #var2 := 'A';
    2 : #var2 := 'B';
ELSE
    CASE "var3" OF

        65..90: #var2 := ,Großbuchstabe';
        97..122: #var2 := ,Kleinbuchstabe';

    ELSE

        #var2:= 'Sonderzeichen';

    END_CASE;
END_CASE;

```

8.8.10.4 FOR-Anweisung

Tabelle 8-161 Elemente der FOR-Anweisung

SCL	Beschreibung
<pre> FOR "Steuervariable" := "Anfang" TO "Ende" [BY "Inkrement"] DO Anweisung; ; END_FOR; </pre>	<p>Eine FOR-Anweisung dient zur Wiederholung einer Folge von Anweisungen, solange eine Steuervariable innerhalb des angegebenen Wertebereichs liegt. Die Definition einer Schleife mit FOR umfasst die Angabe eines Anfangs- und eines Endwerts. Beide Werte müssen denselben Typ haben wie die Steuervariable.</p> <p>FOR-Schleifen können Sie verschachteln. Die END_FOR-Anweisung bezieht sich auf die zuletzt ausgeführte FOR-Anweisung.</p>

Tabelle 8-162 Parameter

Parameter	Beschreibung
"Steuervariable"	Erforderlich. Eine Ganzzahl (Int oder DInt), die als Schleifenzähler dient.
"Anfang"	Erforderlich. Einfacher Ausdruck, der den Anfangswert der Steuervariablen angibt.
"Ende"	Erforderlich. Einfacher Ausdruck, der den Abschlusswert der Steuervariablen angibt.
"Inkrement"	Optional. Betrag, um den eine "Steuervariable" nach jeder Schleife geändert wird. Das "Inkrement" hat denselben Datentyp wie "Steuervariable". Wenn der Wert von "Inkrement" nicht angegeben ist, wird der Wert der Variablen nach jeder Schleife um 1 erhöht. Sie können "Inkrement" nicht während der Ausführung der FOR-Anweisung ändern.

Die FOR-Anweisung wird wie folgt ausgeführt:

- Beim Start der Schleife wird die Steuervariable auf den Anfangswert gesetzt (Anfangszuweisung) und bei jedem Schleifendurchlauf um das angegebene Inkrement erhöht (positives Inkrement) oder verringert (negatives Inkrement), bis der Endwert erreicht ist.
- Nach jedem Schleifendurchlauf wird geprüft, ob die Bedingung (Endwert erreicht) erfüllt ist oder nicht. Ist die Endbedingung nicht erfüllt, so wird die Anweisungssequenz erneut ausgeführt; andernfalls endet die Schleife, und die Ausführung wird mit der unmittelbar auf die Schleife folgenden Anweisung fortgesetzt.

Regeln für die Formulierung von FOR-Anweisungen:

- Die Steuervariable darf nur vom Datentyp Int oder DInt sein.
- Sie können die Anweisung BY [Inkrement] weglassen. Wenn kein Inkrement angegeben wird, wird automatisch ein Inkrement von +1 angenommen.

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die Anweisung EXIT-Anweisung (Seite 317). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END_FOR-Anweisung folgt.

Mit der Anweisung CONTINUE-Anweisung (Seite 316) überspringen Sie die nachfolgenden Anweisungen einer FOR-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

8.8.10.5 WHILE-DO-Anweisung

Tabelle 8-163 WHILE-Anweisung

SCL	Beschreibung
<pre>WHILE "Bedingung" DO Anweisung; Anweisung; ...; END_WHILE;</pre>	<p>Die WHILE-Anweisung führt eine Anweisungsfolge aus, bis eine vorgegebene Bedingung WAHR ist.</p> <p>WHILE-Schleifen können Sie verschachteln. Die END_WHILE-Anweisung bezieht sich auf die zuletzt ausgeführte WHILE-Anweisung.</p>

Tabelle 8-164 Parameter

Parameter	Beschreibung
"Bedingung"	Erforderlich. Ein logischer Ausdruck, dessen Auswertung WAHR oder FALSCH ergibt. (Die Bedingung "Null" wird als FALSCH ausgewertet.)
Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, bis die Bedingung WAHR ist.

Hinweis

Die WHILE-Anweisung wertet den Zustand der "Bedingung" aus, bevor die Anweisungen ausgeführt werden. Um die Anweisungen unabhängig vom Zustand der "Bedingung" mindestens einmal auszuführen, verwenden Sie die REPEAT-Anweisung (Seite 315).

Die WHILE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Vor jeder Ausführung des Schleifeninhalts, wird die Ausführungsbedingung ausgewertet.
- Der auf DO folgende Schleifeninhalt wird solange wiederholt, bis die Ausführungsbedingung den Wert WAHR annimmt.
- Tritt der Wert FALSCH auf, wird die Schleife übersprungen und die auf die Schleife folgende Anweisung wird ausgeführt.

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die EXIT-Anweisung (Seite 317). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END_WHILE-Anweisung folgt.

Mit der CONTINUE-Anweisung überspringen Sie die nachfolgenden Anweisungen einer WHILE-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

8.8.10.6 REPEAT-UNTIL-Anweisung

Tabelle 8-165 REPEAT-Anweisung

SCL	Beschreibung
<pre> REPEAT Anweisung; ; UNTIL "Bedingung" END_REPEAT;</pre>	<p>Die REPEAT-Anweisung führt eine Gruppe von Anweisungen aus, bis eine vorgegebene Bedingung WAHR ist.</p> <p>REPEAT-Schleifen können Sie verschachteln. Die END_REPEAT-Anweisung bezieht sich immer auf die zuletzt ausgeführte REPEAT-Anweisung.</p>

Tabelle 8-166 Parameter

Parameter	Beschreibung
Anweisung	Optional. Eine oder mehrere Anweisungen, die ausgeführt werden, bis die Bedingung WAHR ist.
"Bedingung"	Erforderlich. Einer oder mehrere Ausdrücke der zwei folgenden Arten: Ein numerischer Ausdruck oder ein Zeichenkettenausdruck, dessen Auswertung WAHR oder FALSCH ergibt. Die Bedingung "Null" wird als FALSCH ausgewertet.

Hinweis

Vor der Auswertung des Zustands der "Bedingung" führt die REPEAT-Anweisung die Anweisungen während der ersten Durchführung der Schleife aus (auch wenn die "Bedingung" FALSCH ist). Um den Zustand der "Bedingung" vor der Ausführung der Anweisungen zu prüfen, verwenden Sie die WHILE-Anweisung (Seite 314).

Um die Schleife unabhängig vom Zustand des Ausdrucks "Bedingung" zu beenden, verwenden Sie die EXIT-Anweisung (Seite 317). Die EXIT-Anweisung führt die Anweisung aus, die unmittelbar auf die END_REPEAT-Anweisung folgt.

Mit der CONTINUE-Anweisung (Seite 316) überspringen Sie die nachfolgenden Anweisungen einer REPEAT-Schleife und setzen die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist.

8.8.10.7 CONTINUE-Anweisung

Tabelle 8-167 CONTINUE-Anweisung

SCL	Beschreibung
CONTINUE Anweisung; ;	Die CONTINUE-Anweisung überspringt die nachfolgenden Anweisungen einer Programmschleife (FOR, WHILE, REPEAT) und setzt die Schleife mit der Überprüfung fort, ob die Bedingung für die Beendigung erfüllt ist. Ist dies nicht der Fall, wird die Schleife fortgesetzt.

Die CONTINUE-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Diese Anweisung beendet die Ausführung des Schleifeninhalts sofort.
- Abhängig davon, ob die Bedingung für die Wiederholung der Schleife erfüllt ist oder nicht, wird der Schleifeninhalt erneut ausgeführt oder die Wiederholungsanweisung beendet und die unmittelbar nachfolgende Anweisung ausgeführt.
- In einer FOR-Anweisung wird die Steuervariable direkt nach einer CONTINUE-Anweisung um das angegebene Inkrement erhöht.

Verwenden Sie die CONTINUE-Anweisung nur innerhalb einer Schleife. In verschachtelten Schleifen bezieht sich die CONTINUE-Anweisung immer auf die Schleife, in der sich die Anweisung befindet. CONTINUE wird üblicherweise in Verbindung mit einer IF-Anweisung verwendet.

Wenn die Schleife unabhängig von der Beendigungsprüfung beendet werden soll, verwenden Sie die EXIT-Anweisung.

Beispiel: CONTINUE-Anweisung

Das folgende Beispiel zeigt die Verwendung der CONTINUE-Anweisung, mit der beim Berechnen eines Prozentwerts der Fehler "Division durch 0" verhindert werden soll:

```
FOR i := 0 TO 10 DO
  IF Wert[i] = 0 THEN CONTINUE; END_IF;
  p := Teil / Wert[i] * 100;
  s := INT_TO_STRING(p);
  Prozent := CONCAT(IN1:=s, IN2:="%");
END_FOR;
```

8.8.10.8 EXIT-Anweisung

Tabelle 8-168 Anweisung EXIT

SCL	Beschreibung
EXIT;	Die EXIT-Anweisung dient zum Beenden einer Schleife (FOR, WHILE oder REPEAT) an beliebiger Stelle und unabhängig davon, ob die Beendigungsbedingung erfüllt ist.

Die EXIT-Anweisung wird entsprechend den folgenden Regeln ausgeführt:

- Diese Anweisung bewirkt, dass die Wiederholungsanweisung, die die EXIT-Anweisung unmittelbar umgibt, sofort beendet wird.
- Die Ausführung des Programm wird nach dem Ende der Schleife fortgesetzt (z. B. nach END_FOR).

Verwenden Sie die EXIT-Anweisung innerhalb einer Schleife. In verschachtelten Schleifen gibt die EXIT-Anweisung die Verarbeitung an die nächsthöhere Schachtelungebene zurück.

Beispiel: EXIT-Anweisung

```
FOR i := 0 TO 10 DO
CASE Wert[i, 0] OF
  1..10: Wert [i, 1]:="A";
  11..40: Wert [i, 1]:="B";
  41..100: Wert [i, 1]:="C";
ELSE
EXIT;
END_CASE;
END_FOR;
```

8.8.10.9 GOTO-Anweisung

Tabelle 8-169 GOTO-Anweisung

SCL	Beschreibung
GOTO JumpLabel; Anweisung; ... ; JumpLabel: Anweisung;	Die GOTO-Anweisung überspringt Anweisungen, indem sie zu einer Sprungmarke in demselben Baustein springt. Die Sprungmarke ("JumpLabel") und die GOTO-Anweisung müssen sich in demselben Baustein befinden. Der Name einer Sprungmarke darf innerhalb eines Bausteins nur einmal vergeben werden. Jede Sprungmarke kann Ziel verschiedener GOTO-Anweisungen sein.

Es ist nicht möglich, in einen Schleifenabschnitt zu springen (FOR, WHILE oder REPEAT). Es ist möglich, aus einer Schleife herauszuspringen.

Beispiel: GOTO-Anweisung

Im folgenden Beispiel gilt: Je nach Wert des Operanden "Variablenwert" wird die Ausführung des Programms an dem Punkt wieder aufgenommen, der von der entsprechenden Sprungmarke vorgegeben wird. Ist "Variablenwert" gleich 2, nimmt das Programm die Ausführung an Sprungmarke "MyLabel2" wieder auf und überspringt "MyLabel1".

```
CASE "Variablenwert" OF
  1 : GOTO MeineBeschriftung1;
  2 : GOTO MeineBeschriftung2;
```

```

ELSE GOTO MeineBeschriftung3;
END_CASE;
MeineBeschriftung1: "Variable_1" := 1;
MeineBeschriftung2: "Variable_2" := 1;
MeineBeschriftung3: "Variable_4" := 1;

```

8.8.10.10 RETURN-Anweisung

Tabelle 8-170 Anweisung RETURN

SCL	Beschreibung
RETURN;	Die RETURN-Anweisung beendet den Codebaustein, der gerade ausgeführt wird, ohne Bedingungen. Die Programmausführung kehrt zum aufrufenden Baustein oder zum Betriebssystem zurück (beim Beenden eines OB).

Beispiel: RETURN-Anweisung:

```

IF "Fehler" <> 0 THEN
RETURN;
END_IF;

```

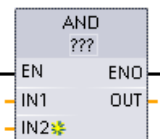
Hinweis

Nach der Ausführung der letzten Anweisung kehrt der Codebaustein automatisch zum aufrufenden Baustein zurück. Fügen Sie eine RETURN-Anweisung nicht am Ende des Codebausteins ein.

8.9 Wortverknüpfung

8.9.1 Verknüpfungsoperationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER)

Tabelle 8-171 Verknüpfungsoperationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER)

KOP/FUP	SCL	Beschreibung
	<code>out := in1 AND in2;</code>	AND: UND-Verknüpfung
	<code>out := in1 OR in2;</code>	OR: ODER-Verknüpfung
	<code>out := in1 XOR in2;</code>	XOR: EXKLUSIV-ODER-Verknüpfung

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 8-172 Datentypen für die Parameter

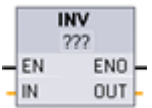
Parameter	Datentyp	Beschreibung
IN1, IN2	Byte, Word, DWord	Verknüpfungseingänge
OUT	Byte, Word, DWord	Verknüpfungsausgang

¹ Durch die Auswahl des Datentyps werden die Parameter IN1, IN2 und OUT auf den gleichen Datentyp gesetzt.

Die entsprechenden Bitwerte von IN1 und IN2 werden zu einem binären Ergebnis in Parameter OUT verknüpft. ENO ist nach der Ausführung dieser Anweisungen immer WAHR.

8.9.2 INV (Einerkomplement erstellen)

Tabelle 8-173 Anweisung INV

KOP/FUP	SCL	Beschreibung
	Nicht verfügbar	Berechnet das binäre Einerkomplement des Parameters IN. Das Einerkomplement wird gebildet durch Invertieren jedes Bitwerts des Parameters IN (Ändern von 0 in 1 und 1 in 0). ENO ist nach der Ausführung dieser Anweisung immer WAHR.

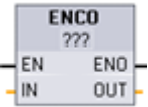
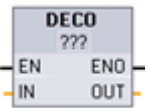
¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-174 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord	Datenelement für die Invertierung
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Byte, Word, DWord	Invertierter Ausgangswert

8.9.3 Anweisungen DECO (Decodieren) and ENCO (Encodieren)

Tabelle 8-175 Anweisungen ENCO und DECO

KOP/FUP	SCL	Beschreibung
	<pre>out := ENCO(_in_);</pre>	<p>Wandelt ein Bitmuster in eine Binärzahl um.</p> <p>Die Anweisung ENCO wandelt den Parameter IN in die Binärzahl um, die der Bitposition des niederwertigsten Bits von Parameter IN entspricht, und gibt das Ergebnis in Parameter OUT aus. Ist Parameter IN entweder 0000 0001 oder 0000 0000, so wird der Wert 0 an Parameter OUT zurückgegeben. Ist der Wert des Parameters IN gleich 0000 0000, wird ENO auf FALSCH gesetzt.</p>
	<pre>out := DECO(_in_);</pre>	<p>Wandelt eine Binärzahl in ein Bitmuster um.</p> <p>Die Anweisung DECO wandelt eine Binärzahl aus Parameter IN um, indem die entsprechende Bitposition in Parameter OUT auf 1 gesetzt wird (alle anderen Bits werden auf 0 gesetzt). ENO ist nach der Ausführung der Anweisung DECO immer WAHR.</p> <p>Hinweis: Der Standarddatentyp für die Anweisung DECO ist DWORD. In SCL ändern Sie den Anweisungsnamen in DECO_BYTE oder DECO_WORD, um einen Byte- oder Wortwert zu decodieren, und Sie weisen eine Byte- oder Wortvariable oder -adresse zu.</p>

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 8-176 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	ENCO: Byte, Word, DWord DECO: UInt	ENCO: Bitmuster für die Umwandlung DECO: Binärwert für die Umwandlung
OUT	ENCO: Int DECO: Byte, Word, DWord	ENCO: Umgewandelter Wert DECO: Umgewandeltes Bitmuster

Tabelle 8-177 ENO-Status

ENO	Bedingung	Ergebnis (OUT)
1	Kein Fehler	Gültige Bitnummer
0	IN ist Null	OUT wird auf Null gesetzt

Die möglichen Datentypen Byte, Word oder DWord für den DECO-Parameter OUT begrenzen den nützlichen Bereich des Parameters IN. Wenn der Wert des Parameters IN den nützlichen Bereich überschreitet, wird eine Modulo-Operation durchgeführt, um die niederwertigsten Bits zu extrahieren (siehe unten).

Bereich für den DECO-Parameter IN:

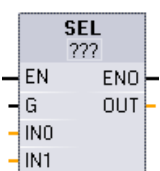
- 3 Bits (Werte 0-7) IN werden verwendet, um 1 Bitposition in einem Byte OUT zu setzen
- 4 Bits (Werte 0-15) IN werden verwendet, um 1 Bitposition in einem Word OUT zu setzen
- 5 Bits (Werte 0-31) IN werden verwendet, um 1 Bitposition in einem DWord OUT zu setzen

Tabelle 8-178 Beispiele

Wert IN für DECO			Wert OUT für DECO (Einzelne Bitposition umwandeln)
Byte OUT 8 Bits	Min. IN	0	00000001
	Max. IN	7	10000000
Word OUT 16 Bits	Min. IN	0	0000000000000001
	Max. IN	15	1000000000000000
DWord OUT 32 Bits	Min. IN	0	00000000000000000000000000000001
	Max. IN	31	10000000000000000000000000000000

8.9.4 Anweisungen SEL (Selektieren), MUX (Multiplexen) und DEMUX (Demultiplexen)

Tabelle 8-179 Anweisung SEL (Selektieren)

KOP/FUP	SCL	Beschreibung
	<pre>out := SEL(g:=_bool_in, in0:=-_variant_in, in1:=-_variant_in);</pre>	Die Anweisung SEL weist einen von zwei Eingangswerten, abhängig vom Wert des Parameters G, dem Parameter OUT zu.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

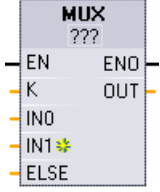
Tabelle 8-180 Datentypen für die Anweisung SEL

Parameter	Datentyp ¹	Beschreibung
G	Bool	<ul style="list-style-type: none"> 0 wählt IN0 1 wählt IN1
IN0, IN1	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Eingänge
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Ausgang

¹ Die Eingangsvariablen und die Ausgangsvariable müssen vom gleichen Datentyp sein.

Bedingungscode: ENO ist nach der Ausführung der Anweisung SEL immer WAHR.

Tabelle 8-181 Anweisung MUX (Multiplexen)

KOP/FUP	SCL	Beschreibung
	<pre> out := MUX(k:=_unit_in, in1:=variant_in, in2:=variant_in, [...in32:=variant_in ,] inelse:=variant_in); </pre>	Die Anweisung MUX kopiert einen von vielen Eingangswerten, abhängig vom Wert des Parameters K, in den Parameter OUT. Überschreitet der Wert von Parameter K ($INn - 1$), so wird der Wert des Parameters ELSE in den Parameter OUT kopiert.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.



Um einen Eingang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".

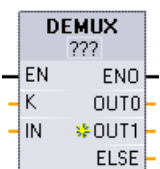
Um einen Eingang zu löschen, klicken Sie bei einem der vorhandenen Parameter IN mit der rechten Maustaste auf den Eingangsanschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 8-182 Datentypen für die Anweisung MUX

Parameter	Datentyp	Beschreibung
K	UInt	<ul style="list-style-type: none"> • 0 wählt IN1 • 1 wählt IN2 • n wählt INn
IN0, IN1, .. INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Eingänge
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Eingang Ersatzwert (optional)
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Ausgang

¹ Die Eingangsvariablen und die Ausgangsvariable müssen vom gleichen Datentyp sein.

Tabelle 8-183 Anweisung DEMUX (Demultiplexen)

KOP/FUP	SCL	Beschreibung
	<pre> DEMUX (k:=_unit_in, in:=variant_in, out1:=variant_in, out2:=variant_in, [...out32:=variant_i n,] outelse:=variant_in) ; </pre>	DEMUX kopiert den Wert der Adresse, die dem Parameter IN zugewiesen ist, in einen von vielen Ausgängen. Der Wert des Parameters K gibt an, welcher Ausgang als Ziel des Werts IN ausgewählt wird. Ist der Wert von K größer als die Zahl (OUTn - 1), wird der Wert IN in die Adresse kopiert, die dem Parameter ELSE zugewiesen ist.

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.



Um einen Ausgang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss und wählen den Befehl "Ausgang einfügen".

Um einen Ausgang zu löschen, klicken Sie bei einem der vorhandenen Parameter OUT mit der rechten Maustaste auf den Ausgangsanschluss (sofern mehr als die zwei ursprünglichen Ausgänge vorhanden sind) und wählen den Befehl "Löschen".

Tabelle 8-184 Datentypen für die Anweisung DEMUX

Parameter	Datentyp ¹	Beschreibung
K	UInt	Auswahlwert: <ul style="list-style-type: none"> • 0 wählt OUT1 • 1 wählt OUT2 • n wählt OUTn
IN	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Eingang
OUT0, OUT1, .. OUTn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Ausgänge
ELSE	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord, Time, Date, TOD, Char, WChar	Ausgang ersetzen, wenn K größer als (OUTn - 1) ist

¹ Die Eingangsvariable und die Ausgangsvariablen müssen vom gleichen Datentyp sein.


Tabelle 8-185 ENO-Zustand für die Anweisungen MUX und DEMUX

ENO	Bedingung	Ergebnis OUT
1	Kein Fehler	MUX: Ausgewählter IN-Wert wird in OUT kopiert DEMUX: IN-Wert wird in ausgewählten OUT kopiert
0	MUX: K ist größer als die Anzahl der Eingänge -1	<ul style="list-style-type: none"> Kein ELSE angegeben: OUT wird nicht verändert, ELSE angegeben, ELSE-Wert wird OUT zugewiesen
	DEMUX: K ist größer als die Anzahl der Ausgänge -1	<ul style="list-style-type: none"> Kein ELSE angegeben: Ausgänge werden nicht verändert, ELSE angegeben, IN-Wert wird in ELSE kopiert

8.10 Schieben und Rotieren

8.10.1 Anweisungen SHL (Rechts schieben) und SHL (Links schieben)

Tabelle 8-186 Anweisungen SHR und SHL

KOP/FUP	SCL	Beschreibung
	<pre> out := SHR(in:=_variant_in_, n:=_uint_in_); out := SHL(in:=_variant_in_, n:=_uint_in_); </pre>	<p>Mit den Schiebeanweisungen (SHL und SHR) schieben Sie das Bitmuster von Parameter IN. Das Ergebnis wird dem Parameter OUT zugewiesen. Parameter N gibt die Anzahl der geschobenen Bitpositionen an:</p> <ul style="list-style-type: none"> SHR: Bitmuster nach rechts schieben SHL: Bitmuster nach links schieben

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 8-187 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Ganzzahlen	Bitmuster, das verschoben werden soll
N	USInt, UDint	Anzahl der Bitpositionen, die verschoben werden sollen
OUT	Ganzzahlen	Bitmuster nach dem Schieben

- Bei N=0 wird nicht geschoben. Der Wert von IN wird OUT zugewiesen.
- In die Bitpositionen, die von der Schiebeanweisung geleert wurden, werden Nullen geschoben.

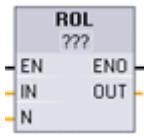
- Falls die Anzahl der zu schiebenden Positionen (N) die Anzahl der Bits im Zielwert überschreitet (8 bei Byte, 16 bei Word, 32 bei DWord), werden alle ursprünglichen Bitwerte hinausgeschoben und durch Nullen ersetzt (OUT wird Null zugewiesen).
- ENO ist bei den Schiebeanweisungen immer WAHR.

Tabelle 8-188 Beispiel: SHL mit Word-Daten

Schiebt die Bits eines Worts nach links, indem von rechts Nullen eingefügt werden (N = 1)			
IN	1110 0010 1010 1101	Wert von OUT vor dem ersten Schieben:	1110 0010 1010 1101
		Nach dem ersten Schieben links:	1100 0101 0101 1010
		Nach dem zweiten Schieben links:	1000 1010 1011 0100
		Nach dem dritten Schieben links:	0001 0101 0110 1000

8.10.2 Anweisungen ROR (Rechts rotieren) und ROL (Links rotieren)

Tabelle 8-189 Anweisungen ROR und ROL

KOP/FUP	SCL	Beschreibung
	<pre> out := ROL(in:=_variant_in_, n:=_uint_in_); out := ROR(in:=_variant_in_, n:=_uint_in_); </pre>	<p>Mit den Rotieranweisungen (ROR und ROL) können Sie das Bitmuster von Parameter IN rotieren. Das Ergebnis wird dem Parameter OUT zugewiesen. Parameter N legt die Zahl der rotierten Bitpositionen fest.</p> <ul style="list-style-type: none"> • ROR: Bitmuster nach rechts rotieren • ROL: Bitmuster nach links rotieren

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 8-190 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	Ganzzahlen	Bitmuster, das rotiert werden soll
N	USInt, UDint	Anzahl der Bitpositionen, die rotiert werden sollen
OUT	Ganzzahlen	Bitmuster nach dem Rotieren

- Bei N=0 wird nicht rotiert. Der Wert von IN wird OUT zugewiesen.
- Bitdaten, die auf einer Seite des Zielwerts hinausrotiert werden, werden auf der anderen Seite des Zielwerts hineinrotiert, so dass keine ursprünglichen Bitwerte verlorengehen.
- Falls die Anzahl der zu rotierenden Bitpositionen (N) die Anzahl der Bits im Zielwert überschreitet (8 bei Byte, 16 bei Word, 32 bei DWord), wird trotzdem rotiert.
- ENO ist nach der Ausführung der Rotieranweisungen immer WAHR.

Tabelle 8-191 Beispiel: ROR mit Word-Daten

Bits rechts hinausrotieren und links hineinrotieren (N = 1)			
IN	0100 0000 0000 0001	Wert von OUT vor dem ersten Rotieren:	0100 0000 0000 0001
		Nach dem ersten Rotieren rechts:	1010 0000 0000 0000
		Nach dem zweiten Rotieren rechts:	0101 0000 0000 0000

Erweiterte Anweisungen

9.1 Datums-, Uhrzeit- und Uhrfunktionen

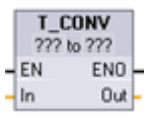
9.1.1 Datums- und Uhrzeitanweisungen

Verwenden Sie die Datum- und Uhrzeitanweisungen für Kalender- und Zeitberechnungen.

- T_CONV konvertiert einen Wert in oder aus (Datum- und Zeit-Datentypen) und (Byte-, Wort- und Doppelwort-Datentypen).
- T_ADD addiert Time- und DTL-Werte: (Time + Time = Time) oder (DTL + Time = DTL)
- T_SUB subtrahiert Time- und DTL-Werte: (Time - Time = Time) oder (DTL - Time = DTL)
- T_DIFF gibt die Differenz zwischen zwei DTL-Werten als Time-Wert aus: DTL - DTL = Time
- T_COMBINE verknüpft einen Date-Wert und einen Time_and_Date-Wert, um einen DTL-Wert zu erhalten

Informationen zum Format von DTL- und Time-Daten finden Sie im Abschnitt zu den Uhrzeit- und Datums-Datentypen (Seite 109).

Tabelle 9-1 Anweisung T_CONV (Zeiten umwandeln und extrahieren)

KOP/FUP	SCL-Beispiel	Beschreibung
	<pre>out := DINT_TO_TIME (in:=_variant_in); out := TIME_TO_DINT (in:=_variant_in);</pre>	T_CONV konvertiert einen Wert in oder aus (Datum- und Zeit-Datentypen) und (Byte-, Wort- und Doppelwort-Datentypen).

- ¹ In KOP- und FUP-Boxen: Klicken Sie auf die "???" und wählen Sie die Datentypen für Quelle und Ziel aus der Klappliste aus.
- ² In SCL: Ziehen Sie die Anweisung T_CONV aus dem Anweisungsverzeichnis in den Programmiereditor. Wählen Sie dann die Datentypen für Quelle und Ziel aus.

Tabelle 9-2 Gültige Datentypen für T_CONV-Umwandlungen

Datentyp IN (oder OUT)	Datentypen OUT (oder IN)
TIME (Millisekunden)	DInt, Int, SInt, UDInt, UInt, USInt, TOD Nur in SCL: Byte, Word, Dword
DATE (Anzahl der Tage seit 1. Januar 1990)	DInt, Int, SInt, UDInt, UInt, USInt, DTL Nur in SCL: Byte, Word, Dword
TOD (Millisekunden seit Mitternacht - 24:00:00.000)	DInt, Int, SInt, UDInt, UInt, USInt, TIME, DTL Nur in SCL: Byte, Word, Dword

Hinweis

Mit T_CONV eine größere Datengröße in eine kleinere Datengröße umwandeln


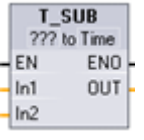
Wenn Sie einen größeren Datentyp mit mehr Bytes in einen kleineren Datentyp mit weniger Bytes umwandeln, können Datenwerte abgeschnitten werden. Tritt dieser Fehler auf, wird ENO auf 0 gesetzt.

Umwandlung aus dem/in den Datentyp DTL

Der Datentyp DTL (Date and Time Long) enthält Daten für Jahr, Monat, Datum und Uhrzeit. DTL-Daten können in/aus Datentypen DATE und TOD umgewandelt werden. Die DTL-Umwandlung nur mit Datentyp DATE betrifft jedoch nur die Werte für Jahr, Monat und Tag. Eine DTL-Umwandlung mit TOD-Daten betrifft nur die Werte für Stunden, Minuten und Sekunden.

Wenn T_CONV in DTL umwandelt, werden die nicht betroffenen Datenelemente im DTL-Format nicht verändert.

Tabelle 9-3 Anweisungen T_ADD (Zeiten addieren) und T_SUB (Zeiten subtrahieren)

KOP/FUP	SCL	Beschreibung
	<pre>out := T_ADD(in1:=_variant_in, in2:=_time_in);</pre>	<p>T_ADD addiert den Wert aus Eingang IN1 (Datentyp DTL oder Time) zum Wert in Eingang IN2 (Datentyp Time). Der Parameter OUT liefert das Ergebnis als Wert vom Datentyp DTL oder Time. Zwei Datentyp-Anweisungen sind möglich:</p> <ul style="list-style-type: none"> • Time + Time = Time • DTL + Time = DTL
	<pre>out := T_SUB(in1:=_variant_in, in2:=_time_in);</pre>	<p>T_SUB subtrahiert den Wert aus Eingang IN2 (Datentyp Time) vom Wert von IN1 (Datentyp DTL oder Time). Der Parameter OUT liefert den Differenzwert als Datentyp DTL oder Time. Zwei Datentyp-Anweisungen sind möglich.</p> <ul style="list-style-type: none"> • Time - Time = Time • DTL - Time = DTL

¹ In KOP und FUP: Klicken Sie auf die "???" und wählen Sie die Datentypen in der Klappliste aus.

Tabelle 9-4 Datentypen für die Parameter von T_ADD und T_SUB

Parameter und Datentyp	Datentyp	Beschreibung
IN1 ¹	IN	DTL, Time
IN2	IN	Time
OUT	OUT	DTL, Time

¹ Wählen Sie den Datentyp für IN1 aus der Klappliste unter dem Namen der Anweisung aus. Durch die Auswahl des Datentyps für IN1 wird auch der Datentyp für den Parameter OUT gesetzt.

Tabelle 9-5 Anweisung T_DIFF (Zeitdifferenz)


KOP/FUP	SCL	Beschreibung
	<pre>out := T_DIFF(in1:=_DTL_in, in2:=_DTL_in);</pre>	<p>T_DIFF subtrahiert den DTL-Wert (IN2) vom DTL-Wert (IN1). Der Parameter OUT liefert den Differenzwert als Datentyp Time.</p> <ul style="list-style-type: none"> DTL - DTL = Time

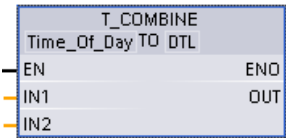
Tabelle 9-6 Datentypen für die Parameter von T_DIFF

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	DTL
IN2	IN	DTL
OUT	OUT	Time

Bedingungscode: ENO = 1 bedeutet, dass kein Fehler aufgetreten ist. ENO = 0 und Parameter OUT = 0 Fehler:

- Ungültiger DTL-Wert
- Ungültiger Time-Wert

Tabelle 9-7 Anweisung T_COMBINE (Zeiten verknüpfen)

KOP/FUP	SCL	Beschreibung
	<pre>out := CONCAT_DATE_TOD(In1 := _date_in, In2 := _tod_in);</pre>	<p>T_COMBINE verknüpft einen Date-Wert und einen Time_of_Day-Wert, um einen DTL-Wert zu erhalten.</p>

¹ Beachten Sie, dass die Anweisung T_COMBINE in den erweiterten Anweisungen der Funktion CONCAT_DATE_TOD in SCL entspricht.

Tabelle 9-8 Datentypen für die Parameter von T_COMBINE

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	Date
IN2	IN	Time_of_Day
OUT	OUT	DTL

9.1.2 Uhrzeitfunktionen

⚠️ WARNUNG

Gefahr unbefugter Zugriffe auf Ihre Netzwerke durch NTP-Synchronisation (Network Time Protocol)

Wenn ein Angreifer über die NTP-Synchronisation (Network Time Protocol) auf Ihre Netzwerke zugreifen kann, kann er Ihnen möglicherweise durch Veränderung der CPU-Systemzeit die Kontrolle über Ihren Prozess entziehen. Durch eine solche Übernahme der Prozesskontrolle kann es zu Tod, schweren Verletzungen oder Sachschäden kommen.


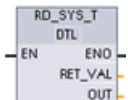
Die NTP-Client-Funktion der S7-1200 CPU ist standardmäßig deaktiviert und ermöglicht im aktivierten Zustand lediglich konfigurierten IP-Adressen, als NTP-Server zu fungieren. Die CPU deaktiviert diese Funktion standardmäßig, und Sie müssen die Funktion konfigurieren, um ferngesteuerte Korrekturen der CPU-Systemzeit zu gestatten.

Die S7-1200 CPU unterstützt Uhrzeitalarme und Uhrzeitoperationen, die von einer korrekten CPU-Systemzeit abhängig sind. Wenn Sie NTP konfigurieren und die Uhrzeitsynchronisierung von einem Server akzeptieren, müssen Sie sicherstellen, dass es sich bei dem Server um eine vertrauenswürdige Quelle handelt. Ist dies nicht der Fall, kann es zu einer Sicherheitsverletzung kommen, bei der ein unbekannter Benutzer die CPU-Systemzeit verändern und Ihnen dadurch die Kontrolle über Ihren Prozess entziehen kann.

Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren Operational Guidelines für Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) auf der Website "Service & Support" von Siemens.

Mit den Uhrzeitanweisungen können Sie die Systemuhr der CPU einstellen und lesen. Der Datentyp DTL (Seite 109) stellt Datum- und Uhrzeitwerte bereit.

Tabelle 9-9 Anweisungen für die Systemzeit

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := WR_SYS_T(in:=_DTL_in_);</pre>	<p>WR_SYS_T (Uhrzeit einstellen) stellt die Echtzeituhr in der CPU anhand des DTL-Werts von Parameter IN ein. Dieser Zeitwert berücksichtigt weder die jeweilige Ortszeit noch die Sommer-/Winterzeitschaltung.</p>
	<pre>ret_val := RD_SYS_T(out=>_DTL_out);</pre>	<p>Mit RD_SYS_T (Uhrzeit lesen) wird die aktuelle Systemzeit der CPU ausgelesen. Dieser Zeitwert berücksichtigt weder die jeweilige Ortszeit noch die Sommer-/Winterzeitschaltung.</p>

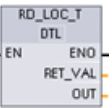

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := RD_LOC_T(out=>_DTL_out);</pre>	RD_LOC_T (Lokalzeit lesen) stellt die aktuelle Lokalzeit der CPU als Datentyp DTL bereit. Dieser Uhrzeitwert entspricht der lokalen Zeitzone sowie der Anpassung an Sommer-/Winterzeit (sofern konfiguriert).
	<pre>ret_val := WR_LOC_T(LOCTIME:=DTL_in_, DST:_in_;</pre>	WR_LOC_T (Lokalzeit schreiben) stellt das Datum und die Uhrzeit der CPU-Uhr ein. Die Datums- und Uhrzeitangaben weisen Sie über den Parameter LOCTIME mit dem Datentyp DTL als Ortszeit zu. Die Anweisung berechnet die Systemzeit mit Hilfe der DB-Struktur "TimeTransformationRule (Seite 332)". Die Detailgenauigkeit der Zeitangaben für die Ortszeit und die Systemzeit ist produktspezifisch und beträgt mindestens eine Millisekunde. Sind Eingangswerte am Parameter LOCTIME kleiner als von der CPU unterstützt, werden diese Werte während der Berechnung der Systemzeit aufgerundet. Hinweis: Sie müssen die Eigenschaften "Uhrzeit" in der CPU-Gerätekonfiguration einstellen (Zeitzone, Aktivierung der Sommerzeit, Beginn und Ende der Sommerzeit). Andernfalls kann WR_LOC_T den Wechsel zur Sommerzeit nicht interpretieren.

Tabelle 9-10 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	DTL	Einzustellende Uhrzeit für die Systemuhr der CPU
OUT	OUT	DTL	RD_SYS_T: Aktuelle Systemzeit der CPU RD_LOC_T: Aktuelle Lokalzeit, einschließlich Anpassung an Sommer-/Winterzeit (sofern konfiguriert)
LOCTIME	IN	DTL	WR_LOC_T: Lokale Uhrzeit
DST	IN	BOOL	WR_LOC_T: Daylight Saving Time (Sommerzeit) wird nur während der "doppelten Stunde" ausgewertet, wenn die Uhr zur Sommerzeit wechselt. <ul style="list-style-type: none"> TRUE = Sommerzeit (erste Stunde) FALSE = Winterzeit (zweite Stunde)
RET_VAL	OUT	Int	Ausführungsbedingung

- Zur Berechnung der Lokalzeit werden die Zeitzone und die Sommer-/Winterzeitschaltungen, die Sie im Register "Allgemein" der Gerätekonfiguration unter "Uhrzeit" eingegeben haben, verwendet.
- Die Zeitzone ist bezogen auf die UTC- oder GMT-Systemzeit.
- Für die Sommerzeitschaltung sind Monat, Woche, Tag und Stunde der Umstellung auf Sommerzeit einzugeben.

- Für die Winterzeitschaltung sind ebenfalls Monat, Woche, Tag und Stunde der Umstellung auf Winterzeit einzugeben.
- Die Zeitzonendifferenz gilt immer bezogen auf die Systemzeit. Die Sommerzeitverschiebung gilt nur, wenn eine Umstellung auf Sommerzeit stattfindet.

Hinweis

Anfangszeit von Sommer- und Winterzeit einstellen

Für die Eigenschaften "Uhrzeit" unter "Anfangszeit von Sommer- und Winterzeit" in der CPU-Gerätekonfiguration müssen Sie Ihre lokale Uhrzeit angeben.

Bedingungscode: ENO = 1 bedeutet, dass kein Fehler aufgetreten ist. ENO = 0 bedeutet, dass ein Ausführungsfehler aufgetreten ist und ein Bedingungscode am Ausgang RET_VAL ansteht.

RET_VAL (W#16#....)	Beschreibung
0000	Die aktuelle Lokalzeit wird als Winterzeit angegeben.
0001	Es wurde die Sommerzeit wurde konfiguriert und die aktuelle Lokalzeit wird als Sommerzeit angegeben.
8080	Ortszeit nicht verfügbar oder LOCTIME-Wert ist ungültig.
8081	Vom Parameter LOCTIME zugewiesener unzulässiger Jahreswert oder Zeitwert ist ungültig
8082	Unzulässiger Monatswert (Byte 2 im DTL-Format)
8083	Unzulässiger Tageswert (Byte 3 im DTL-Format)
8084	Unzulässiger Stundenwert (Byte 5 im DTL-Format)
8085	Unzulässiger Minutenwert (Byte 6 im DTL-Format)
8086	Unzulässiger Sekundenwert (Byte 7 im DTL-Format)
8087	Unzulässiger Nanosekundenwert (Byte 8 bis 11 im DTL-Format)
8089	Zeitwert nicht vorhanden (Stunde bei Wechsel zur Sommerzeit bereits vergangen)
80B0	Die Echtzeituhr ist ausgefallen
80B1	Die Struktur "TimeTransformationRule" wurde nicht definiert.

9.1.3 Datenstruktur TimeTransformationRule

Beschreibung

Die Regeln für den Wechsel zwischen Sommer- und Winterzeit sind in der Struktur TimeTransformationRule festgelegt. Die Struktur ist wie folgt:

Name	Datentyp	Beschreibung
TimeTransformationRule	STRUCT	
Bias	INT	Zeitunterschied zwischen Ortszeit und UTC [Minuten] Bereich: -1439 bis 1439
DaylightBias	INT	Zeitunterschied zwischen Sommerzeit und Winterzeit [Minuten] Bereich: 0 bis 60
DaylightStartMonth	USINT	Monat der Umstellung auf Sommerzeit Bereich: 1 bis 12

Name	Datentyp	Beschreibung
DaylightStartWeek	USINT	Woche der Umstellung auf Sommerzeit 1 = Erstes Auftreten des Wochentags im Monat, ..., 5 = Letztes Auftreten des Wochentags im Monat
DaylightStartWeekday	USINT	Wochentag der Umstellung auf Sommerzeit: 1 = Sonntag
DaylightStartHour	USINT	Stunde der Umstellung auf Sommerzeit Bereich: 0 bis 23
DaylightStartMinute	USINT	Minute der Umstellung auf Sommerzeit Bereich: 0 bis 59
StandardStartMonth	USINT	Monat der Umstellung auf Winterzeit Bereich: 1 bis 12
StandardStartWeek	USINT	Woche der Umstellung auf Winterzeit 1 = Erstes Auftreten des Wochentags im Monat, ..., 5 = Letztes Auftreten des Wochentags im Monat
StandardStartWeekday	USINT	Wochentag der Umstellung auf Winterzeit: 1 = Sonntag
StandardStartHour	USINT	Stunde der Umstellung auf Winterzeit Bereich: 0 bis 23
StandardStartMinute	USINT	Minute der Umstellung auf Winterzeit Bereich: 0 bis 59
TimeZoneName	STRING[80]	Name der Zeitzone: "(GMT +01:00) Amsterdam, Berlin, Bern, Rom, Stockholm, Wien "

9.1.4 SET_TIMEZONE (Zeitzone setzen)

Tabelle 9-11 Anweisung SET_TIMEZONE

KOP/FUP	SCL	Beschreibung
<pre> graph LR subgraph SET_TIMEZONE EN --> EN_IN[EN] REQ --> REQ_IN[REQ] Timezone --> TZ_IN[Timezone] EN_IN --> END_OUT[END] REQ_IN --> DONE_OUT[DONE] TZ_IN --> BUSY_OUT[BUSY] END_OUT --> END_OUT DONE_OUT --> DONE_OUT BUSY_OUT --> BUSY_OUT ERROR_OUT[ERROR] --> ERROR_OUT[ERROR] STATUS_OUT[STATUS] --> STATUS_OUT[STATUS] end </pre>	<pre> "SET_TIMEZONE_DB" (REQ:=_bool_in, Timezone:=_struct_in, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	Stellt die Parameter für lokale Zeitzone und Sommer-/Winterzeitumschaltung ein, anhand deren die CPU-Systemzeit in die jeweilige Ortszeit umgewandelt wird.

¹ Im SCL-Beispiel ist "SET_TIMEZONE_DB" der Name des Instanz-DBs.

Tabelle 9-12 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
Timezone	IN	TimeTransformationRule

Parameter und Datentyp		Datentyp	Beschreibung
DONE	OUT	Bool	Funktion beendet
BUSY	OUT	Bool	Funktion besetzt
ERROR	OUT	Bool	Fehler erkannt
STATUS	OUT	Word	Funktionsergebnis / Fehlermeldung

Um die Zeitonenparameter für die CPU manuell zu konfigurieren, verwenden Sie die Eigenschaften "Uhrzeit" im Register "Allgemein" der Gerätekonfiguration.

Mit der Anweisung SET_TIMEZONE stellen Sie die Ortszeit ein. Die Parameter der Struktur "TimeTransformationRule (Seite 332)" weisen die lokale Zeitzone sowie die Zeiteinstellungen für die automatische Umschaltung zwischen Sommer- und Winterzeit zu.

Hinweis

Auswirkung der Anweisung SET_TIMEZONE auf den Flash-Speicher

Die Anweisung SET_TIMEZONE führt Schreibvorgänge im Flash-Speicher durch (im internen Ladespeicher oder auf einer Memory Card). Um eine Verkürzung der Lebensdauer des Flash-Speichers zu verhindern, verwenden Sie die Anweisung SET_TIMEZONE für weniger häufige Aktualisierungen.

Bedingungscode: ENO = 1 bedeutet, dass kein Fehler aufgetreten ist. ENO = 0 bedeutet, dass ein Ausführungsfehler aufgetreten ist und ein Bedingungscode am Ausgang STATUS ansteht.

STATUS (W#16#....)	Beschreibung
0	Kein Fehler
7000	Keine laufende Auftragsbearbeitung
7001	Start der Auftragsbearbeitung. Parameter BUSY = 1, DONE = 0
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; BUSY auf Wert "1".
808x	Fehler an der x.ten Komponente: Zum Beispiel zeigt 8084 an, dass DaylightStartWeekif kein Wert von 1 bis 5 ist.

9.1.5 RTM (Betriebsstundenzähler)

Tabelle 9-13 Anweisung RTM

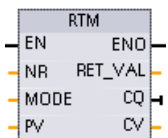
KOP/FUP	SCL	Beschreibung
	<pre>RTM(NR:=_uint_in_, MODE:=_byte_in_, PV:=_dint_in_, CQ=>_bool_out_, CV=>_dint_out_);</pre>	<p>Die Anweisung RTM (Betriebsstundenzähler) kann die Betriebsstundenzähler in der CPU setzen, starten, stoppen und lesen.</p>

Tabelle 9-14 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
NR	IN	UInt	Nummer des Betriebsstundenzählers: (mögliche Werte: 0..9)
MODE	IN	Byte	Nummer des RTM-Ausführungsmodus: <ul style="list-style-type: none"> • 0 = Werte abrufen (der Zustand wird dann in CQ und der aktuelle Wert in CV geschrieben) • 1 = Starten (ab dem letzten Zählerwert) • 2 = Stoppen • 4 = Setzen (auf den in PV angegebenen Wert) • 5 = Setzen (auf den in PV angegebenen Wert) und dann starten • 6 = Setzen (auf den in PV angegebenen Wert) und dann stoppen • 7 = Alle RTM-Werte der CPU auf der Memory Card (MC) speichern
PV	IN	DInt	Stundenwert für den angegebenen Betriebsstundenzähler voreinstellen
RET_VAL	OUT	Int	Funktionsergebnis / Fehlermeldung
CQ	OUT	Bool	Zustand Betriebsstundenzähler (1 = läuft)
CV	OUT	DInt	Aktueller Betriebsstundenwert für den angegebenen Betriebsstundenzähler

Die CPU betreibt bis zu 10 Betriebsstundenzähler, um die Betriebsstunden von kritischen Untersystemen der Steuerung zu verfolgen. Sie müssen die einzelnen Betriebsstundenzähler mit einer RTM-Anweisung je Zähler starten. Wenn die CPU von RUN in STOP wechselt, werden alle Betriebsstundenzähler gestoppt. Einzelne Zähler können Sie auch mit Ausführungsmodus 2 der RTM-Anweisung stoppen.

Wenn eine CPU von STOP in RUN wechselt, müssen Sie die Betriebsstundenzähler mit einer RTM-Anweisung je Zähler neu starten. Ist der Wert eines Betriebsstundenzählers größer als 2.147.483.647 Stunden, wird der Zählbetrieb beendet und ein Überlauffehler gemeldet. Um einen Zähler zurückzusetzen oder zu ändern, müssen Sie die Anweisung RTM einmal pro Zähler ausführen.

Ein Spannungsausfall oder Neustart der CPU bewirkt einen Abschaltvorgang, bei dem die aktuellen Werte der Betriebsstundenzähler im remanenten Speicher abgelegt werden. Beim Anlauf der CPU werden die gespeicherten Werte der Betriebsstundenzähler erneut in die Zähler geladen, so dass die vorherigen Betriebsstunden nicht verloren sind. Die Betriebsstundenzähler müssen neu gestartet werden, um weitere Betriebsstunden zu zählen.

Ihr Programm kann zudem über Ausführungsmodus 7 der RTM-Anweisung die Werte der Betriebsstundenzähler auf einer Memory Card speichern. Zum Zeitpunkt der Ausführung von Modus 7 der RTM-Anweisung werden die Zustände aller Betriebsstundenzähler auf der Memory Card gespeichert. Diese gespeicherten Werte können im Lauf der Zeit durch Start- und Stoppvorgänge der Betriebsstundenzähler im Programmverlauf inkorrekt werden. Sie müssen die Werte auf der Memory Card regelmäßig aktualisieren, um wichtige Laufzeitergebnisse zu erfassen. Der Vorteil dabei, die RTM-Werte auf der Memory Card zu speichern, ist der, dass Sie die Memory Card in eine Ersatz-CPU einschieben können, wo Ihr Programm und die gespeicherten RTM-Werte dann zur Verfügung stehen. Wenn Sie die RTM-Werte nicht auf einer

Memory Card gespeichert haben, sind die Betriebsstundenwerte verloren (bei Nutzung einer Ersatz-CPU).

Hinweis

Vermeiden Sie übermäßige Programmaufrufe bei Schreibvorgängen auf der Memory Card

Halten Sie die Anzahl der Schreibvorgänge auf der Memory Card möglichst gering, um die Lebensdauer der Memory Card zu verlängern.

Tabelle 9-15 Bedingungs-codes

RET_VAL (W#16#....)	Beschreibung
0	Kein Fehler
8080	Falsche Nummer des Betriebsstundenzählers
8081	Ein negativer Wert wurde an den Parameter PV übergeben
8082	Überlauf des Betriebsstundenzählers
8091	Der Eingangsparameter MODE enthält einen unzulässigen Wert
80B1	Wert kann nicht auf MC gespeichert werden (MODUS = 7)

9.2 Zeichenketten- und Zeichenanweisungen

9.2.1 Datentyp String

Datentyp STRING

Daten des Typs String werden mit einem Header aus 2 Bytes gefolgt von bis zu 254 Zeichenbytes im ASCII-Code gespeichert. Ein Header für String-Daten hat zwei Längenbytes. Das erste Byte enthält die maximale Länge, die bei der Initialisierung einer Zeichenkette in eckigen Klammern angegeben bzw. auf 254 voreingestellt ist. Das zweite Header-Byte enthält die tatsächliche Länge, die der Anzahl gültiger Zeichen in der Zeichenkette entspricht. Die tatsächliche Länge muss kleiner oder gleich der maximalen Länge sein. Die Anzahl gespeicherter Bytes für das String-Format ist um 2 Bytes größer als die maximale Länge.

String-Daten initialisieren

Die Ein- und Ausgangsdaten des Datentyps String müssen vor der Ausführung von Anweisungen als gültige Zeichenketten im Speicher initialisiert werden.

Gültige String-Daten

Gültige String-Daten haben eine maximale Länge, die größer als null und kleiner als 255 sein muss. Die tatsächliche Länge muss kleiner oder gleich der maximalen Länge sein.

String-Daten können den Speicherbereichen E oder A nicht zugewiesen werden.

Weitere Informationen finden Sie unter: Format des Datentyps String (Seite 110).

9.2.2 S_MOVE (Zeichenkette verschieben)

Tabelle 9-16 Anweisung Zeichenkette verschieben

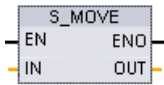
KOP/FUP	SCL	Beschreibung
	<code>out := in;</code>	Zeichenkette der Quelle IN in die Adresse OUT kopieren. Die Ausführung S_MOVE wirkt sich nicht auf den Inhalt der Quellzeichenkette aus.

Tabelle 9-17 Datentypen für die Parameter

Parameter	Datentyp	Beschreibung
IN	String	Quellzeichenkette
OUT	String	Zieladresse

Wenn die tatsächliche Länge der Zeichenkette am Eingang IN die maximale Länge einer am Ausgang OUT gespeicherten Zeichenkette überschreitet, wird der Teil der Zeichenkette IN, der in OUT passt, kopiert.

9.2.3 Anweisungen für die Zeichenkettenkonvertierung

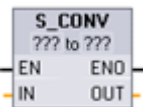
9.2.3.1 Anweisungen S_CONV, STRG_VAL und VAL_STRG (In/von Zeichenkette und Zahlenwert umwandeln)

Mit den folgenden Anweisungen können Sie Zahlenzeichenketten in Zahlenwerte und Zahlenwerte in Zahlenzeichenketten umwandeln:

- Mit S_CONV wird eine Zahlenzeichenkette in einen Zahlenwert oder ein Zahlenwert in eine Zahlenzeichenkette umgewandelt.
- Mit STRG_VAL wird eine Zahlenzeichenkette in einen Zahlenwert mit Formatoptionen umgewandelt.
- Mit VAL_STRG wird ein Zahlenwert in eine Zahlenzeichenkette mit Formatoptionen umgewandelt.

S_CONV (Zeichenkette umwandeln)

Tabelle 9-18 Anweisungen für die Zeichenkettenkonvertierung

KOP/FUP	SCL	Beschreibung
	<pre>out := <Type>_TO_<Type>(in) ;</pre>	Konvertiert eine Zeichenkette in den entsprechenden Wert bzw. einen Wert in die entsprechende Zeichenkette. Die Anweisung S_CONV verfügt über keine Optionen zur Formatierung des Ausgangswerts. Daher ist S_CONV einfacher, aber auch weniger flexibel als die Anweisungen STRG_VAL und VAL_STRG.

- ¹ In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.
- ² In SCL: Wählen Sie S_CONV aus den erweiterten Anweisungen und beantworten Sie die Abfragen der Datentypen für die Umwandlung. STEP 7 stellt daraufhin die entsprechende Umwandlungsanweisung zur Verfügung.

Tabelle 9-19 Datentypen (Zeichenkette in Zahlenwert umwandeln)

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String, WString	Eingegebene Zeichenkette
OUT	OUT	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Ausgegebener Zahlenwert

Die Umwandlung des Zeichenkettenparameters IN beginnt mit dem ersten Zeichen und wird bis zum Ende der Zeichenkette oder bis zum ersten Zeichen, das nicht "0" bis "9", "+", "-" oder "." ist, fortgesetzt. Das Ergebnis steht an der mit Parameter OUT angegebenen Adresse bereit. Passt der ausgegebene Zahlenwert nicht in den Bereich des Datentyps von OUT, so wird der Parameter OUT auf 0 und ENO auf FALSCH gesetzt. Andernfalls enthält der Parameter OUT ein gültiges Ergebnis und ENO wird auf WAHR gesetzt.

Formatregeln für die Zeichenketteneingabe:

- Enthält die Zeichenkette in IN ein Dezimalzeichen, so muss das Zeichen "." verwendet werden.
- Kommazeichen "," als Tausendertrennzeichen links vom Dezimalpunkt sind zulässig, werden aber ignoriert.
- Führende Leerzeichen werden ignoriert.

S_CONV (Umwandlung von Wert in Zeichenkette)

Tabelle 9-20 Datentypen (Zahlenwert in Zeichenkette umwandeln)

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String, WString, Char, WChar, SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Eingegebener Zahlenwert
OUT	OUT	String, WString	Ausgegebene Zeichenkette

Ein ganzzahliger Wert mit oder ohne Vorzeichen oder ein Gleitpunktwert IN wird in die entsprechende Zeichenkette umgewandelt und steht in Parameter OUT bereit. Der Parameter OUT muss eine gültige Zeichenkette referenzieren, bevor die Umwandlung erfolgt. Eine gültige

Zeichenkette besteht aus der maximalen Zeichenkettenlänge im ersten Byte, der tatsächlichen Zeichenkettenlänge im zweiten Byte und den tatsächlichen Zeichen in den nächsten Bytes. Die umgewandelte Zeichenkette ersetzt Zeichen in der Zeichenkette OUT, beginnend mit dem ersten Zeichen, und passt das Byte mit der tatsächlichen Länge der ausgegebenen Zeichenkette entsprechend an. Das Byte mit der maximalen Länge der ausgegebenen Zeichenkette wird nicht verändert.

Wie viele Zeichen ersetzt werden, hängt von Datentyp und Zahlenwert des Parameters IN ab. Die ersetzte Zeichenzahl muss in die in Parameter OUT festgelegte Zeichenkettenlänge passen. Die maximale Zeichenkettenlänge (erstes Byte) der Zeichenkette in Parameter OUT muss größer oder gleich der maximal erwarteten Zahl umgewandelter Zeichen sein. Die folgende Tabelle zeigt Beispiele für die Umwandlung von Werten in Zeichenketten mit S_CONV:

Formatregeln für die Zeichenkettenausgabe:

- In Parameter OUT geschriebene Werte verwenden kein führendes "+" Zeichen.
- Es wird die Festpunktdarstellung verwendet (keine Exponentialdarstellung).
- Mit dem Punktzeichen "." wird der Dezimalpunkt dargestellt, wenn der Parameter IN vom Datentyp Real ist.
- Die Werte werden in der ausgegebenen Zeichenkette rechtsbündig angeordnet und ihnen stehen Leerzeichen voran, die leere Zeichenpositionen belegen.


Tabelle 9-21 Maximale Zeichenkettenlänge für jeden Datentyp

Datentyp IN	Von S_CONV zugeordnete Zeichenpositionen	Beispiel für eine umgewandelte Zeichenkette ¹	Gesamtlänge der Zeichenkette mit den Bytes für maximale und tatsächliche Länge
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 Die Zeichen "x" stellen Leerzeichen dar, die in dem umgewandelten Wert zugeordneten rechtsbündigen Feld die leeren Positionen belegen.

STRG_VAL (Zeichenkette in Zahlenwert umwandeln)

Tabelle 9-22 Anweisung Zeichenkette in Zahlenwert wandeln

KOP/FUP	SCL	Beschreibung
	<pre>"STRG_VAL" (in:=_string_in, format:=_word_in, p:=uint_in, out=>_variant_out);</pre>	Konvertiert eine Zahlenzeichenkette in die entsprechende ganzzahlige oder Gleitpunktdarstellung.

¹ In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 9-23 Datentypen für die Anweisung STRG_VAL

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String, WString	ASCII-Zeichenkette, die umgewandelt werden soll
FORMAT	IN	Word	Optionen für Ausgangsformate
P	IN	UInt, Byte, USInt	IN: Verweis zum ersten Zeichen, das umgewandelt werden soll (erstes Zeichen = 1)
OUT	OUT	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal	Umgewandelter Zahlenwert

Die Umwandlung beginnt in der eingegebenen Zeichenkette an Zeichenversatz P und läuft bis zum Ende der Zeichenkette oder bis zum ersten Zeichen, das nicht "+", "-", ".", ";", "e", "E" oder eines der Zeichen von "0" bis "9" ist. Das Ergebnis wird an der in Parameter OUT angegebenen Adresse ausgegeben.

Die Daten vom Datentyp String müssen vor der Ausführung als gültige Zeichenkette im Speicher initialisiert werden.

Der Parameter FORMAT der Anweisung STRG_VAL ist nachstehend definiert. Die freien Bitpositionen müssen auf Null gesetzt werden.

Tabelle 9-24 Format der Anweisung STRG_VAL

Bit 16								Bit 8	Bit 7							Bit 0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f	r

- f = Darstellungsformat
- r = Dezimalpunktformat
- 1 = Exponentielle Darstellung
- 0 = Festpunktdarstellung (Kommazeichen)
- 0 = "." (Punktzeichen)

Tabelle 9-25 Werte des Parameters FORMAT


FORMAT (W#16#)	Darstellungsformat	Dezimalpunktdarstellung
0000 (Voreinstellung)	Festpunkt	"."
0001		","
0002	Exponentiell	"."
0003		","
0004 bis FFFF	Ungültige Werte	

Regeln für die STRG_VAL-Umwandlung:

- Wird das Punktzeichen "." für den Dezimalpunkt benutzt, so werden Kommazeichen "," links vom Dezimalpunkt als Tausendertrennzeichen interpretiert. Kommazeichen sind zulässig und werden ignoriert.
- Wird das Kommazeichen "," als Dezimaltrennzeichen benutzt, so werden Punktzeichen "." links vom Dezimalzeichen als Tausendertrennzeichen interpretiert. Diese Punktzeichen sind zulässig und werden ignoriert.
- Führende Leerzeichen werden ignoriert.

VAL_STRG (Zahlenwert in Zeichenkette umwandeln)

Tabelle 9-26 Anweisung Zahlenwert in Zeichenkette wandeln

KOP/FUP	SCL	Beschreibung
	<pre>"VAL_STRG" (in:=_variant_in, size:=_usint_in, prec:=_usint_in, format:=_word_in, p:=_uint_in, out=>_string_out);</pre>	Konvertiert einen ganzzahligen Wert mit oder ohne Vorzeichen oder einen Gleitpunktwert in die entsprechende Zeichenkettendarstellung.

¹ In KOP/FUP: Klicken Sie auf die "???" und wählen Sie den Datentyp in der Klappliste aus.

Tabelle 9-27 Datentypen für die Anweisung VAL_STRG

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	SInt, Int, DInt, USInt, UInt, UDInt, Real, LReal
SIZE	IN	USInt
PREC	IN	USInt
FORMAT	IN	Word
P	IN	UInt, Byte, USInt
OUT	OUT	String, WString

FORMAT (WORD)	Vorzeichen	Darstellungsformat	Dezimalpunktdarstellung
W#16#0004	"+" und "-"	Festpunkt	" "
W#16#0005			","
W#16#0006		Exponentiell	" "
W#16#0007			","
W#16#0008 bis W#16#FFFF	Ungültige Werte		

Formatregeln für Zeichenketten in Parameter OUT:

- Führende Leerzeichen werden ganz links an die Zeichenkette angefügt, wenn die umgewandelte Zeichenkette kleiner als die angegebene Größe ist.
- Ist das Vorzeichenbit von Parameter FORMAT FALSCH, so werden ganzzahlige Werte mit und ohne Vorzeichen ohne führendes Vorzeichen "+" in den Ausgangspuffer geschrieben. Vorzeichen "-" wird bei Bedarf verwendet.
<führende Leerzeichen><Ziffern ohne führende Nullen>'.<PREC-Ziffern>
- Ist das Vorzeichenbit von Parameter WAHR, so werden ganzzahlige Werte mit und ohne Vorzeichen immer mit einem führenden Vorzeichen in den Ausgangspuffer geschrieben.
<führende Leerzeichen><Vorzeichen><Ziffern ohne führende Nullen>'.<PREC-Ziffern>
- Ist als FORMAT die exponentielle Darstellung eingestellt, so werden Werte des Datentyps Real wie folgt in den Ausgangspuffer geschrieben:
<führende Leerzeichen><Vorzeichen><Ziffer> '.' <PREC-Ziffern>'E' <Vorzeichen><Ziffern ohne führende Null>
- Ist als FORMAT die Festpunktdarstellung eingestellt, so werden ganzzahlige Werte mit und ohne Vorzeichen und Werte des Datentyps Real wie folgt in den Ausgangspuffer geschrieben:
<führende Leerzeichen><Vorzeichen><Ziffern ohne führende Nullen>'.<PREC-Ziffern>
- Führende Nullen links vom Dezimalpunkt (mit Ausnahme der Ziffer direkt neben dem Dezimalpunkt) werden unterdrückt.
- Der Wert rechts vom Dezimalpunkt wird gerundet, damit er der mit Parameter PREC eingestellten Anzahl der Ziffern entspricht, die rechts vom Dezimalpunkt angegeben werden sollen.
- Die Größe der ausgegebenen Zeichenkette muss mindestens drei Bytes mehr umfassen als die Anzahl der Ziffern rechts vom Dezimalpunkt.
- Die Werte werden rechtsbündig in der ausgegebenen Zeichenkette angeordnet.

Mit ENO gemeldete Bedingungen

Tritt bei der Umwandlung ein Fehler auf, gibt die Anweisung die folgenden Ergebnisse aus:

- ENO wird auf 0 gesetzt.
- OUT wird auf 0 gesetzt oder auf den Wert, der in den Beispielen für die Umwandlung von Zeichenketten in Werte angegeben ist.
- OUT bleibt unverändert oder auf dem in den Beispielen gezeigten Wert, wenn OUT eine Zeichenkette ist.

Tabelle 9-30 ENO-Zustand

ENO	Beschreibung
1	Kein Fehler
0	Unzulässiger oder ungültiger Parameter, zum Beispiel ein Zugriff auf einen DB, der nicht existiert
0	Unzulässige Zeichenkette mit maximaler Länge von 0 oder 255
0	Unzulässige Zeichenkette, deren Länge größer als die maximale Länge ist
0	Der umgewandelte Zahlenwert ist zu groß für den für OUT angegebenen Datentyp.
0	Die maximale Zeichenkettengröße für Parameter OUT muss groß genug sein für die Zahl der mit Parameter SIZE angegebenen Zeichen, beginnend mit der Zeichenposition in Parameter P.
0	Unzulässiger Wert P, wobei P=0 oder P größer als die aktuelle Zeichenkettenlänge ist
0	Parameter SIZE muss größer als Parameter PREC sein.

Tabelle 9-31 Beispiel für die Umwandlung von Zeichenketten in Werte mit S_CONV

IN-Zeichenkette	Datentyp OUT	OUT-Wert	ENO
"123"	Int oder DInt	123	WAHR
"-00456"	Int oder DInt	-456	WAHR
"123,45"	Int oder DInt	123	WAHR
"+2345"	Int oder DInt	2345	WAHR
"00123AB"	Int oder DInt	123	WAHR
"123"	Real	123,0	WAHR
"123,45"	Real	123,45	WAHR
"1.23e-4"	Real	1,23	WAHR
"1.23E-4"	Real	1,23	WAHR
"12.345,67"	Real	12345,67	WAHR
"3.4e39"	Real	3,4	WAHR
"-3.4e39"	Real	-3,4	WAHR
"1.17549e-38"	Real	1,17549	WAHR
"12345"	SInt	0	FALSCH
"A123"	-/-	0	FALSCH
""	-/-	0	FALSCH
"++123"	-/-	0	FALSCH
"+-123"	-/-	0	FALSCH

Tabelle 9-32 Beispiele für die Umwandlung von Werten in Zeichenketten mit S_CONV

Datentyp	IN-Wert	OUT-Zeichenkette ¹	ENO
UInt	123	"xxx123"	WAHR
UInt	0	"xxxxx0"	WAHR
UDInt	12345678	"xxx12345678"	WAHR
Real	+9123,456	"xx+9.123456E+3"	WAHR

Datentyp	IN-Wert	OUT-Zeichenkette ¹	ENO
LReal	+9123,4567890123	"xx +9.1234567890123E +3"	WAHR
Real	-INF	"xxxxxxxxxxxINF"	FALSCH
Real	+INF	"xxxxxxxxxxxINF"	FALSCH
Real	NaN	"xxxxxxxxxxxNaN"	FALSCH

¹ Die Zeichen "x" stellen Leerzeichen dar, die in dem umgewandelten Wert zugeordneten rechtsbündigen Feld die leeren Positionen belegen.

Tabelle 9-33 Beispiel: STRG_VAL-Umwandlung

IN-Zeichenkette	FORMAT (W#16#....)	Datentyp OUT	OUT-Wert	ENO
"123"	0000	Int oder DInt	123	WAHR
"-00456"	0000	Int oder DInt	-456	WAHR
"123,45"	0000	Int oder DInt	123	WAHR
"+2345"	0000	Int oder DInt	2345	WAHR
"00123AB"	0000	Int oder DInt	123	WAHR
"123"	0000	Real	123,0	WAHR
"-00456"	0001	Real	-456,0	WAHR
"+00456"	0001	Real	456,0	WAHR
"123,45"	0000	Real	123,45	WAHR
"123,45"	0001	Real	12345,0	WAHR
"123,45"	0000	Real	12345,0	WAHR
"123,45"	0001	Real	123,45	WAHR
".00123AB"	0001	Real	123,0	WAHR
"1.23e-4"	0000	Real	1,23	WAHR
"1.23E-4"	0000	Real	1,23	WAHR
"1.23E-4"	0002	Real	1.23E-4	WAHR
"12.345,67"	0000	Real	12345,67	WAHR
"12.345,67"	0001	Real	12,345	WAHR
"3.4e39"	0002	Real	+INF	WAHR
"-3.4e39"	0002	Real	-INF	WAHR
"1.1754943e-38" (und kleiner)	0002	Real	0,0	WAHR
"12345"	-/-	SInt	0	FALSCH
"A123"	-/-	-/-	0	FALSCH
""	-/-	-/-	0	FALSCH
"++123"	-/-	-/-	0	FALSCH
"+-123"	-/-	-/-	0	FALSCH

Die folgenden Beispiele für VAL_STRG-Umwandlungen beruhen auf einer Zeichenkette OUT, die wie folgt initialisiert wurde:

"Current Temp = xxxxxxxxxxx C"

Dabei steht das Zeichen "x" für Leerzeichen, die dem umgewandelten Wert zugeordnet sind.

Tabelle 9-34 Beispiel: VAL_STRG-Umwandlung

Datentyp	IN-Wert	P	SIZE	FORMAT (W#16#....)	PREC	OUT-Zeichenkette	ENO
UInt	123	16	10	0000	0	Current Temp = xxxxxxxx123 C	WAHR
UInt	0	16	10	0000	2	Current Temp = xxxxxx0.00 C	WAHR
UDInt	12345678	16	10	0000	3	Current Temp = x12345.678 C	WAHR
UDInt	12345678	16	10	0001	3	Current Temp = x12345,678 C	WAHR
Int	123	16	10	0004	0	Current Temp = xxxxxx+123 C	WAHR
Int	-123	16	10	0004	0	Current Temp = xxxxxx-123 C	WAHR
Real	-0,00123	16	10	0004	4	Current Temp = xxx-0.0012 C	WAHR
Real	-0,00123	16	10	0006	4	Current Temp = -1.2300E-3 C	WAHR
Real	-INF	16	10	-/-	4	Current Temp = xxxxxx-INF C	FALSCH
Real	+INF	16	10	-/-	4	Current Temp = xxxxxx+INF C	FALSCH
Real	NaN	16	10	-/-	4	Current Temp = xxxxxxxNaN C	FALSCH
UDInt	12345678	16	6	-/-	3	Current Temp = xxxxxxxxxxx C	FALSCH

9.2.3.2 Anweisungen Strg_TO_Chars und Chars_TO_Strg (In/aus Zeichenkette und Array aus CHAR umwandeln)

Strg_TO_Chars kopiert eine ASCII-Zeichenkette in ein Array aus Zeichenbytes.

Chars_TO_Strg kopiert ein Array aus ASCII-Zeichen-Bytes in eine Zeichenkette.

Hinweis

Nur die nullbasierten Array-Typen (Array [0..n] of Char) oder (Array [0..n] of Byte) sind als Eingangsparameter Chars bei der Anweisung Chars_TO_Strg zulässig, oder als IN_OUT-Parameter Chars bei der Anweisung Strg_TO_Chars.

Tabelle 9-35 Anweisung Strg_TO_Chars

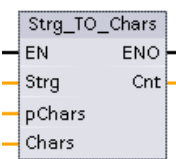
KOP/FUP	SCL	Beschreibung
	<pre>Strg_TO_Chars (Strg:=_string_in_, pChars:=_dint_in_, Cnt=>_uint_out_, Chars:=_variant_inout_) ;</pre>	<p>Die vollständige Eingangszeichenkette Strg wird in ein Array aus Zeichen am IN_OUT-Parameter Chars. kopiert.</p> <p>Die Anweisung überschreibt Bytes beginnend mit der Array-Elementnummer, die vom Parameter pChars angegeben wird.</p> <p>Zeichenketten aller unterstützten maximalen Längen (1 .. 254) können verwendet werden.</p> <p>Es wird kein Abschlusstrennzeichen geschrieben, hierfür sind Sie selbst verantwortlich. Wenn Sie das Abschlusstrennzeichen direkt nach dem letzten geschriebenen Array-Zeichen setzen möchten, verwenden Sie die nächste Array-Elementnummer [pChars+Cnt].</p>

Tabelle 9-36 Datentypen für die Parameter (Strg_TO_Chars)

Parameter und Datentyp	Datentyp	Beschreibung
Strg	IN	String, WString
pChars	IN	DInt
Chars	IN_OUT	Variante
Cnt	OUT	UInt

Tabelle 9-37 Anweisung Chars_TO_Strg

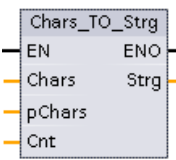
KOP/FUP	SCL	Beschreibung
	<pre>Chars_TO_Strg(Chars:=_variant_in_, pChars:=_dint_in_, Cnt:=_uint_in_, Strg=>_string_out_);</pre>	<p>Das gesamte Array oder ein Teil eines Arrays aus Zeichen wird in eine Zeichenkette kopiert.</p> <p>Die Ausgangszeichenkette muss deklariert sein, bevor Chars_TO_Strg ausgeführt wird. Die Zeichenkette wird dann von der Anweisung Chars_TO_Strg überschrieben.</p> <p>Zeichenketten aller unterstützten maximalen Längen (1 ... 254) können verwendet werden.</p> <p>Der Wert für die maximale Länge der Zeichenkette wird von der Anweisung Chars_TO_Strg nicht verändert. Das Kopieren aus dem Array in die Zeichenkette endet, wenn die maximale Länge der Zeichenkette erreicht ist.</p> <p>Das nul-Zeichen "\$00" oder der Wert 16#00 im Zeichen-Array fungiert als Trennzeichen und beendet das Kopieren von Zeichen in die Zeichenkette.</p>

Tabelle 9-38 Datentypen für die Parameter (Chars_TO_Strg)

Parameter und Datentyp		Datentyp	Beschreibung
Chars	IN	Variante	Der Parameter Chars ist ein Pointer auf das nullbasierte Array [0..n] aus Zeichen, das in eine Zeichenkette umgewandelt werden soll. Das Array kann in einem DB oder als lokale Variablen in der Bausteinschnittstelle deklariert werden. Beispiel: "DB1".MyArray verweist auf die Elementwerte MyArray [0..10] of Char in DB1.
pChars	IN	Dint	Elementnummer des ersten zu kopierenden Zeichens im Array. Array-Element [0] ist der Standardwert.
Cnt	IN	UInt	Anzahl der zu kopierenden Zeichen: 0 bedeutet alle
Strg	OUT	String, WString	Zielzeichenkette

Tabelle 9-39 ENO-Status

ENO	Beschreibung
1	Kein Fehler
0	Chars_TO_Strg: Versuch, mehr Zeichenbytes in die Ausgangszeichenkette zu kopieren als entsprechend dem Byte für die maximale Länge in der Zeichenkettendeklaration zulässig sind
0	Chars_TO_Strg: Der Nullzeichenwert (16#00) wurde im Byte-Array des Eingangszeichens gefunden.
0	Strg_TO_Chars: Versuch, mehr Zeichenbytes in das Ausgangs-Array zu kopieren als entsprechend dem Grenzwert für die Elementanzahl zulässig sind

9.2.3.3 ATH und HTA (In/aus ASCII-Zeichenkette und Hexadezimalzahl umwandeln)

Mit den Anweisungen ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln) und HTA (Hexadezimalzahl in ASCII-Zeichenkette wandeln) können Sie ASCII-Zeichen (0 bis 9 und Großbuchstaben A bis F) in die entsprechenden 4-Bit-Hexadezimaldarstellungen umwandeln und umgekehrt.

Tabelle 9-40 Anweisung ATH

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ATH(in:=_variant_in_, n:=_int_in_, out=>_variant_out_);</pre>	Konvertiert ASCII-Zeichen in gepackte Hexadezimalziffern.

Tabelle 9-41 Datentypen für die Anweisung ATH

Parametertyp		Datentyp	Beschreibung
IN	IN	Variante	Pointer zum ASCII-Zeichenbyte-Datenfeld
N	IN	UInt	Anzahl der ASCII-Zeichenbytes, die umzuwandeln sind
RET_VAL	OUT	Word	Ausführungsbedingung
OUT	OUT	Variante	Zeigt zum umgewandelten Hexadezimalbyte-Datenfeld

Die Umwandlung beginnt an der von Parameter IN festgelegten Stelle und umfasst N Bytes. Das Ergebnis wird an der in Parameter OUT angegebenen Adresse ausgegeben. Es können nur gültige ASCII-Zeichen von 0 bis 9, Kleinbuchstaben a bis f und Großbuchstaben A bis F umgewandelt werden. Alle anderen Zeichen werden in Null umgewandelt.

8-Bit ASCII-codierte Zeichen werden in 4-Bit Hexadezimalzeichen umgewandelt. Zwei ASCII-Zeichen können in ein einzelnes Byte mit zwei 4-Bit Hexadezimalzeichen umgewandelt werden.

Die Parameter IN und OUT geben Byte-Arrays, keine hexadezimalen String-Daten an. Die Umwandlung der ASCII-Zeichen und ihre Ausgabe in Hexadezimalzeichen erfolgt in der Reihenfolge, in der sie gelesen werden. Bei einer ungeraden Anzahl ASCII-Zeichen werden Nullen in das äußerste rechte Halbbyte der zuletzt umgewandelten Hexadezimalstelle geschrieben.

Tabelle 9-42 Beispiele: Umwandlung von ASCII-Zeichenketten in Hexadezimalzahlen (ATH)

IN-Zeichenbytes	N	OUT-Wert	ENO
'0a23'	4	W#16#0A23	WAHR
'123AFx1a23'	10	16#123AF01023	FALSCH
'a23'	3	W#16#A230	WAHR

Tabelle 9-43 Anweisung HTA

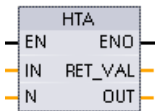
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := HTA(in:=_variant_in_, n:=_uint_in_, out=>_variant_out_);</pre>	Konvertiert gepackte Hexadezimalziffern in die entsprechenden ASCII-Zeichenbytes.

Tabelle 9-44 Datentypen für die Anweisung HTA

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	Variante
N	IN	UInt
RET_VAL	OUT	Word
OUT	OUT	Variante

Die Umwandlung beginnt an der von Parameter IN festgelegten Stelle und umfasst N Bytes. Jedes 4-Bit-Halbbyte wird in ein einzelnes 8-Bit-ASCII-Zeichen umgewandelt und erzeugt 2N-ASCII-Zeichenbytes am Ausgang. Alle 2N Bytes des Ausgangs werden als ASCII-Zeichen 0 bis 9 bzw. Großbuchstaben A bis F geschrieben. Der Parameter OUT gibt ein Byte-Array, keine Zeichenkette, an.

9.2 Zeichenketten- und Zeichenanweisungen

Jedes Halbbyte des Hexadezimalbytes wird in ein Zeichen in der Reihenfolge des Einlesens umgewandelt (das linke Halbbyte einer Hexadezimalziffer wird zuerst umgewandelt, dann das rechte Halbbyte des gleichen Bytes).

Tabelle 9-45 Beispiele: Umwandlung von Hexadezimalzahlen in ASCII-Zeichenketten (HTA)

IN-Wert	N	OUT-Zeichenbytes	ENO (ENO immer WAHR nach Ausführung von HTA)
W#16#0123	2	'0123'	WAHR
DW#16#123AF012	4	'123AF012'	WAHR

Tabelle 9-46 ATH and HTA-Bedingungscodes

RET_VAL (W#16#....)	Beschreibung	ENO
0000	Kein Fehler	WAHR
0007	Ungültiges ATH-Eingangszeichen: Es wurde ein Zeichen gefunden, bei dem es sich nicht um ein ASCII-Zeichen 0-9, einen Kleinbuchstaben a-f oder um einen Großbuchstaben A-F handelte	FALSCH
8101	Unzulässiger oder ungültiger Eingangspointer, zum Beispiel ein Zugriff auf einen DB, der nicht existiert.	FALSCH
8120	Eingangszeichenkette hat ein ungültiges Format, d. h. Max. = 0, Max. = 255, aktuell > Max. oder Länge im Pointer < Max.	FALSCH
8182	Eingangspuffer ist zu klein für N	FALSCH
8151	Datentyp für Eingangspuffer nicht zulässig	FALSCH
8301	Unzulässiger oder ungültiger Ausgangspointer, zum Beispiel ein Zugriff auf einen DB, der nicht existiert.	FALSCH
8320	Ausgangszeichenkette hat ein ungültiges Format, d. h. Max. = 0, Max. = 255, aktuell > Max. oder Länge im Pointer < Max.	FALSCH
8382	Ausgangspuffer ist zu klein für N	FALSCH
8351	Datentyp für Ausgangspuffer nicht zulässig	FALSCH

9.2.4 Zeichenkettenanweisungen

Ihr Steuerungsprogramm kann mit den folgenden Zeichenketten- und Zeichenanweisungen Meldungen für Bedieneranzeigen und Prozessprotokolle erzeugen.

9.2.4.1 MAX_LEN (Maximale Länge einer Zeichenkette)

Tabelle 9-47 Anweisung Maximale Länge

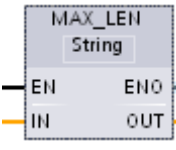
KOP/FUP	SCL	Beschreibung
 <p>The diagram shows a rectangular block labeled 'MAX_LEN String'. It has four ports: 'EN' (top-left), 'END' (top-right), 'IN' (bottom-left), and 'OUT' (bottom-right). The 'IN' and 'OUT' ports are highlighted in yellow.</p>	<pre>out := MAX_LEN(in);</pre>	<p>MAX_LEN (Maximale Länge der Zeichenkette) gibt die der Zeichenkette IN am Ausgang OUT zugewiesene maximale Länge an. Wenn während der Verarbeitung der Anweisung Fehler auftreten, wird eine leere Zeichenkettenlänge ausgegeben. Die Datentypen String und WString enthalten zwei Längen: Das erste Byte (oder Wort) gibt die maximale Länge an und das zweite Byte (oder Wort) gibt die aktuelle Länge an (dies ist die derzeitige Anzahl gültiger Zeichen).</p> <ul style="list-style-type: none"> Die maximale Länge der Zeichenkette wird für jede String- oder WString-Deklaration in eckigen Klammern angegeben. Die Anzahl der von einem String belegten Bytes ist um 2 Bytes größer als die maximale Länge. Die Anzahl der von einem WString belegten Wörter ist um 2 Wörter größer als die maximale Länge. Die aktuelle Länge stellt die Anzahl der tatsächlich verwendeten Zeichen dar. Die tatsächliche Länge muss kleiner oder gleich der maximalen Länge sein. Die aktuelle Länge wird für einen String in Bytes und für einen WString in Wörtern angegeben. <p>Mit der Anweisung MAX_LEN i rufen Sie die maximale Länge der Zeichenkette ab und mit der Anweisung LEN die aktuelle Länge einer Zeichenkette.</p>

Tabelle 9-48 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	String, WString
OUT	OUT	DInt
		Maximal zulässige Anzahl Zeichen in der Zeichenkette IN

9.2.4.2 LEN (Länge einer Zeichenkette ermitteln)

Tabelle 9-49 Anweisung LEN

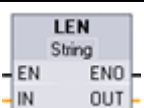
KOP/FUP	SCL	Beschreibung
 <p>The diagram shows a rectangular block labeled 'LEN String'. It has four ports: 'EN' (top-left), 'END' (top-right), 'IN' (bottom-left), and 'OUT' (bottom-right). The 'IN' and 'OUT' ports are highlighted in yellow.</p>	<pre>out := LEN(in);</pre>	<p>LEN (Länge) gibt die aktuelle Länge der Zeichenkette IN am Ausgang OUT aus. Eine leere Zeichenkette hat die Länge Null.</p>

Tabelle 9-50 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN	IN	String, WString
OUT	OUT	Int, DInt, Real, LReal
		Anzahl gültiger Zeichen in Zeichenkette IN

Tabelle 9-51 ENO-Zustand

ENO	Bedingung	OUT
1	Keine gültige Zeichenkette	Gültige Zeichenkettenlänge
0	Aktuelle Länge von IN überschreitet die maximale Länge von IN	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN passt nicht in den zugewiesenen Speicherbereich	
	Maximale Länge von IN ist 255 (unzulässige Länge)	

9.2.4.3 CONCAT (Zeichenketten verketten)

Tabelle 9-52 Anweisung Zeichenketten verketten

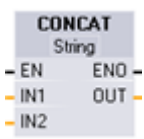
KOP/FUP	SCL	Beschreibung
	<pre>out := CONCAT(in1, in2);</pre>	CONCAT (Zeichenketten verketten) verknüpft die String-Parameter IN1 und IN2 zu einer Zeichenkette, die am Ausgang OUT bereitsteht. Nach der Verknüpfung ist Zeichenkette IN1 der linke und Zeichenkette IN2 der rechte Teil der kombinierten Zeichenkette.

Tabelle 9-53 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	String, WString
IN2	IN	String, WString
OUT	OUT	String, WString
		Kombinierte Zeichenkette (Zeichenkette 1 + Zeichenkette 2)

Tabelle 9-54 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	Resultierende Zeichenkette nach Verknüpfung ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Die maximale Länge von IN1 oder IN2 ist 255, die maximale Länge von OUT ist 0 oder 255 (Datentyp String)	
	Die maximale Länge von IN1 oder IN2 ist 65534, die maximale Länge von OUT ist 0 oder 65534 (Datentyp WString)	

9.2.4.4 Anweisungen LEFT, RIGHT und MID (Teilzeichenketten in einer Zeichenkette lesen)

Tabelle 9-55 Anweisungen LEFT, RIGHT und MID

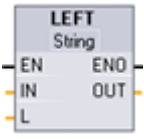
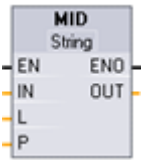
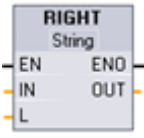
KOP/FUP	SCL	Beschreibung
	<code>out := LEFT(in, L);</code>	<p>LEFT (Linke Teilzeichenkette) gibt eine Teilzeichenkette aus, die aus den ersten L Zeichen von Parameter IN besteht.</p> <ul style="list-style-type: none"> Ist L größer als die aktuelle Länge der Zeichenkette IN, so wird die gesamte Zeichenkette IN an Ausgang OUT ausgegeben. Wird eine leere Zeichenkette eingegeben, so wird auch eine leere Zeichenkette an OUT ausgegeben.
	<code>out := MID(in, L, p);</code>	<p>MID (Mittlere Teilzeichenkette) gibt den mittleren Teil einer Zeichenkette aus. Die mittlere Teilzeichenkette ist L Zeichen lang und beginnt an Zeichenposition P (einschließlich).</p> <p>Wenn die Summe aus L und P die aktuelle Länge des String-Parameters IN überschreitet, wird eine Teilzeichenkette von Zeichenposition P bis zum Ende von Zeichenkette IN ausgegeben.</p>
	<code>out := RIGHT(in, L);</code>	<p>RIGHT (Rechte Teilzeichenkette) gibt die letzten L Zeichen einer Zeichenkette aus.</p> <ul style="list-style-type: none"> Ist L größer als die aktuelle Länge der Zeichenkette IN, so wird die gesamte Zeichenkette IN an Ausgang OUT ausgegeben. Wird eine leere Zeichenkette eingegeben, so wird auch eine leere Zeichenkette an OUT ausgegeben.

Tabelle 9-56 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
IN	IN	String, WString	Eingegebene Zeichenkette
L	IN	Int	<p>Länge der zu erstellenden Teilzeichenkette:</p> <ul style="list-style-type: none"> LEFT verwendet die Anzahl der Zeichen ganz links in der Zeichenkette RIGHT verwendet die Anzahl der Zeichen ganz rechts in der Zeichenkette MID verwendet die Anzahl der Zeichen ab Position P in der Zeichenkette
P	IN	Int	<p>Nur MID: Position des ersten Zeichens in der Teilzeichenkette, das kopiert werden soll</p> <p>P = 1, für die Position des ersten Zeichens der Zeichenkette IN</p>
OUT	OUT	String, WString	Ausgegebene Zeichenkette

Tabelle 9-57 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	<ul style="list-style-type: none"> L oder P ist kleiner oder gleich 0 P ist größer als die maximale Länge von IN Aktuelle Länge von IN überschreitet maximale Länge von IN, oder aktuelle Länge von OUT überschreitet maximale Länge von OUT Maximale Länge von IN oder OUT passt nicht in den zugewiesenen Speicher Die maximale Länge von IN oder OUT ist 0 oder 255 (Datentyp String) oder 0 oder 65534 (Datentyp WString) 	Aktuelle Länge wird auf 0 gesetzt
	Länge (L) der Teilzeichenkette, die kopiert werden soll, ist größer als die maximale Länge von Zeichenkette OUT	Es werden so viele Zeichen kopiert, bis die maximale Länge von OUT erreicht ist
	Nur MID: L oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Nur MID: P ist größer als die maximale Länge von IN	
	Aktuelle Länge von IN1 überschreitet maximale Länge von IN1 oder aktuelle Länge von IN2 überschreitet maximale Länge von IN2 (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
Die maximale Länge von IN1, IN2 oder OUT ist unzulässig: 0 oder 255 (Datentyp String) oder 0 oder 65534 (Datentyp WString)		

9.2.4.5 DELETE (Zeichen in einer Zeichenkette löschen)

Tabelle 9-58 Anweisung Teilzeichenkette löschen

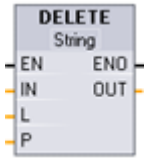
KOP/FUP	SCL	Beschreibung
	<pre>out := DELETE(in, L, p);</pre>	<p>Löscht L Zeichen aus der Zeichenkette IN. Mit dem Löschen der Zeichen wird an Zeichenposition P (einschließlich) begonnen, und die restliche Teilzeichenkette wird an Parameter OUT ausgegeben.</p> <ul style="list-style-type: none"> Ist L gleich Null, so wird die eingegebene Zeichenkette an OUT ausgegeben. Ist die Summe aus L und P größer als die Länge der eingegebenen Zeichenkette, so wird die Zeichenkette bis zum Ende gelöscht.

Tabelle 9-59 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
IN	IN	String, WString	Eingegebene Zeichenkette
L	IN	Int	Anzahl der zu löschenden Zeichen
P	IN	Int	Position des ersten Zeichens, das gelöscht werden soll: Das erste Zeichen der Zeichenkette IN hat Position 1.
OUT	OUT	String, WString	Ausgegebene Zeichenkette

Tabelle 9-60 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die aktuelle Länge von IN	IN wird in OUT kopiert, es werden keine Zeichen gelöscht
	Resultierende Zeichenkette nach Löschen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	L ist kleiner als 0 oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Aktuelle Länge von IN überschreitet maximale Länge von IN, oder aktuelle Länge von OUT überschreitet maximale Länge von OUT	
	Maximale Länge von IN oder OUT passt nicht in den zugewiesenen Speicher	
Maximale Länge von IN oder OUT ist 0 oder 255		

9.2.4.6 INSERT (Zeichen in einer Zeichenkette einfügen)

Tabelle 9-61 Anweisung Teilzeichenkette einfügen

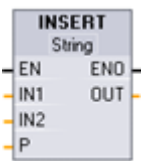
KOP/FUP	SCL	Beschreibung
	<code>out := INSERT(in1, in2, p);</code>	Fügt die Zeichenkette IN2 in die Zeichenkette IN1 ein. Das Einfügen beginnt nach dem Zeichen an Position P.

Tabelle 9-62 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	String, WString Eingegebene Zeichenkette 1
IN2	IN	String, WString Eingegebene Zeichenkette 2
P	IN	Int Letzte Zeichenposition in Zeichenkette IN1 vor dem Einfügungspunkt für Zeichenkette IN2. Das erste Zeichen der Zeichenkette IN1 hat Position 1.
OUT	OUT	String, WString Resultierende Zeichenkette

Tabelle 9-63 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die Länge von IN1	IN2 wird unmittelbar nach dem letzten Zeichen von IN1 mit IN1 verkettet
	P ist kleiner als 0	Aktuelle Länge wird auf 0 gesetzt
	Resultierende Zeichenkette nach Einfügen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT (ungültige Zeichenkette)	Aktuelle Länge wird auf 0 gesetzt
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Die maximale Länge von IN1 oder IN2 ist 255 oder maximale Länge von OUT ist 0 oder 255 (Datentyp String)	
	Die maximale Länge von IN1 oder IN2 ist 65534 oder maximale Länge von OUT ist 0 oder 65534 (Datentyp WString)	

9.2.4.7 REPLACE (Zeichen in einer Zeichenkette ersetzen)

Tabelle 9-64 Anweisung Teilzeichenkette ersetzen

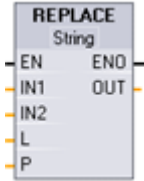
KOP/FUP	SCL	Beschreibung
	<pre> out := REPLACE(in1:=_string_in_, in2:=_string_in_, L:=_int_in_, p:=_int_in_); </pre>	Ersetzt L Zeichen in der Zeichenkette im Parameter IN1. Das Ersetzen beginnt an Zeichenposition P (einschließlich) der Zeichenkette IN1, die Ersetzungszeichen werden aus Parameter IN2 übernommen.

Tabelle 9-65 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
IN1	IN	String, WString	Eingegebene Zeichenkette
IN2	IN	String, WString	Zeichenkette mit Ersatzzeichen
L	IN	Int	Anzahl der Zeichen, die ersetzt werden sollen
P	IN	Int	Position des ersten Zeichens, das ersetzt werden soll
OUT	OUT	String, WString	Resultierende Zeichenkette

Ist Parameter L gleich Null, so wird Zeichenkette IN2 an Position P von Zeichenkette IN1 eingefügt, ohne dass Zeichen in Zeichenkette IN1 gelöscht werden.

Ist P gleich 1, werden die ersten L Zeichen von Zeichenkette IN1 durch die Zeichen der Zeichenkette IN2 ersetzt.

Tabelle 9-66 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichen
0	P ist größer als die Länge von IN1	IN2 wird unmittelbar nach dem letzten Zeichen von IN1 mit IN1 verkettet
	P Punkte in IN1, aber weniger als L Zeichen in IN1	IN2 ersetzt das Endzeichen von IN1 ab Position P
	Resultierende Zeichenkette nach Ersetzen ist größer als die maximale Länge von Zeichenkette OUT	Die resultierenden Zeichen werden kopiert, bis die maximale Länge von OUT erreicht ist
	Maximale Länge von IN1 ist 0	IN2-Zeichen werden in OUT kopiert
	L ist kleiner als 0 oder P ist kleiner oder gleich 0	Aktuelle Länge wird auf 0 gesetzt
	Die aktuelle Länge von IN1 überschreitet die maximale Länge von IN1, die aktuelle Länge von IN2 überschreitet die maximale Länge von IN2 oder die aktuelle Länge von OUT überschreitet die maximale Länge von OUT	
	Maximale Länge von IN1, IN2 oder OUT passt nicht in den zugewiesenen Speicherbereich	
	Die maximale Länge von IN1 oder IN2 ist 255 oder maximale Länge von OUT ist 0 oder 255 (Datentyp String)	
	Die maximale Länge von IN1 oder IN2 ist 65534 oder maximale Länge von OUT ist 0 oder 65534 (Datentyp WString)	

9.2.4.8 FIND (Zeichen in einer Zeichenkette finden)

Tabelle 9-67 Anweisung Teilzeichenkette finden

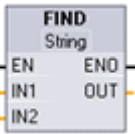
KOP/FUP	SCL	Beschreibung
	<pre>out := FIND(in1:=_string_in_, in2:=_string_in);</pre>	Liefert die Zeichenposition der von IN2 angegebenen Zeichenkette in der Zeichenkette IN1. Die Suche beginnt links. Die Zeichenposition des ersten Vorkommens von IN2 wird an OUT ausgegeben. Wird die Zeichenkette IN2 nicht in der Zeichenkette IN1 gefunden, so wird Null ausgegeben.

Tabelle 9-68 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
IN1	IN	String, WString
IN2	IN	String, WString
OUT	OUT	Int

Tabelle 9-69 ENO-Zustand

ENO	Bedingung	OUT
1	Keine Fehler	Gültige Zeichenposition
0	IN2 ist größer als IN1 Aktuelle Länge von IN1 überschreitet maximale Länge von IN1 oder aktuelle Länge von IN2 überschreitet maximale Länge von IN2 (ungültige Zeichenkette) Maximale Länge von IN1 oder IN2 passt nicht in den zugewiesenen Speicherbereich Die maximale Länge von IN1 oder IN2 ist 255 (Datentyp String) oder 65535 (Datentyp WString)	Zeichenposition wird auf 0 gesetzt

9.2.5 Informationen zur Laufzeit

9.2.5.1 GetSymbolName (Namen einer Variable am Eingangsparameter auslesen)

Tabelle 9-70 Anweisung GetSymbolName

KOP/FUP	SCL	Beschreibung
<p style="text-align: center;">GetSymbolName</p> <p> - EN ENO - - variable OUT - - size </p>	<pre> OUT := GetSymbolName (variable:=_parameter_in_, size:=_dint_in_); </pre>	<p>Die Anweisung GetSymbolName gibt eine Zeichenkette aus, die dem Namen einer Variablen aus der Bausteinschnittstelle entspricht.</p> <p>Ihr Programm kann die Anweisung mehrere Male mit unterschiedlichen Variablen aufrufen. Der Prozesswert der Variable ist irrelevant.</p> <p>Die Anweisung gibt den gelesenen Namen am Parameter OUT aus.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetSymbolName:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
VARIABLE	Input	PARAMETER	Parameterabschnitte Input, Output, InOut	Variable aus der lokalen Bausteinschnittstelle, deren Name als String-Wert ausgegeben werden soll
SIZE	Input	DINT	E, A, M, D, L	Begrenzt die Anzahl von Zeichen, die am Parameter OUT ausgegeben werden: <ul style="list-style-type: none"> • SIZE > 0: GetSymbolName gibt die ersten SIZE Zeichen des Namens aus. • SIZE = 0: GetSymbolName gibt den vollständigen Namen aus. • SIZE < 0: GetSymbolName gibt die letzten SIZE Zeichen des Namens aus.
OUT	Return	WSTRING	E, A, M, D, L	Ausgabe des vom Eingangsparameter gelieferten Variablennamens

Sie geben die Eingangsparameter der Bausteinschnittstelle am Parameter VARIABLE an. Verwenden Sie nur einen Schnittstellenparameter für diesen Parameter und keine PLC-Variable oder Datenbausteinvariable.

Um die Länge des gelesenen Variablennamens zu begrenzen, verwenden Sie den Parameter SIZE. Wenn die Anweisung den Namen abschneidet, wird dies durch die Zeichen "..." (Unicode-Zeichen 16#2026) am Ende des Namens deutlich gemacht. Beachten Sie, dass dieses Zeichen die Länge 1 hat.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Beispiel: Bedeutung des Parameters SIZE

Das folgende Beispiel veranschaulicht die Bedeutung des Parameters SIZE. Der folgende Variablenname wird aus der Bausteinschnittstelle gelesen: "MyPLCTag" (die doppelten Anführungszeichen am Anfang und Ende des Namens gehören zum Namen).

SIZE	Ausgabe von GetSymbol-Name	Bedeutung
1	'...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Kennzeichen dafür, dass der Name abgeschnitten wurde: ... • Letztes Zeichen von WSTRING:'
2	""...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Das erste Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... • Letztes Zeichen von WSTRING:'

SIZE	Ausgabe von GetSymbol-Name	Bedeutung
3	"M...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten beiden Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... "M... • Letztes Zeichen von WSTRING:'
6	"MyPL...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten fünf Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde: "MyPL... • Letztes Zeichen von WSTRING:'
0	"MyPLCTag"	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Alle Zeichen des Namens: "MyPLCTag" • Letztes Zeichen von WSTRING:'

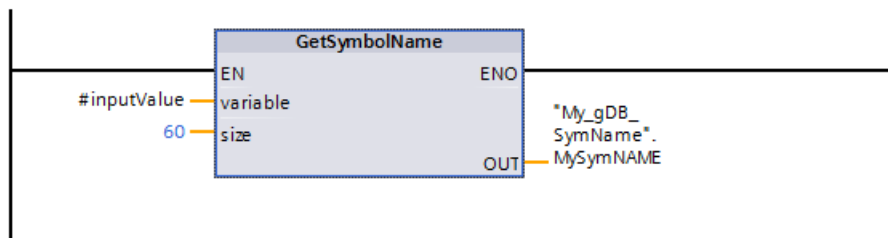
Beispiel: Lesen eines symbolischen Namens

Im folgenden Beispiel lesen Sie den Namen einer Variablen, die über den Eingangsparameter eines Bausteins verschaltet ist.

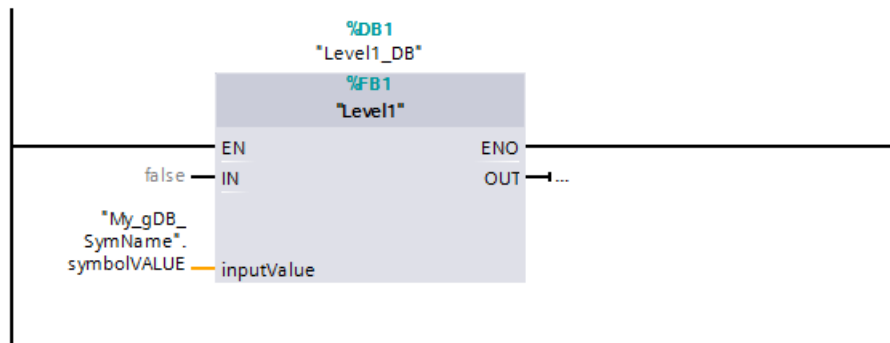
Legen Sie zwei Variablen in einem globalen Datenbaustein zum Speichern der Daten an.

My_gDB_SymName			
	Name	Datentyp	Startwert
1	Static		
2	MySymNAME	Wstring	WSTRING#"
3	symbolVALUE	Byte	16#42

Legen Sie einen Eingangsparameter inputValue mit dem Datentyp BYTE im Baustein Level1 an. Rufen Sie die Anweisung GetSymbolName im Baustein Level1 auf. Verschalten Sie die Parameter der Anweisung wie folgt:



Verschalten Sie den Parameter inputValue des Bausteins Level1 wie folgt:



Die Anweisung GetSymbolName wird im Baustein Level1 ausgeführt. Der Eingangsparameter inputValue des Bausteins Level1 wird mit Hilfe des Eingangsparameters VARIABLE der Anweisung auf seine Verschaltung hin untersucht. Dabei wird die Variable symbolVALUE ausgelesen und am Ausgangparameter OUT als Zeichenkette ausgegeben ("MySymNAME"). Entsprechend dem Wert des Eingangsparameters SIZE wird die Länge der Zeichenkette auf 60 Zeichen begrenzt.

My_gDB_SymName				
	Name	Datentyp	Startwert	Beobachtungswert
1	Static			
2	MySymNAME	WString	WSTRING#"	WSTRING#"My_gDB_SymName".symbolVALUE"
3	symbolVALUE	Byte	16#42	16#42

9.2.5.2 GetSymbolPath (Zusammengesetzten globalen Namen der Eingangsparameterversorgung abfragen)

Tabelle 9-71 Anweisung GetSymbolPath

KOP/FUP	SCL	Beschreibung
<p>GetSymbolPath</p> <p>EN ENO</p> <p>variable OUT</p> <p>size</p>	<pre>OUT := GetSymbolPath(variable:=_parameter_in_, size:=_dint_in_);</pre>	<p>Die Anweisung GetSymbolPath liest den zusammengesetzten globalen Namen eines Eingangsparameters an der lokalen Schnittstelle eines Bausteins (FB oder FC). Der Name besteht aus dem Speicherpfad und dem Variablennamen.</p> <p>Ihr Programm kann die Anweisung mehrere Male mit unterschiedlichen Variablen aufrufen. Der Prozesswert der Variable ist irrelevant.</p> <p>Die Anweisung gibt den gelesenen Namen am Parameter OUT aus.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetSymbolPath:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
VARIABLE	Input	PARAMETER	Parameterabschnitte Input, Output, InOut	Auswahl der lokalen Schnittstelle, für die Sie den globalen Namen der Eingangsparameterversorgung lesen möchten.
SIZE	Input	DINT	E, A, M, D, L oder Konstante	Begrenzt die Anzahl von Zeichen, die am Parameter OUT ausgegeben werden. <ul style="list-style-type: none"> • SIZE > 0: GetSymbolPath gibt die ersten SIZE Zeichen des Namens aus. • SIZE = 0: GetSymbolPath gibt den vollständigen Namen aus. • SIZE < 0: GetSymbolPath gibt die letzten SIZE Zeichen des Namens aus.
OUT	Output	WSTRING	E, A, M, D, L	Ausgabe des Variablennamens der Eingangsparameterversorgung.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Verwendung

Beachten Sie die folgenden Tipps zur Verwendung der Anweisung GetSymbolPath:

- Geben Sie die Bausteinschnittstelle, über die der Name der Eingangsvariablen gelesen wird, am Parameter VARIABLE der Anweisung an:
 - Wenn eine Datenbausteinvariable den Eingangsparameter liefert, gibt GetSymbolPath den Namen des DB, darin enthaltene Strukturen und den Namen der Variablen aus.
 - Wenn eine PLC-Variable den Eingangsparameter liefert, gibt GetSymbolPath den Namen der PLC-Variablen aus.
 - Wenn eine Konstante den Eingangsparameter liefert, gibt GetSymbolPath den Wert der Konstanten aus.
- Um die Länge des gelesenen Variablennamens zu begrenzen, verwenden Sie den Parameter SIZE. Wenn der Name abgeschnitten wurde, wird dies durch die Zeichen "..." (Unicode-Zeichen 16#2026) am Ende des Namens deutlich gemacht. Beachten Sie, dass dieses Zeichen die Länge 1 hat.

Beispiel: Bedeutung des Parameters SIZE

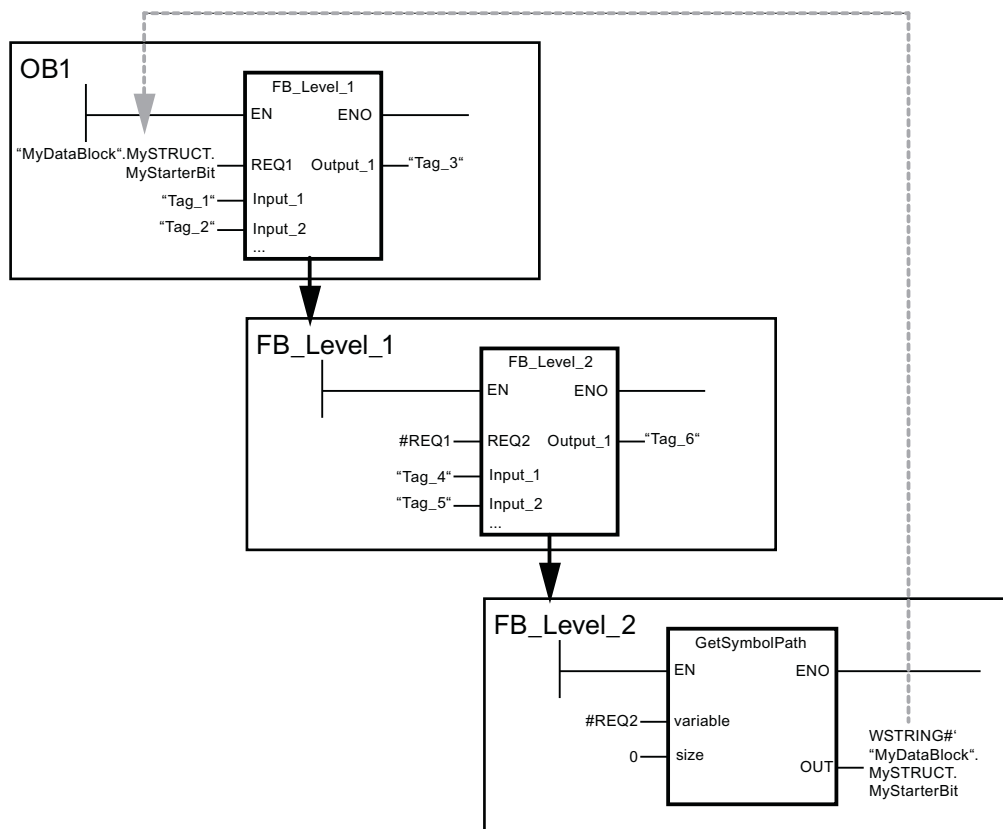
Das folgende Beispiel veranschaulicht die Bedeutung des Parameters SIZE. GetSymbolPath hat den folgenden Variablennamen aus der Bausteinschnittstelle gelesen: "MyPLCTag" (die doppelten Anführungszeichen am Anfang und Ende des Namens gehören zum Namen).

SIZE	Ausgabe von GetSymbol-Path	Bedeutung
1	'...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Kennzeichen dafür, dass der Name abgeschnitten wurde: ... • Letztes Zeichen von WSTRING:'
2	"...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Das erste Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... • Letztes Zeichen von WSTRING:'
3	"M...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten beiden Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... "M... • Letztes Zeichen von WSTRING:'
6	"MyPL...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten fünf Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde: "MyPL... • Letztes Zeichen von WSTRING:'
0	"MyPLCTag"	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Alle Zeichen des Namens: "MyPLCTag" • Letztes Zeichen von WSTRING:'

Beispiel: Aufrufen von GetSymbolPath über mehrere Stufen des Bausteinaufrufs

Das folgende Beispiel zeigt die Verwendung von GetSymbolPath über mehrere Aufrufstufen:

- Organisationsbaustein OB1 ruft den Baustein FB_Level_1 auf, der wiederum den Baustein FB_Level_2 aufruft.
- Der Baustein FB_Level_2 führt GetSymbolPath aus, um den Pfad des Parameters an der Schnittstelle REQ2 zu lesen.
- Da die Schnittstelle REQ1 den Wert REQ2 liefert, ermittelt die Anweisung den Pfad des Eingangsparameters von REQ1.
- Die Variable MyStarterBit ist der Eingangsparameter REQ1. Das Bit befindet sich in der Struktur MySTRUCT im Datenbaustein MyDataBlock. GetSymbolPath liest diese Informationen und gibt den Pfad ("MyDataBlock".MySTRUCT.MyStarterBit) am Parameter OUT aus.



9.2.5.3 GetInstanceName (Namen der Baustein-Instanz auslesen)

Tabelle 9-72 Anweisung GetInstanceName

KOP/FUP	SCL	Beschreibung
<p>GetInstanceName</p> <p>- EN ENO -</p> <p>- size OUT -</p>	<pre>OUT := GetInstanceName(size:=_dint_in_);</pre>	<p>Mit der Anweisung GetInstanceName können Sie den Namen des Instanz-Datenbausteins in einem Funktionsbaustein lesen.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetInstanceName:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SIZE	Input	DINT	E, A, M, D, L oder Konstante	Begrenzt die Anzahl von Zeichen, die am Parameter OUT ausgegeben werden. <ul style="list-style-type: none"> • SIZE > 0: GetInstanceName gibt die ersten SIZE Zeichen des Namens aus. • SIZE = 0: GetInstanceName gibt den vollständigen Namen aus. • SIZE < 0: GetInstanceName gibt die letzten SIZE Zeichen des Namens aus.
OUT	Output	WSTRING	E, A, M, D, L	Gelesener Name des Instanz-Datenbausteins

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Beispiel: Bedeutung des Parameters SIZE

Um die Länge des gelesenen Instanznamens zu begrenzen, verwenden Sie den Parameter SIZE. Wenn die Anweisung den Namen abgeschnitten hat, wird dies durch die Zeichen "..." (Unicode-Zeichen 16#2026) am Ende des Namens deutlich gemacht. Beachten Sie, dass dieses Zeichen die Länge 1 hat.

Das folgende Beispiel veranschaulicht die Bedeutung des Parameters SIZE. GetInstanceName hat den folgenden Instanznamen aus der Bausteinschnittstelle gelesen: "Level1_DB" (die doppelten Anführungszeichen am Anfang und Ende des Namens gehören zum Namen).

SIZE	Ausgabe von GetSymbol-Path	Bedeutung
1	'...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Kennzeichen dafür, dass der Name abgeschnitten wurde: ... • Letztes Zeichen von WSTRING:'
2	""...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Das erste Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"...' • Letztes Zeichen von WSTRING:'
3	""L...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten beiden Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... "L...' • Letztes Zeichen von WSTRING:'

SIZE	Ausgabe von GetSymbol-Path	Bedeutung
6	"Leve...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten fünf Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde: "Leve... • Letztes Zeichen von WSTRING:'
0	"Level1_DB"	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Alle Zeichen des Namens: "Level1_DB" • Letztes Zeichen von WSTRING:'

GetInstanceName schreibt den Namen des Instanz-Datenbausteins in den Parameter OUT. Die Anweisung schneidet den Namen ab, wenn der Name des Instanz-Datenbausteins länger ist als die maximale Länge von WSTRING.

Beispiel: Lesen des Namens eines Instanz-Datenbausteins

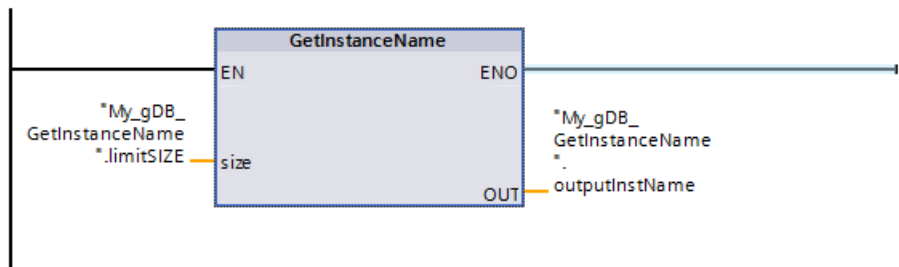
Das folgende Beispiel zeigt, wie Sie den Namen eines Instanz-Datenbausteins lesen.

Legen Sie zwei Variablen in einem globalen Datenbaustein zum Speichern der Daten an.

Definieren Sie die Parameter der Anweisung wie folgt:

My_gDB_GetInstanceName			
	Name	Datentyp	Startwert
1	Static		
2	limitSIZE	DInt	0
3	outputInstName	WString	WSTRING#''

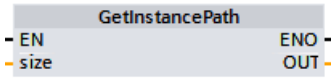
Der Baustein Level1_gin führt die Anweisung GetInstanceName aus, wodurch der zugehörige Instanz-Datenbaustein des Bausteins Level1_gin ermittelt und der Name am Ausgangsparameter OUT als Zeichenkette ausgegeben wird (outputInstName). Entsprechend dem Wert 0 von Parameter SIZE (limitSIZE) ist die Länge der Zeichenkette unbegrenzt.



My_gDB_GetInstanceName				
	Name	Datentyp	Startwert	Beobachtungswert
1	Static			
2	limitSIZE	DInt	0	0
3	outputInstName	WString	WSTRING#''	WSTRING#"Level_1_DB"

9.2.5.4 GetInstancePath (Zusammengesetzten globalen Namen der Baustein-Instanz abfragen)

Tabelle 9-73 Anweisung GetInstancePath

KOP/FUP	SCL	Beschreibung
	<pre>OUT := GetInstancePath(size:=_dint_in_);</pre>	<p>Mit der Anweisung GetInstancePath lesen Sie den zusammengesetzten globalen Namen der Bausteininstanz in einem Funktionsbaustein. Der zusammengesetzte globale Name der Bausteininstanz ist der Pfad der vollständigen Aufrufhierarchie, wenn das Programm mehrere Instanzen aufruft.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetInstancePath:

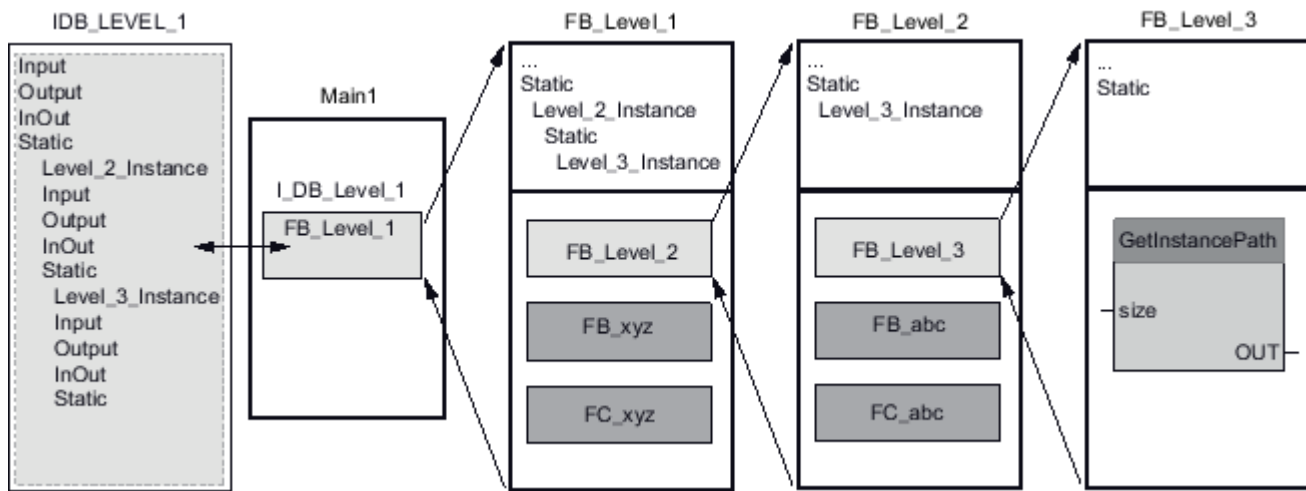
Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SIZE	Input	DINT	E, A, M, D, L oder Konstante	<p>Begrenzt die Anzahl von Zeichen, die am Parameter OUT ausgegeben werden.</p> <ul style="list-style-type: none"> • SIZE > 0: GetInstancePath gibt die ersten SIZE Zeichen des Namens aus. • SIZE = 0: GetInstancePath gibt den vollständigen Namen aus. • SIZE < 0: GetInstancePath gibt die letzten SIZE Zeichen des Namens aus.
OUT	Output	WSTRING	E, A, M, D, L	<p>Ausgelesener globaler Namen der Baustein-Instanz.</p> <p>Wenn der globale Name der Bausteininstanz länger als die maximale Länge von WSTRING (254 Zeichen) ist, schneidet GetInstancePath den Namen ab.</p>

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Beispiel: Aufrufen von GetInstancePath, um den Pfad des Aufrufs eines FB mit mehreren Instanzen abzufragen

Im folgenden Beispiel ruft der Funktionsbaustein FB_Level_3 die Anweisung GetInstancePath auf.

- Der Funktionsbaustein FB_Level_3 speichert seine Daten in dem aufrufenden Funktionsbaustein FB_Level_2.
- Der Funktionsbaustein FB_Level_2 wiederum speichert seine Daten in dem aufrufenden Funktionsbaustein FB_Level_1.
- Der Funktionsbaustein FB_Level_1 wiederum speichert seine Daten in seinem Instanz-Datenbaustein IDB_LEVEL_1. Durch die Verwendung mehrerer Instanzen enthält der Instanz-Datenbaustein von FB_Level_1 alle Daten der drei Funktionsbausteine.



Die Anweisung GetInstancePath gibt abhängig vom Wert des Parameters SIZE die folgenden Werte für dieses Beispiel aus:

SIZE	Ausgabe von GetInstance-Path	Bedeutung
1	'...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Kennzeichen dafür, dass der Name abgeschnitten wurde: ... • Letztes Zeichen von WSTRING:'
2	"'...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Das erste Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"'... • Letztes Zeichen von WSTRING:'
3	"'l...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten beiden Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"'l... • Letztes Zeichen von WSTRING:'
6	"'IDB_...'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Die ersten fünf Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"'IDB_... • Letztes Zeichen von WSTRING:'
0	"'IDB_LEVEL_1".Level_2_Instance.Level_3_Instance'	<ul style="list-style-type: none"> • Erstes Zeichen von WSTRING:' • Alle Zeichen des Namens: "IDB_LEVEL_1".Level_2_Instance.Level_3_Instance • Letztes Zeichen von WSTRING:'

Hinweis**Verwendung von GetInstancePath in Funktionsbausteinen mit einer Instanz**

Wenn der Funktionsbaustein, in dem Sie GetInstancePath aufrufen, Daten in seinem eigenen Instanz-Datenbaustein speichert, gibt GetInstancePath den Namen der einzelnen Instanz als den globalen Namen aus. Das Ergebnis am Parameter OUT entspricht in diesem Fall der Anweisung GetInstanceName (Seite 364).

9.2.5.5 GetBlockName (Name des Bausteins auslesen)

Tabelle 9-74 Anweisung GetBlockName

KOP/FUP	SCL	Beschreibung
<div style="border: 1px solid black; padding: 5px;"> <div style="text-align: center; background-color: #f0f0f0;">GetBlockName</div> <div style="display: flex; justify-content: space-between;"> - EN ENO - </div> <div style="display: flex; justify-content: space-between;"> - SIZE RET_VAL - </div> </div>	<pre>RET_VAL := GetBlockName(size:=_dint_in_);</pre>	Mit der Anweisung GetBlockName lesen Sie den Name des Bausteins, in dem die Anweisung aufgerufen wird.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetBlockName:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
SIZE	Input	UINT	E, A, M, D, L oder Konstante	Begrenzt die Anzahl von Zeichen, die am Parameter RET_VAL ausgegeben werden. <ul style="list-style-type: none"> SIZE > 0: GetBlockName gibt die ersten SIZE Zeichen des Namens aus. SIZE = 0: GetBlockName gibt den vollständigen Namen aus. SIZE < 0: GetBlockName gibt die letzten SIZE Zeichen des Namens aus.
RET_VAL	Output	WSTRING	E, A, M, D, L	Gelesener Name des Instanz-Datenbausteins

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Beispiel: Bedeutung des Parameters SIZE

Um die Länge des Bausteinennamens auf eine bestimmte Anzahl von Zeichen zu begrenzen, geben Sie die maximale Länge am Parameter SIZE an. Wenn GetBlockName den Namen abschneidet, wird dies durch das Zeichen "..." (Unicode-Zeichen 16#2026) am Ende des Namens deutlich gemacht. Beachten Sie, dass dieses Zeichen die Länge 1 hat.

Das folgende Beispiel veranschaulicht die Bedeutung des Parameters SIZE. GetBlockName hat den folgenden Bausteinnamen ausgelesen: Level1_gbn (die doppelten Anführungszeichen am Anfang und Ende des Namens gehören zum Namen).

SIZE	Ausgabe von GetBlockName	Bedeutung
1	'...'	<ul style="list-style-type: none"> Erstes Zeichen von WSTRING:' Kennzeichen dafür, dass der Name abgeschnitten wurde: ... Letztes Zeichen von WSTRING:'
2	""...'	<ul style="list-style-type: none"> Erstes Zeichen von WSTRING:' Das erste Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"...' Letztes Zeichen von WSTRING:'
3	""L...'	<ul style="list-style-type: none"> Erstes Zeichen von WSTRING:' Die ersten beiden Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde:"... "L...' Letztes Zeichen von WSTRING:'
6	""Leve...'	<ul style="list-style-type: none"> Erstes Zeichen von WSTRING:' Die ersten fünf Zeichen des Namens und Kennzeichen dafür, dass der Name abgeschnitten wurde: "Leve...' Letztes Zeichen von WSTRING:'
0	""Level1_gbn""	<ul style="list-style-type: none"> Erstes Zeichen von WSTRING:' Alle Zeichen des Namens: "Level1_gbn" Letztes Zeichen von WSTRING:'

GetBlockName schreibt den Namen des Bausteins in den Parameter RET_VAL. Wenn der Name des Bausteins länger ist als die maximale Länge von WSTRING, wird der Name abgeschnitten.

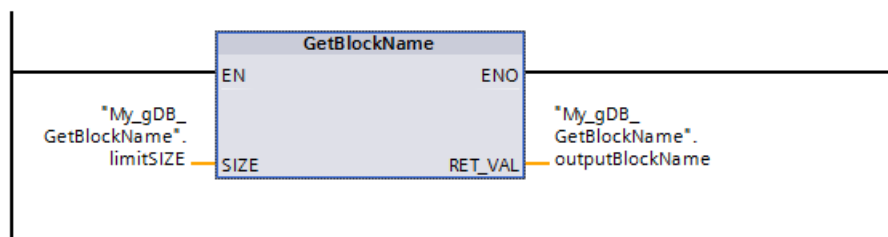
Beispiel: Lesen eines Bausteinnamens

Das folgende Beispiel zeigt, wie Sie einen Bausteinnamen auslesen.

- Legen Sie zwei Variablen in einem globalen Datenbaustein zum Speichern der Daten an.

My_gDB_GetBlockName			
	Name	Datentyp	Startwert
1	Static		
2	limitSIZE	DInt	0
3	outputBlockName	WString	WSTRING#''

- Definieren Sie die Parameter der Anweisung wie folgt:



Der Baustein Level1_gbn führt die Anweisung GetBlockName aus. GetBlockName liest den Namen des Bausteins Level1_gbn und gibt den Namen am Ausgangsparameter RET_VAL als Zeichenkette aus (outputBlockName). Da der Parameter SIZE gleich 0 ist (limitSIZE), ist die Länge der Zeichenkette unbegrenzt.

My_gDB_GetBlockName				
	Name	Datentyp	Startwert	Beobachtungswert
1	Static			
2	limitSIZE	DInt	0	0
3	outputBlockName	WString	WSTRING#"	WSTRING#"Level_1_gbn"

9.3 Dezentrale E/A (PROFINET, PROFIBUS oder AS-i)

9.3.1 Anweisungen für die dezentrale E/A

Die folgenden Anweisungen für die dezentrale Peripherie können mit PROFINET, PROFIBUS oder AS-i verwendet werden:

- Anweisung RDREC (Seite 373): Liest einen Datensatz mit der Nummer INDEX aus einem Modul oder Gerät.
- Anweisung WRREC (Seite 373): Überträgt einen Datensatz mit der Nummer INDEX in ein von ID angegebenes Modul oder Gerät.
- Anweisung GETIO (Seite 376): Liest konsistent alle Eingänge eines DP-Normslaves/PROFINET IO-Device.
- Anweisung SETIO (Seite 377): Überträgt konsistent Daten aus dem vom Parameter OUTPUTS definierten Quellbereich in den adressierten DP-Normslave bzw. das PROFINET IO-Device.
- Anweisung GETIO_PART (Seite 378): Liest konsistent einen zugehörigen Teil der Eingänge eines IO-Moduls.
- Anweisung SETIO_PART (Seite 379): Schreibt konsistent Daten aus dem vom Parameter OUTPUTS angegebenen Quellbereich in die Ausgänge eines IO-Moduls.
- Anweisung RALRM (Seite 380): Ermöglicht Ihnen, einen Alarm mit allen entsprechenden Informationen von einem Modul oder Gerät zu empfangen und diese Informationen an die Ausgangsparameter zu übergeben.
- Anweisung DPRD_DAT (Seite 393): Ermöglicht Ihnen, mit der Anweisung DPRD_DAT konsistente Datenbereiche größer als 64 Byte aus einem Modul oder Gerät zu lesen.
- Anweisung DPWR_DAT (Seite 393): Ermöglicht Ihnen, mit der Anweisung DPWR_DAT konsistente Datenbereiche größer als 64 Byte in ein Modul oder Gerät zu schreiben.

Die Anweisung D_ACT_DP (Seite 383) ermöglicht Ihnen, konfigurierte PROFINET IO-Devices zielgerichtet zu aktivieren bzw. zu deaktivieren. Außerdem können Sie ermitteln, ob ein zugewiesenes PROFINET IO-Device gegenwärtig aktiviert oder deaktiviert ist.

Hinweis

Hinweis: Die Anweisung D_ACT_DP können Sie nur bei PROFINET IO-Devices verwenden. Sie können die Anweisung nicht mit PROFIBUS DP-Slaves verwenden.

Mit der Anweisung DPNRM_DG (Seite 401) können Sie die aktuellen Diagnosedaten eines DP-Slaves in dem in EN 50 170 Band 2, PROFIBUS, spezifizierten Format lesen.

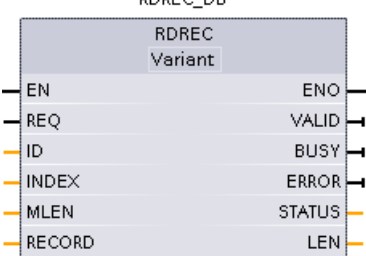
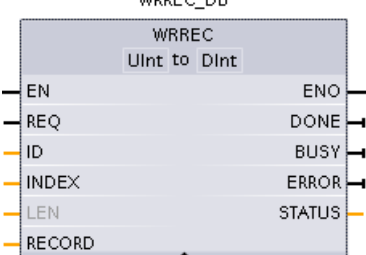
Hinweis

Die Anweisung DPNRM_DG können Sie nur für PROFIBUS einsetzen.

9.3.2 RDREC und WRREC (Datensatz lesen/schreiben)

Die Anweisungen RDREC (Datensatz lesen) und WRREC (Datensatz schreiben) können Sie für PROFINET, PROFIBUS und AS-i einsetzen.

Tabelle 9-75 Anweisungen RDREC und WRREC

KOP/FUP	SCL	Beschreibung
<p>"RDREC_DB"</p> 	<pre>"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_);</pre>	<p>Mit der Anweisung RDREC lesen Sie einen Datensatz mit der Nummer INDEX aus der Komponente, die von ID angegeben wird, z. B. aus einem zentralen Baugruppenträger oder einer dezentralen Komponente (PROFIBUS DP oder PROFINET IO). Geben Sie die maximale Anzahl der zu lesenden Bytes in MLEN an. Die ausgewählte Länge des Zielbereichs RECORD muss mindestens die Länge der von MLEN angegebenen Bytes haben.</p>
<p>"WRREC_DB"</p> 	<pre>"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_);</pre>	<p>Mit der Anweisung WRREC übertragen Sie einen Datensatz (RECORD) mit der Nummer INDEX an einen DP-Slave bzw. eine PROFINET IO-Device-Komponente, der/die von ID angegeben wird, z. B. ein Modul im zentralen Baugruppenträger oder eine dezentrale Komponente (PROFIBUS DP oder PROFINET IO).</p> <p>Geben Sie die Byte-Länge des zu übertragenden Datensatzes an. Die ausgewählte Länge des Quellbereichs RECORD muss deshalb mindestens die Länge der von LEN angegebenen Bytes haben.</p>

- ¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- ² In den SCL-Beispielen sind "RDREC_DB" und "WRREC_DB" die Namen der Instanz-DBs.

Tabelle 9-76 Datentypen für die Parameter von RDREC und WRREC

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	REQ = 1: Datensatz übertragen
ID	IN	HW_IO (Word)	<p>Logische Adresse der DP-Slave/PROFINET IO-Komponente (Modul oder Submodul):</p> <ul style="list-style-type: none"> Bei einem Ausgabemodul muss Bit 15 gesetzt werden (zum Beispiel für Adresse 5: ID:=DW#16#8005). Bei einem Kombimodul muss die kleinere der beiden Adressen angegeben werden. <p>Hinweis: In V3.0 kann die Geräte-ID wahlweise auf zwei Arten ermittelt werden:</p> <ul style="list-style-type: none"> Durch Auswahl der folgenden Optionen in der "Netzschritt": <ul style="list-style-type: none"> Gerät (graues Feld) "Eigenschaften" des Geräts "Hardwarekennung" <p>Hinweis: Es wird jedoch nicht für alle Geräte die Hardwarekennung angezeigt.</p> Durch Auswahl der folgenden Optionen in der "Projektnavigation": <ul style="list-style-type: none"> PLC-Variablen Standardvariablen-tabelle Register "Konstanten" des Systems <p>Die Hardwarekennungen aller konfigurierten Geräte werden angezeigt.</p> <p>Hinweis: In V4.0 wird die Geräte-ID (Hardwarekennung) für das Schnittstellenmodul in der Variablen-tabelle über den Parameter "Gerätename [HEAD]" unter den Systemkonstanten ermittelt.</p>
INDEX	IN	Byte, Word, USInt, UInt, SInt, Int, DInt	Datensatznummer
MLEN	IN	Byte, USInt, UInt	Maximale Länge in Bytes der abzurufenden Informationen aus dem Datensatz (RDREC)
VALID	OUT	Bool	Neuer Datensatz wurde empfangen und ist gültig (RDREC). Das Bit VALID ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
DONE	OUT	Bool	Datensatz wurde übertragen (WRREC). Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool	<ul style="list-style-type: none"> BUSY = 1: Der Lesevorgang (RDREC) oder Schreibvorgang (WRREC) ist noch nicht beendet. BUSY = 0: Die Übertragung des Datensatzes ist beendet.
ERROR	OUT	Bool	ERROR = 1: Ein Lesefehler (RDREC) bzw. Schreibfehler (WRREC) ist aufgetreten. Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	DWord	Bausteinstatus (Seite 388) oder Fehlerinformationen (Seite 535)

Parameter und Datentyp		Datentyp	Beschreibung
LEN	OUT (RDREC) IN (WRREC)	UInt	<ul style="list-style-type: none"> Länge der abgerufenen Informationen aus dem Datensatz (RDREC) Maximale Byte-Länge des zu übertragenden Datensatzes (WRREC)
RECORD	IN_OUT	Variant	<ul style="list-style-type: none"> Zielbereich für den abgerufenen Datensatz (RDREC) Datensatz (WRREC)

Die Anweisungen RDREC und WRREC funktionieren asynchron, d.h. die Verarbeitung deckt mehrere Anweisungsaufrufe ab. Sie starten den Auftrag durch Aufruf von RDREC oder WRREC mit REQ = 1.

Der Auftragsstatus wird über den Ausgangsparameter BUSY und die beiden zentralen Bytes des Ausgangsparameters STATUS angezeigt. Die Übertragung des Datensatzes ist beendet, wenn der Ausgangsparameter BUSY auf den Wert FALSE gesetzt wurde.

Ist der Ausgangsparameter VALID (RDREC) oder DONE (WRREC) gleich TRUE (nur einen Zyklus lang), weist dies darauf hin, dass der Datensatz erfolgreich in den Zielbereich RECORD (RDREC) oder zum Zielgerät (WRREC) übertragen wurde. Bei RDREC enthält der Ausgangsparameter LEN die Länge der abgerufenen Daten in Bytes.

Der Ausgangsparameter ERROR (nur einen Zyklus lang, wenn ERROR = TRUE) zeigt an, dass beim Übertragen des Datensatzes ein Fehler aufgetreten ist. In diesem Fall enthält der Ausgangsparameter STATUS (nur während des einen Zyklus, in dem ERROR = TRUE) die Fehlerinformationen.

Datensätze werden vom Hersteller des Hardwaregeräts definiert. Ausführliche Informationen zu einem Datensatz finden Sie in der Gerätedokumentation des jeweiligen Hardwareherstellers.

Sie können bis zu vier RDREC-Anweisungen und vier WRREC-Anweisungen gleichzeitig verwenden.

Hinweis

Wenn Sie einen DPV1-Slave mit einer GSD-Datei (GSD ab Version 3) konfigurieren und die DP-Schnittstelle des DP-Masters auf "S7-kompatibel" gesetzt ist, können Sie im Anwenderprogramm mit "RDREC" keine Datensätze aus den E/A-Modulen lesen und mit "WRREC" keine Datensätze in die E/A-Module schreiben. In einem solchen Fall adressiert der DP-Master den falschen Steckplatz (konfigurierter Steckplatz + 3).

Abhilfe: Richten Sie als Schnittstelle des DP-Masters "DPV1" ein.

Hinweis

Die Schnittstellen der Anweisungen "RDREC" und "WRREC" sind mit den FBs "RDREC" und "WRREC" identisch, die in der "PROFIBUS-Richtlinie für die PROFIBUS-Kommunikation und Proxy-Funktionsbausteine nach IEC 61131-3" definiert sind.

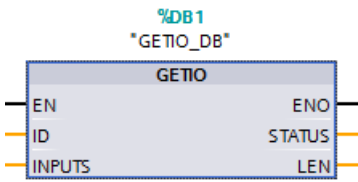
Hinweis

Wenn Sie "RDREC" oder "WRREC" verwenden, um einen Datensatz für PROFINET IO zu lesen oder zu schreiben, interpretiert die CPU negative Werte in den Parameter INDEX, MLEN und LEN als vorzeichenlose 16-Bit-Ganzzahlen.

9.3.3 GETIO (Prozessabbild lesen)

Mit der Anweisung "GETIO" lesen Sie konsistent Eingänge von Modulen oder Submodulen von DP-Slaves und PROFINET IO-Devices. Die Anweisung "GETIO" ruft die Anweisung "DPRD_DAT (Seite 393)" auf. Tritt während der Datenübertragung kein Fehler auf, werden die ausgelesenen Daten in den von INPUTS angegebenen Zielbereich eingegeben.

Tabelle 9-77 Anweisung GETIO (Prozessabbild lesen)

KOP/FUP	SCL	Beschreibung
	<pre>"GETIO_DB" (id:=_uint_in_, status=>_dword_out_, len=>_int_out_, inputs:=_variant_inout_);</pre>	<p>Mit der Anweisung "GETIO" lesen Sie konsistent alle Eingänge eines DP-Normslaves/PROFINET IO-Device.</p>

¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "GETIO_DB" der Name des Instanz-DB.

Der Zielbereich muss eine Länge größer oder gleich der Länge der ausgewählten Komponente haben.

Wenn Sie aus einem DP-Normslave mit modularem Aufbau oder mit mehreren DP-Kennungen lesen, greifen Sie pro Aufruf der Anweisung "GETIO" nur auf die Daten einer Komponente/DP-Kennung an der konfigurierten Startadresse zu.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "GETIO":

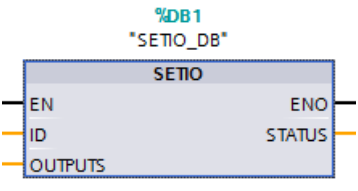
Parameter	Deklaration	Datentyp	Beschreibung
ID	IN	HW_SUBMODULE	Hardware-ID des DP-Normslaves/PROFINET IO-Device
STATUS ¹	OUT	DWord	Enthält die Fehlerinformationen von "DPRD_DAT (Seite 393)" im Format DW#16#40xxxx00
LEN	OUT	Int	Gelesene Datenmenge in Byte
INPUTS	IN_OUT	Variant	<p>Zielbereich für die gelesenen Daten: Der Zielbereich muss eine Länge größer oder gleich der Länge des ausgewählten DP-Normslaves/PROFINET IO-Device haben.</p> <p>Die folgenden Datentypen können Sie verwenden:</p> <ul style="list-style-type: none"> Systemdatentypen und Array von Systemdatentypen: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT Anwenderdefinierte Datentypen (UDT) Strukturen (STRUCT), jedoch nur in nicht optimierten Datenbausteinen (DB)

¹ Verwenden Sie bei der Anzeige von GETIO -Fehlercodes den Datentyp DWord.

9.3.4 SETIO (Prozessabbild übertragen)

Mit der Anweisung "SETIO" übertragen Sie konsistent Daten aus dem vom Parameter OUTPUTS definierten Quellbereich in die adressierten Module oder Submodule von DP-Slaves und PROFINET IO-Devices. Wenn Sie den relevanten Adressbereich des DP-Normslaves/PROFINET IO-Device als einen konsistenten Bereich in einem Prozessabbild konfiguriert haben, werden die Daten an das Prozessabbild übertragen. "SETIO" ruft während dieser Übertragung die Anweisung "DPWR_DAT (Seite 393)" auf.

Tabelle 9-78 Anweisung SETIO (Prozessabbild lesen)

KOP/FUP	SCL	Beschreibung
	<pre>"SETIO_DB" (id:=_uint_in_, status=>_dword_out_, outputs:=_variant_inout_);</pre>	<p>Mit der Anweisung "SETIO" übertragen Sie konsistent Daten aus dem vom Parameter OUTPUTS definierten Quellbereich in den adressierten DP-Normslave bzw. das PROFINET IO-Device.</p>

- STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "SETIO_DB" der Name des Instanz-DB.

Der Quellbereich muss eine Länge größer oder gleich der Länge der ausgewählten Komponente haben.

Bei einem DP-Normslave/PROFINET IO-Device mit modularem Aufbau oder mit mehreren DP-Kennungen, können Sie je SETIO-Aufruf auf nur eine DP-Kennung/Komponente zugreifen.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "SETIO":

Parameter	Deklaration	Datentyp	Beschreibung
ID	IN	HW_SUBMODULE	Hardware-ID des DP-Normslaves/PROFINET IO-Device
STATUS ¹	OUT	DWord	Enthält die Fehlerinformationen von "DPWR_DAT (Seite 393)" im Format DW#16#40xxxx00
OUTPUTS	IN_OUT	Variant	<p>Quellbereich für die zu schreibenden Daten: Der Quellbereich muss eine Länge größer oder gleich der Länge des ausgewählten DP-Normslaves/PROFINET IO-Device haben.</p> <p>Die folgenden Datentypen können Sie verwenden:</p> <ul style="list-style-type: none"> Systemdatentypen und Array von Systemdatentypen: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT Anwenderdefinierte Datentypen (UDT) Strukturen (STRUCT), jedoch nur in nicht optimierten Datenbausteinen (DB)

¹ Verwenden Sie bei der Anzeige von SETIO-Fehlercodes den Datentyp DWord.

9.3.5 GETIO_PART (Prozessabbildbereich lesen)

Mit der Anweisung "GETIO_PART" lesen Sie konsistent einen zugehörigen Teil der Eingänge von Modulen oder Submodulen von DP-Saves und PROFINET IO-Devices. GETIO_PART ruft die Anweisung "DPRD_DAT (Seite 393)" auf.

Tabelle 9-79 Anweisung GETIO_PART (Prozessabbildbereich lesen)

KOP/FUP	SCL	Beschreibung
	<pre>"GETIO_PART_DB" (id:=_uint_in_, offset:=_int_in_, len:=_int_in_, status=>_dword_out_, error=>_bool_out_, inputs:=_variant_inout_);</pre>	<p>Mit der Anweisung GETIO_PART lesen Sie konsistent einen zugehörigen Teil der Eingänge eines IO-Moduls.</p>

- ¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- ² Im SCL-Beispiel ist "GETIO_PART_DB" der Name des Instanz-DB.

Mit dem Eingangsparameter ID wählen Sie das IO-Modul über die Hardware-ID aus.

Mit den Parametern OFFSET und LEN geben Sie den Teil des Prozessabbaus an, der gelesen werden soll. Wenn der von OFFSET und LEN angegebene Eingangsbereich vom Modul nicht vollständig abgedeckt wird, gibt der Baustein den Fehlercode DW#16#4080B700 aus.

Die Länge des Zielbereichs muss größer oder gleich der zu lesenden Bytezahl sein:

- Tritt während der Datenübertragung kein Fehler auf, empfängt ERROR den Wert FALSE. Die gelesenen Daten werden in den vom Parameter INPUTS definierten Zielbereich geschrieben.
- Tritt während der Datenübertragung ein Fehler auf, empfängt ERROR den Wert TRUE. Der Parameter STATUS empfängt die Fehlerinformationen von DPRD_DAT.
- Wenn der Zielbereich größer als LEN ist, schreibt die Anweisung in die ersten LEN-Bytes des Zielbereichs. ERROR empfängt den Wert FALSE.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GETIO_PART:

Parameter	Deklaration	Datentyp	Beschreibung
ID	IN	HW_SUBMODULE	Hardwarekennung des Moduls
OFFSET	IN	Int	Nummer des ersten im Prozessabbild zu lesenden Bytes für die Komponente (kleinster möglicher Wert: 0)
LEN	IN	Int	Zu lesende Bytezahl
STATUS ¹	OUT	DWord	Enthält die Fehlerinformationen von "DPRD_DAT (Seite 393)" im Format DW#16#40xxx00, wenn ERROR = TRUE

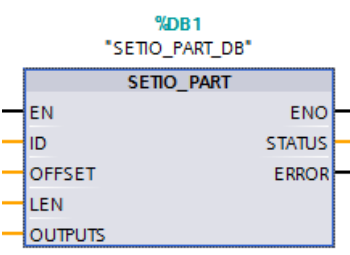
Parameter	Deklaration	Datentyp	Beschreibung
ERROR	OUT	Bool	Fehleranzeige: ERROR = TRUE, wenn bei Aufruf von DPRD_DAT (Seite 393) ein Fehler auftritt
INPUTS	IN_OUT	Variant	Zielbereich für die gelesenen Daten: Wenn der Zielbereich größer als LEN ist, schreibt die Anweisung in die ersten LEN-Bytes des Zielbereichs. Die folgenden Datentypen können Sie verwenden: <ul style="list-style-type: none"> Systemdatentypen und Array von Systemdatentypen: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT Anwenderdefinierte Datentypen (UDT) Strukturen (STRUCT), jedoch nur in nicht optimierten Datenbausteinen (DB)

¹ Verwenden Sie bei der Anzeige von GETIO_PART -Fehlercodes den Datentyp DWord.

9.3.6 SETIO_PART (Prozessabbildbereich übertragen)

Mit der Anweisung "SETIO_PART" schreiben Sie konsistent Daten aus dem von OUTPUTS angegebenen Quellbereich in die Ausgänge von Modulen und Submodulen von DP-Slaves und PROFINET IO-Devices. SETIO_PART ruft die Anweisung "DPWR_DAT (Seite 393)" auf.

Tabelle 9-80 Anweisung SETIO_PART (Prozessabbildbereich übertragen)

KOP/FUP	SCL	Beschreibung
	<pre>"SETIO_PART_DB" (id:=_uint_in_, offset:=_int_in_, len:=_int_in_, status=>_dword_out_, error=>_bool_out_, outputs:=_variant_inout_);</pre>	Mit der Anweisung SETIO_PART schreiben Sie konsistent Daten aus dem vom Parameter OUTPUTS angegebenen Quellbereich in die Ausgänge eines IO-Moduls.

¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "SETIO_PART_DB" der Name des Instanz-DBs.

Mit dem Eingangsparameter ID, wählen Sie das IO-Modul basierend auf der angegebenen Hardware aus.

Mit den Parametern OFFSET und LEN, weisen Sie den Teil des Prozessabbildbereichs zu, der für die von ID adressierte Komponente geschrieben werden soll. Wenn der von OFFSET und LEN angegebene Ausgangsbereich vom Modul nicht vollständig abgedeckt wird, gibt der Baustein den Fehlercode DW#16#4080B700 aus.

Die Länge des Zielbereichs muss größer oder gleich der zu lesenden Bytemenge sein:

- Tritt während der Datenübertragung kein Fehler auf, empfängt ERROR den Wert FALSE.
- Tritt während der Datenübertragung ein Fehler auf, empfängt ERROR den Wert TRUE und STATUS empfängt die Fehlerinformationen von DPWR_DAT.
- Wenn der Quellbereich größer als LEN ist, überträgt die Anweisung die ersten LEN Bytes von OUTPUTS. ERROR empfängt den Wert FALSE.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung SETIO_PART:

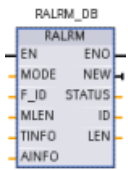
Parameter	Deklaration	Datentyp	Beschreibung
ID	IN	HW_SUBMODULE	Hardwarekennung des IO-Moduls
OFFSET	IN	Int	Nummer des ersten im Prozessabbild zu schreibenden Bytes für die Komponente (kleinster möglicher Wert: 0)
LEN	IN	Int	Zu schreibende Bytezahl
STATUS ¹	OUT	DWord	Enthält die Fehlerinformationen von "DPWR_DAT (Seite 393)" im Format DW#16#40xxx00, wenn ERROR = TRUE
ERROR	OUT	Bool	Fehleranzeige: ERROR = TRUE, wenn bei Aufruf von DPWR_DAT (Seite 393) ein Fehler auftritt
OUTPUTS	IN_OUT	Variant	<p>Quellbereich für die zu schreibenden Daten: Wenn der Quellbereich größer als LEN ist, werden die ersten LEN Bytes von OUTPUTS übertragen.</p> <p>Die folgenden Datentypen können Sie verwenden:</p> <ul style="list-style-type: none"> Systemdatentypen und Array von Systemdatentypen: BYTE, CHAR, SINT, USINT, WORD, INT, UINT, DWORD, DINT, UDINT, REAL, LREAL, LWORD, LINT Anwenderdefinierte Datentypen (UDT) Strukturen (STRUCT), jedoch nur in nicht optimierten Datenbausteinen (DB)

¹ Verwenden Sie bei der Anzeige von SETIO_PART -Fehlercodes den Datentyp DWord.

9.3.7 RALRM (Alarm empfangen)

Die Anweisung RALRM (Alarm lesen) können Sie für PROFINET und PROFIBUS einsetzen.

Tabelle 9-81 Anweisung RALRM

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung RALRM (Alarm lesen) lesen Sie Diagnosealarminformationen aus einem PROFIBUS- oder PROFINET I/O-Modul/Gerät.</p> <p>Die Informationen in den Ausgangsparametern enthalten die Startinformationen des aufgerufenen OBs sowie die Informationen der Alarmquelle.</p> <p>Rufen Sie RALRM in einem Alarm-OB auf, um zu den Ereignissen, die den Alarm verursacht haben, Informationen auszugeben. Bei der S7-1200 werden die folgenden Diagnosealarm-OBs unterstützt: Status, Aktualisierung, Profil, Diagnosefehleralarm, Ziehen oder Stecken von Modulen, Baugruppenträger- oder Stationsausfall</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "RALRM_DB" der Name des Instanz-DB.

Tabelle 9-82 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
MODE	IN	Byte, USInt, SInt, Int	Betriebszustand
F_ID	IN	HW_IO (Word)	<p>Logische Anfangsadresse der Komponente (des Moduls), von der Alarme empfangen werden sollen.</p> <p>Hinweis: Die Geräte-ID kann wahlweise auf zwei Arten ermittelt werden:</p> <ul style="list-style-type: none"> • Durch Auswahl der folgenden Optionen in der "Netzsicht": <ul style="list-style-type: none"> – Gerät (graues Feld) – "Eigenschaften" des Geräts – "Hardwarekennung" <p>Hinweis: Es wird nicht für alle Geräte die Hardwarekennung angezeigt.</p> • Durch Auswahl der folgenden Optionen in der "Projektnavigation": <ul style="list-style-type: none"> – PLC-Variablen – Standardvariablen-Tabelle – Register "Konstanten" des Systems – Die Hardwarekennungen aller konfigurierten Geräte werden angezeigt.
MLEN	IN	Byte, USInt, UInt	Maximale Länge in Bytes der zu empfangenden Informationen aus dem Datensatz. Ist MLENGleich 0, können so viele Alarminformationen empfangen werden, wie im Zielbereich AINFO zur Verfügung stehen.
NEW	OUT	Bool	Ein neuer Alarm wurde empfangen.
STATUS	OUT	DWord	Status der Anweisung RALRM. Weitere Informationen finden Sie unter "STATUS-Parameter für RDREC, WRREC und RALRM" (Seite 388).
ID	OUT	HW_IO (Word)	Hardwarekennung des E/A-Moduls, das den Diagnosealarm verursacht hat Hinweis: In der Beschreibung des Parameters F_ID finden Sie eine Erläuterung dazu, wie Sie die Geräte-ID ermitteln.
LEN	OUT	DWord, UInt, UInt, DInt, Real, LReal	Länge der empfangenen Alarminformationen AINFO
TINFO	IN_OUT	Variant	Aufgabeninformationen: Zielbereich für Start- und Verwaltungsinformationen des OBs. Die Länge TINFO ist immer 32 Byte.
AINFO	IN_OUT	Variant	Alarminformationen: Zielbereich für Header-Informationen und zusätzliche Alarminformationen. Für AINFO müssen Sie eine Länge von mindestens der in MLENGegebenen Bytezahl angeben, wenn MLEN größer als 0 ist. Die Länge AINFO ist variabel.

Hinweis

Wenn Sie "RALRM" in einem OB aufrufen, dessen Starterereignis kein E/A-Alarm ist, liefert die Anweisung entsprechend weniger Informationen in ihren Ausgängen.

Stellen Sie sicher, dass Sie unterschiedliche Instanz-DBs verwenden, wenn Sie "RALRM" in unterschiedlichen OBs aufrufen. Wenn Sie die Daten von einem "RALRM"-Aufruf außerhalb des zugehörigen Alarm-OBs auswerten, verwenden Sie für jedes OB-Starterereignis einen eigenen Instanz-DB.

Hinweis

Die Schnittstelle der Anweisung "RALRM" ist mit dem FB "RALRM" identisch, der in der "PROFIBUS-Richtlinie für die PROFIBUS-Kommunikation und Proxy-Funktionsbausteine nach IEC 61131-3" definiert ist.

RALRM aufrufen

Die Anweisung RALRM können Sie in drei verschiedenen Betriebsarten (MODE) aufrufen.

Tabelle 9-83 Betriebsarten der Anweisung RALRM

MODE	Beschreibung
0	<ul style="list-style-type: none"> • ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat. • Der Ausgangsparameter NEW wird auf WAHR gesetzt. • LEN erzeugt einen Ausgang von 0. • AINFO und TINFO werden nicht mit Informationen aktualisiert.
1	<ul style="list-style-type: none"> • ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat. • Der Ausgangsparameter NEW wird auf WAHR gesetzt. • LEN erzeugt einen Ausgang in der Größe der Byteanzahl der ausgegebenen AINFO-Daten. • AINFO und TINFO werden mit alarmbezogenen Informationen aktualisiert.
2	<p>Wenn die Hardwarekennung des Eingangsparameters F_ID den Alarm ausgelöst hat, gilt Folgendes:</p> <ul style="list-style-type: none"> • ID enthält die Hardwarekennung des E/A-Moduls, das den Alarm ausgelöst hat. Dieser Wert muss mit dem Wert von F_ID übereinstimmen. • Der Ausgangsparameter NEW wird auf WAHR gesetzt. • LEN erzeugt einen Ausgang in der Größe der byteanzahl der ausgegebenen AINFO-Daten. • AINFO und TINFO werden mit alarmbezogenen Informationen aktualisiert.

Hinweis

Wenn Sie einen zu kurzen Zielbereich für TINFO oder AINFO zuweisen, kann RALRM nicht die vollständigen Informationen zurückgeben.

MLEN kann die ausgegebene AINFO-Datenmenge begrenzen.

Informationen dazu, wie Sie die Daten von TINFO und AINFO interpretieren, finden Sie bei den Parametern von AINFO und TINFO im Online-Informationssystem von STEP 7.

Daten des Organisationsbausteins TInfo

Die folgende Tabelle zeigt, wie die TInfo-Daten für die Anweisung RALRM angeordnet sind:

Ebenso für OBs: Status, Aktualisierung, Profil, Diagnosefehleralarm, Ziehen oder Stecken von Modulen, Baugruppenträger- oder Stationsausfall	0	SI-Format	OB-Klasse	OB-Nr.
	4	LADDR		

TI_Submodule - OBs: Status, Aktualisierung, Profil	4		Steckplatz	
	8	Spezifikator	0	
TI_DiagnosticInterrupt - OB: Diagnosefehleralarm	4		E/A-Zustand	
	8	Kanal	Multifehler	0
TI_PlugPullModule - OB: Ziehen oder Stecken von Modulen	4		Ereignisklasse	Fehler-ID
	8	0	0	
TI_StationFailure - OB: Baugruppenträger- oder Stationsausfall	4		Ereignisklasse	Fehler-ID
	8	0	0	
Ebenso für OBs: Status, Aktualisierung, Profil, Diagnosefehleralarm, Ziehen oder Stecken von Modulen, Baugruppenträger- oder Stationsausfall	12	0		
	16			
	20	Adresse	slv_prfl	intr_type
	24	Merker1	Merker2	ID
	28 ¹	Hersteller	Instanz	

¹ Bytes 28 - 31 (Hersteller und Instanz) werden bei PROFIBUS nicht verwendet.

Hinweis

Ausführlichere Informationen zu den TINFO-Daten finden Sie im Online-Informationssystem von STEP 7.

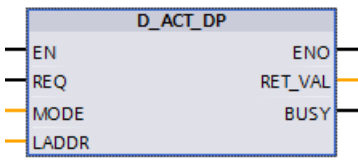
9.3.8 D_ACT_DP (DP-Slaves deaktivieren/aktivieren)

Mit der Anweisung "D_ACT_DP" können Sie konfigurierte PROFINET IO-Devices zielgerichtet aktivieren bzw. deaktivieren. Außerdem können Sie ermitteln, ob jedes zugewiesene PROFINET IO-Device gegenwärtig aktiviert oder deaktiviert ist.

Hinweis

Die Anweisung D_ACT_DP können Sie nur bei PROFINET IO-Devices verwenden. Sie können die Anweisung nicht mit PROFIBUS DP-Slaves verwenden.

Tabelle 9-84 Anweisung D_ACT_DP

KOP/FUP	SCL	Beschreibung
	<pre>"D_ACT_DP_DB" (req:=_bool_in_, mode:=_usint_in_, laddr:=_uint_in_, ret_val=>_int_out_, busy=>_bool_out_);</pre>	<p>Mit der Anweisung D_ACT_DP können Sie konfigurierte PROFINET IO-Devices aktivieren oder deaktiviert und ermitteln, ob ein zugewiesenes PROFINET IO-Device gegenwärtig aktiviert oder deaktiviert ist.</p>

- 1 STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "D_ACT_DP_SFB_DB" der Name des Instanz-DB.

Ein Gateway des Typs IE/PB Link PN IO können Sie mit der Anweisung D_ACT_DP nicht deaktivieren/aktivieren. Wenn Sie D_ACT_DP trotzdem für das genannte Gateway verwenden, gibt die CPU den Wert W#16#8093 aus (es gibt kein Hardwareobjekt, das an der in LADDR angegebenen Adresse aktiviert bzw. deaktiviert werden kann).

Hinweis

Der Deaktivierungs- bzw. Aktivierungsauftrag benötigt mehrere Läufe durch den Zykluskontrollpunkt. Deshalb können Sie in einer programmierten Schleife nicht auf das Ende eines solchen Auftrags warten.

Funktionsbeschreibung

D_ACT_DP ist eine asynchrone Anweisung, was bedeutet, dass sich die Auftragsverarbeitung über mehrere Ausführungen der Anweisung D_ACT_DP erstreckt. Sie starten den Auftrag durch Aufruf von D_ACT_DP mit REQ = 1.

Die Ausgangsparameter RET_VAL und BUSY melden den Status des Auftrags.

Anwendung

Wenn Sie in einer CPU PROFINET IO-Devices konfigurieren, die nicht tatsächlich vorhanden oder gegenwärtig nicht erforderlich sind, greift die CPU trotzdem weiterhin in regelmäßigen Abständen auf diese PROFINET IO-Devices zu. Nach der Deaktivierung dieser Geräte endet der CPU-Zugriff. Die entsprechenden Fehlerereignisse treten nicht mehr auf.

Beispiele

Aus der Sicht eines Maschinenherstellers gibt es bei der Serienherstellung von Maschinen zahlreiche mögliche Geräteoptionen. Jedoch beinhaltet jede ausgelieferte Maschine nur eine Kombination ausgewählter Optionen.

Der Hersteller konfiguriert jede dieser möglichen Maschinenoptionen als ein PROFINET IO-Device. Der Hersteller tut dies, um ein gemeinsames Anwenderprogramm mit allen möglichen Optionen entwickeln und pflegen zu können. Mit D_ACT_DP deaktivieren Sie alle PROFINET IO-Devices, die beim Anlauf einer Maschine nicht vorhanden sind.

Eine ähnliche Situation besteht bei Werkzeugmaschinen mit zahlreichen Werkzeugoptionen, von denen jedoch stets nur einige wenige verwendet werden. Diese Werkzeuge sind als

PROFINET IO-Devices implementiert. Mit `D_ACT_DP` aktiviert das Anwenderprogramm die Werkzeuge, die gegenwärtig benötigt werden, und deaktiviert diejenigen, die später erforderlich sind.

Identifizierung eines Auftrags

Wenn Sie einen Deaktivierungs- oder Aktivierungsauftrag gestartet haben und `D_ACT_DP` erneut aufrufen, bevor der Auftrag vollständig abgearbeitet ist, richtet sich das Verhalten der Anweisung danach, ob der neue Aufruf sich auf den gleichen Auftrag bezieht oder nicht. Wenn der Eingangsparameter `LADDR` identisch ist, wird der Aufruf als Nachfolgeaufruf ausgewertet.

PROFINET IO-Devices deaktivieren

Wenn Sie ein PROFINET IO-Device mit `D_ACT_DP` deaktivieren, werden die Prozessausgänge des Geräts auf die konfigurierten Ersatzwerte oder auf 0 (sicherer Zustand) gesetzt. Der zugeordnete PROFINET IO-Controller spricht diese Komponente nicht weiterhin an. Die Fehler-LEDs auf dem PROFINET IO-Controller oder der CPU melden die deaktivierten PROFINET IO-Devices nicht als fehlerhaft oder abwesend.

Die CPU aktualisiert das Prozessabbild der Eingänge von deaktivierten PROFINET IO-Devices mit 0. Deshalb behandelt die CPU die deaktivierten PROFINET IO-Devices genau wie ausgefallene PROFINET IO-Devices.

Wenn Sie auf die Anwenderdaten eines zuvor deaktivierten PROFINET IO-Device über Ihr Programm direkt zugreifen, ist das Systemverhalten von der für den Baustein ausgewählten Fehlerbehandlung abhängig:

- Wenn die globale Fehlerbearbeitung aktiviert ist, gibt das System ein Zugriffsfehler-Startereignis in den Diagnosepuffer ein und bleibt im Betriebszustand RUN.
- Wenn die bausteinlokale Fehlerbearbeitung aktiviert ist, gibt das System eine Fehlerursache in die Fehlerstruktur ein. Die Fehlerursache können Sie mit der Anweisung `GET_ERROR_ID` (Seite 305) abfragen.
Ein Fehler bei einem Lesezugriff gibt den Wert 0 aus. Weitere Informationen zur Fehlerbehandlung finden Sie unter "Prioritäten und Warteschlange für die Ausführung von Ereignissen" (Seite 86).

Wenn Sie versuchen, mit einer Anweisung (wie `RD_REC` (Seite 373)) auf ein deaktiviertes PROFINET IO-Device zuzugreifen, erhalten Sie die gleichen Fehlerinformationen in `RET_VAL` wie für ein nicht verfügbares PROFINET IO-Device.

Wenn eine PROFINET IO-Station nach der Deaktivierung durch `D_ACT_DP` ausfällt, erkennt das Betriebssystem den Ausfall nicht.

PROFINET IO-Devices aktivieren

Wenn Sie ein PROFINET IO-Device mit `D_ACT_DP` wieder aktivieren, konfiguriert und parametrisiert der zugehörige PROFINET IO-Controller die Komponente (wie bei der Rückkehr einer ausgefallenen PROFINET IO-Station). Diese Aktivierung ist beendet, wenn die Komponente in der Lage ist, Anwenderdaten zu übertragen.

Wenn Sie versuchen, mit einer Anweisung `D_ACT_DP` ein PROFINET IO-Device zu aktivieren, auf das kein Zugriff möglich ist (zum Beispiel, weil es physisch vom Bus getrennt wurde), gibt die Anweisung nach Ablauf der konfigurierten Parametrierungszeit für dezentrale Peripherie den

Fehlercode W#16#80A7 aus. Das PROFINET IO-Device wird aktiviert, und die Tatsache, dass auf das aktivierte PROFINET IO-Device nicht zugegriffen werden kann, führt zu einer entsprechenden Anzeige in der Systemdiagnose.

Kann anschließend wieder auf das PROFINET IO-Device zugegriffen werden, führt dies zu standardmäßigem Systemverhalten.

Hinweis

Die Aktivierung eines PROFINET IO-Device kann zeitaufwändig sein. Wenn Sie einen laufenden Aktivierungsauftrag abbrechen möchten, starten Sie D_ACT_DP mit dem gleichen Wert für LADDR und MODE = 2. Sie wiederholen den Aufruf von D_ACT_DP mit MODE = 2, bis der erfolgreiche Abbruch des Aktivierungsauftrags durch RET_VAL = 0 angezeigt wird.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung D_ACT_DP:

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	Bool	Pegelausgelöster Steuerparameter REQ = 1: Aktivierung oder Deaktivierung ausführen
MODE	IN	USInt	Auftragskennung Mögliche Werte: <ul style="list-style-type: none"> • 0: Informationen anfordern, ob die adressierte Komponente aktiviert oder deaktiviert ist (Ausgabe über Parameter RET_VAL) • 1: PROFINET IO-Device aktivieren • 2: PROFINET IO-Device deaktivieren
LADDR	IN	HW_DEVICE	Hardwarekennung des PROFINET IO-Device (HW_Device) Die Nummer kann den Eigenschaften des PROFINET IO-Device in der Netzsicht oder dem Register "Systemkonstanten" in der Standardvariablen-tabelle entnommen werden. Wenn dort sowohl die Kennung für die Gerätediagnose als auch die ID für Betriebszustandswechsel angegeben ist, müssen Sie den Code für die Gerätediagnose verwenden.
RET_VAL	OUT	Int	Tritt ein Fehler auf, während das Programm die Funktion ausführt, enthält der Rückgabewert einen Fehlercode.
BUSY	OUT	Bool	Aktivcode: <ul style="list-style-type: none"> • BUSY = 1: Der Auftrag ist noch aktiv. • BUSY = 0: Der Auftrag wurde beendet.

Parameter RET_VAL

Fehlercode* (W#16#...)	Bedeutung
0000	Auftrag fehlerfrei ausgeführt.
0001	Das PROFINET IO-Device ist aktiv (dieser Fehlercode ist nur bei MODE = 0 möglich).
0002	Das PROFINET IO-Device ist deaktiviert (dieser Fehlercode ist nur bei MODE = 0 möglich).
7000	Erster Aufruf mit REQ = 0: Der in LADDR angegebene Auftrag ist nicht aktiv; BUSY hat den Wert 0.
7001	Erster Aufruf mit REQ = 1: Das Programm hat den in LADDR angegebenen Auftrag ausgelöst. BUSY hat den Wert 1.
7002	Zwischenzeitlicher Aufruf (REQ irrelevant). Der aktivierte Auftrag ist noch aktiv; BUSY hat den Wert 1.
8090	<ul style="list-style-type: none"> Sie haben kein Modul mit der in LADDR angegebenen Adresse konfiguriert. Sie betreiben Ihre CPU als I-Slave/I-Device und haben eine Adresse dieses I-Slave/I-Device in LADDR angegeben.
8092	Die Deaktivierung des gegenwärtig angesprochenen PROFINET IO-Device (MODE = 2) kann nicht durch Aktivierung (MODE = 1) abgebrochen werden. Aktivieren Sie die Komponente zu einem späteren Zeitpunkt.
8093	Die in LADDR angegebene Adresse gehört nicht zu einem PROFINET IO-Device, das aktiviert oder deaktiviert werden kann, oder der Parameter MODE ist unbekannt.
8094	Sie haben versucht, ein Gerät zu aktivieren, das ein potentieller Partner für einen Werkzeugwechsel-Port ist. An diesem Werkzeugwechsel-Port ist gegenwärtig jedoch bereits ein anderes Gerät aktiviert. Das aktivierte Gerät bleibt aktiviert.
80A0	Fehler bei der Kommunikation zwischen der CPU und dem IO-Controller.
80A1	Die adressierte Komponente kann nicht parametrierbar werden. (Dieser Fehlercode ist nur möglich bei MODE = 1.) Hinweis: Wenn diese Komponente bei der Parametrierung des aktivierten Geräts erneut ausfällt, liefert die Anweisung D_ACT_DP die Fehlerinformationen. Wenn die Parametrierung eines einzelnen Moduls nicht erfolgreich ist, gibt D_ACT_DP die Fehlerinformation W#16#0000 aus.
80A3	Der betroffene PROFINET IO-Controller unterstützt diese Funktion nicht.
80A4	Die CPU unterstützt diese Funktion nicht für einen externen PROFINET IO-Controller.
80A6	Steckplatzfehler im PROFINET IO-Device; es kann nicht auf alle Anwenderdaten zugegriffen werden (dieser Fehlercode ist nur bei MODE = 1 verfügbar). Hinweis: D_ACT_DP gibt diese Fehlerinformation nur aus, wenn die aktivierte Komponente nach der Parametrierung und vor dem Ende der Ausführung der Anweisung D_ACT_DP erneut ausfällt. Wenn nur ein einzelnes Modul nicht verfügbar ist, gibt D_ACT_DP die Fehlerinformation W#16#0000 aus.
80A7	Während der Aktivierung ist eine Zeitüberschreitung aufgetreten: Das Remote-Gerät ist nicht erreichbar oder Sie haben die Parametrierungszeit für zentrale und dezentrale Peripherie zu kurz eingestellt. Der Status des Remote-Geräts ist "aktiviert", doch es ist kein Zugriff möglich.
80AA	Aktivierung mit Fehlern im PROFINET IO-Device: Unterschiede in der Konfiguration
80AB	Aktivierung mit Fehlern im PROFINET IO-Device: Parametrierfehler
80AC	Aktivierung mit Fehlern im PROFINET IO-Device: Wartung notwendig
80C1	D_ACT_DP ist gestartet und wird mit einer anderen Adresse fortgesetzt (dieser Fehlercode ist möglich, wenn MODE = 1 und MODE = 2).

Fehlercode* (W#16#...)	Bedeutung
80C3	<ul style="list-style-type: none"> • Temporärer Ressourcenfehler: Die CPU verarbeitet gegenwärtig die maximal mögliche Anzahl von Aktivierungs- und Deaktivierungsaufträgen (8). (Dieser Fehlercode ist nur möglich, wenn MODE = 1 und MODE = 2.) • Die CPU empfängt gerade eine geänderte Konfiguration. Derzeit können Sie keine PROFINET IO-Devices aktivieren/deaktivieren.
80C6	PROFINET: Vom Anwender nicht erfasste Aufträge werden beim Neustart verworfen.
Allgemeine Fehlerinformationen	Informationen für den Zugriff auf den Fehler finden Sie in der Beschreibung der Anweisung GET_ERROR_ID (Seite 305).
* Die Fehlercodes können im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt werden.	

9.3.9 STATUS-Parameter für RDREC, WRREC und RALRM

Der Ausgangsparameter STATUS enthält Fehlerinformationen, die als ARRAY[1...4] OF BYTE mit der folgenden Struktur ausgewertet werden:

Tabelle 9-85 Ausgangs-Array STATUS

Array-Element	Name	Beschreibung
STATUS[1]	Function_Num	<ul style="list-style-type: none"> • B#16#00, wenn kein Fehler vorliegt • Funktions-ID von DPV1-PDU: Tritt ein Fehler auf, wird , B#16#80 durch ODER verknüpft (beim Lesen des Datensatzes mit: B#16#DE; beim Schreiben des Datensatzes mit: B#16#DF). Wenn kein DPV1-Protokollelement verwendet wird, wird B#16#C0 ausgegeben.
STATUS[2]	Error Decode	Adresse der Fehler-ID
STATUS[3]	Error_Code_1	Fehler-ID
STATUS[4]	Error_Code_2	Herstellerspezifische Erweiterung der Fehler-ID

Tabelle 9-86 STATUS[2]-Werte

Error_decode (B#16#....)	Quelle	Beschreibung
00 bis 7F	CPU	Kein Fehler und keine Warnung
80	DPV1	Fehler gemäß IEC 61158-6
81 bis 8F	CPU	B#16#8x zeigt einen Fehler im "x-ten" Aufrufparameter der Anweisung.
FE, FF	DP-Profil	Profilspezifischer Fehler

Tabelle 9-87 STATUS[3]-Werte

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Erläuterung (DVP1)	Beschreibung
00	00		Kein Fehler, keine Warnung
70	00	Reserviert, ablehnen	Erster Aufruf; keine aktive Datensatzübertragung
	01	Reserviert, ablehnen	Erster Aufruf; Datensatzübertragung hat begonnen
	02	Reserviert, ablehnen	Zwischenzeitlicher Aufruf; Datensatzübertragung bereits aktiv

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Erläuterung (DVP1)	Beschreibung
80	90	Reserviert, übergeben	Ungültige logische Anfangsadresse
	92	Reserviert, übergeben	Unzulässiger Typ für Pointer Variant
	93	Reserviert, übergeben	Die über ID oder F_ID adressierte DP-Komponente ist nicht konfiguriert.
	96		Die Anweisung RALRM (Seite 380) kann die Anlaufinformationen, Verwaltungsinformationen, Headerinformationen und weitere Alarminformationen des OBs nicht liefern. Bei den folgenden OBs können Sie die Anweisung DPNRM_DG (Seite 401) verwenden, um die aktuellen Diagnosemeldungen des betreffenden DP-Slaves asynchron zu lesen (Adressinformationen aus den Anlaufinformationen des OBs): <ul style="list-style-type: none"> • Prozessalarm (Seite 78) • Status (Seite 84), Aktualisierung (Seite 84) oder Profil (Seite 84) • Diagnosefehleralarm (Seite 80) • Ziehen oder Stecken von Modulen (Seite 82)
	A0	Lesefehler	Negative Quittierung beim Lesen aus Modul
	A1	Schreibfehler	Negative Quittierung beim Schreiben in Modul
	A2	Modulausfall	DP-Protokollfehler in Schicht 2 (Beispiel: Slave-Ausfall oder Busprobleme)
	A3	Reserviert, übergeben	<ul style="list-style-type: none"> • PROFIBUS DP: DP-Protokollfehler mit Direct-Data-Link-Mapper oder User-Interface/User • PROFINET IO: Allgemeiner CM-Fehler
	A4	Reserviert, übergeben	Kommunikation auf dem Kommunikationsbus unterbrochen
	A5	Reserviert, übergeben	-
	A7	Reserviert, übergeben	DP-Slave oder Module belegt (temporärer Fehler).
	A8	Versionskonflikt	DP-Slave oder Modul meldet nicht kompatible Versionen.
	A9	Funktion nicht unterstützt	Funktion wird von DP-Slave oder Modul nicht unterstützt
	AA bis AF	Benutzerspezifisch	DP-Slave oder Modul meldet herstellerspezifischen Fehler in der Anwendung. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.
	B0	Ungültiger Index	Datensatz im Modul unbekannt; unzulässige Datensatznummer ≥ 256
	B1	Längenfehler beim Schreiben	Die Längenangaben im Parameter RECORD sind falsch. <ul style="list-style-type: none"> • Bei RALRM: Längenfehler in AINFO Hinweis: Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von AINFO zurückgegebenen Puffer auszuwerten sind. • Bei RDREC (Seite 373) und WRREC (Seite 373): Längenfehler in MLEN
B2	Ungültiger Steckplatz	Der konfigurierte Steckplatz ist nicht belegt.	

Error_decode (B#16#....)	Error_code_1 (B#16#....)	Erläuterung (DVP1)	Beschreibung
	B3	Typkonflikt	Tatsächlicher Modultyp entspricht nicht dem angegebenen Modultyp.
	B4	Ungültiger Bereich	DP-Slave oder Modul meldet Zugriff auf ungültigen Datenbereich.
	B5	Statuskonflikt	DP-Slave oder Modul nicht betriebsbereit
	B6	Zugriff verweigert	DP-Slave oder Modul verweigert den Zugriff.
	B7	Ungültiger Bereich	DP-Slave oder Modul meldet einen ungültigen Bereich für einen Parameter oder Wert.
	B8	Ungültiger Parameter	DP-Slave oder Modul meldet einen ungültigen Parameter.
	B9	Ungültiger Typ	DP-Slave oder Modul meldet einen ungültigen Typ: <ul style="list-style-type: none"> Bei RDREC (Seite 373): Puffer zu klein (Teilmengen können nicht gelesen werden) Bei WRREC (Seite 373): Puffer zu klein (Teilmengen können nicht geschrieben werden)
	BA bis BF	Benutzerspezifisch	DP-Slave oder Modul meldet herstellerspezifischen Fehler beim Zugriff. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.
	C0	Einschränkungskonflikt beim Lesen	<ul style="list-style-type: none"> Bei WRREC (Seite 373): Die Daten können nur geschrieben werden, wenn sich die CPU in der Betriebsart STOP befindet. Hinweis: Das bedeutet, dass die Daten nicht vom Anwenderprogramm geschrieben werden können. Sie können die Daten nur online mit einem PG/PC schreiben. Bei RDREC (Seite 373): Das Modul leitet den Datensatz, doch es liegen entweder keine Daten vor oder die Daten können nur gelesen werden, wenn sich die CPU in der Betriebsart STOP befindet. Hinweis: Wenn nur in der Betriebsart STOP Daten geschrieben werden können, ist keine Auswertung durch das Anwenderprogramm möglich. Sie können die Daten dann nur online mit einem PG/PC lesen.
	C1	Einschränkungskonflikt beim Schreiben	Die Daten der vorherigen Schreibanforderung an das Modul für denselben Datensatz wurden vom Modul noch nicht verarbeitet.
	C2	Ressource ausgelastet	Das Modul verarbeitet derzeit die maximal mögliche Anzahl von Aufträgen für eine CPU.
	C3	Ressource nicht verfügbar	Die erforderlichen Betriebsressourcen sind derzeit ausgelastet.
	C4		Interner temporärer Fehler. Der Auftrag konnte nicht durchgeführt werden. Wiederholen Sie den Auftrag. Tritt dieser Fehler häufig auf, prüfen Sie Ihre Installation auf Quellen elektrischer Störung.
	C5		DP-Slave oder Modul nicht verfügbar
	C6		Datensatzübertragung wurde wegen Prioritätsklasseneinordnung abgebrochen.

Error_decode (B#16#...)	Error_code_1 (B#16#...)	Erläuterung (DVP1)	Beschreibung
	C7		Der Auftrag wurde wegen Warmstarts oder Kaltstarts des DP-Masters abgebrochen.
	C8 bis CF		DP-Slave oder Modul meldet herstellerspezifischen Ressourcenfehler. Prüfen Sie die Dokumentation des Herstellers des DP-Slaves oder des Moduls.
	Dx	Benutzerspezifisch	DP-Slave-spezifisch. Siehe Beschreibung des DP-Slave.
81	00 bis FF		Fehler im ersten Aufrufparameter (bei RALRM (Seite 380): MODE)
	00		Unzulässige Betriebsart
82	00 bis FF		Fehler im zweiten Aufrufparameter
88	00 bis FF		Fehler im achten Aufrufparameter (bei RALRM (Seite 380): TINFO) Hinweis: Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von TINFO zurückgegebenen Puffer auszuwerten sind.
	01		Falsche Syntax-ID
	23		Quantitätsstruktur überschritten oder Zielbereich zu klein
	24		Falsche Bereichs-ID
	32		DB/DI-Nummer außerhalb des Anwenderbereichs
	3A		DB/DI-Nummer ist NULL für die Bereichs-ID DB/DI oder der angegebene DB/DI existiert nicht.
89	00 bis FF		Fehler im neunten Aufrufparameter (bei RALRM (Seite 380): AINFO) Hinweis: Das Online-Informationssystem von STEP 7 bietet unmittelbaren Zugriff auf Informationen dazu, wie die von AINFO zurückgegebenen Puffer auszuwerten sind.
	01		Falsche Syntax-ID
	23		Quantitätsstruktur überschritten oder Zielbereich zu klein
	24		Falsche Bereichs-ID
	32		DB/DI-Nummer außerhalb des Anwenderbereichs
	3A		DB/DI-Nummer ist NULL für die Bereichs-ID DB/DI oder der angegebene DB/DI existiert nicht.
8A	00 bis FF		Fehler im zehnten Aufrufparameter
8F	00 bis FF		Fehler im fünfzehnten Aufrufparameter
FE, FF	00 bis FF		Profilspezifischer Fehler

Array-Element STATUS[4]

Bei DPV1-Fehlern übergibt der DP-Master STATUS[4] an die CPU und die Anweisung. Ohne DPV1-Fehler ist dieser Wert auf 0 gesetzt, mit den folgenden Ausnahmen bei RDREC:

- STATUS[4] enthält die Zielbereichslänge von RECORD, wenn MLEN > die Zielbereichslänge aus RECORD.
- STATUS[4] = MLEN, wenn die tatsächliche Länge des Datensatzes < MLEN < die Länge des Zielbereichs aus RECORD.
- STATUS[4] = 0, wenn STATUS[4] > 255; müsste gesetzt werden

Bei PROFINET IO hat STATUS[4] den Wert 0.

9.3.10 Andere

9.3.10.1 DPRD_DAT und DPWR_DAT (Konsistente Daten lesen/schreiben)

Mit der Anweisung DPRD_DAT (Konsistente Daten lesen) lesen Sie ein oder mehrere Bytes an Daten konsistent, und mit der Anweisung DPWR_DAT (Konsistente Daten schreiben) übertragen Sie ein oder mehrere Bytes an Daten konsistent. Die Anweisung DPRD_DAT und DPWR_DAT können Sie für PROFINET und PROFIBUS einsetzen.

Tabelle 9-88 Anweisungen DPRD_DAT und DPWR_DAT

KOP/FUP	SCL	Beschreibung
<pre> DPRD_DAT - EN END - - LADDR RET_VAL - - RECORD </pre>	<pre> ret_val := DPRD_DAT(laddr:=_word_in_, record=>_variant_out_); </pre>	<p>Mit der Anweisung DPRD_DAT lesen Sie ein Byte oder mehrere Bytes an Daten aus Modulen oder Submodulen an einem der folgenden Speicherorte:</p> <ul style="list-style-type: none"> • Lokale Basis-E/A • DP-Slave • PROFINET IO-Device <p>Die CPU überträgt die gelesenen Daten konsistent. Tritt während der Datenübertragung kein Fehler auf, werden die gelesenen Daten von der CPU in den vom Parameter RECORD angegebenen Zielbereich eingegeben. Der Zielbereich muss dieselbe Länge haben wie die, die Sie für das ausgewählte Modul in STEP 7 konfiguriert haben. Bei Ausführung der Anweisung DPRD_DAT kann nur auf die Daten jeweils eines Moduls oder Submoduls zugegriffen werden. Die Übertragung beginnt an der konfigurierten Anfangsadresse.</p>
<pre> DPWR_DAT - EN END - - LADDR RET_VAL - - RECORD </pre>	<pre> ret_val := DPWR_DAT(laddr:=_word_in_, record:=_variant_in_); </pre>	<p>Verwenden Sie die Anweisung DPWR_DAT, um die Daten in RECORD konsistent in die folgenden Speicherorte zu übertragen:</p> <ul style="list-style-type: none"> • Adressiertes Modul oder Submodul im lokalen Grundgerät • DP-Standard-Slave • PROFINET IO-Device <p>Der Quellbereich muss dieselbe Länge haben wie die, die Sie für das ausgewählte Modul oder Submodul in STEP 7 konfiguriert haben.</p>

- Die S7-1200 CPU unterstützt das konsistente Lesen oder Schreiben von 1, 2 oder 4 Bytes in Peripheriegeräten. Mit der Anweisung DPRD_DAT lesen Sie Daten mit anderen Längen als 1, 2 oder 4 Bytes konsistent und mit der Anweisung DPWR_DAT schreiben Sie diese Daten konsistent.
- Diese Anweisungen können Sie für Datenbereiche von 1 Byte oder mehr Bytes verwenden. Bei Verweigerung des Zugriffs wird der Fehlercode W#16#8090 gemeldet.
- PROFINET unterstützt bis zu 1024 Bytes konsistenter Daten. Für konsistente Übertragungen zwischen der S7-1200 und PROFINET-Geräten brauchen Sie diese Anweisungen nicht zu verwenden.

Hinweis

Wenn Sie die Anweisungen DPRD_DAT und DPWR_DAT mit konsistenten Daten verwenden, müssen Sie diese konsistenten Daten aus der automatischen Aktualisierung des Prozessabbilds herausnehmen. Weitere Informationen hierzu finden Sie unter "PLC-Grundlagen: Ausführung des Anwenderprogramms" (Seite 67).

Tabelle 9-89 Parameter

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	IN	HW_IO (Word)	Hardware-ID des Moduls, aus dem die Daten gelesen werden sollen. (DPRD_DAT) Hardware-ID des Moduls, in das die Daten geschrieben werden sollen. (DPWR_DAT) Die Hardware-ID ist in den Eigenschaften des Moduls in der Gerätesicht oder in den Systemkonstanten zu finden.
RECORD	OUT	Variant	Zielbereich der gelesenen Anwenderdaten (DPRD_DAT) oder Quellbereich der zu schreibenden Anwenderdaten (DPWR_DAT). Dieser Bereich muss genauso groß sein, wie Sie im ausgewählten Modul mit STEP 7 konfiguriert haben.
RET_VAL	OUT	Int	Tritt ein Fehler auf, während die Funktion aktiv ist, enthält der Rückgabewert einen Fehlercode.

Funktionsweise von DPRD_DAT

Mit dem Parameter LADDR wählen Sie das Modul des DP-Normslaves/PROFINET IO-Device aus. Tritt bei dem adressierten Modul ein Zugriffsfehler auf, wird der Fehlercode W#16#8090 ausgegeben.

Mit dem Parameter RECORD definieren Sie die Zielbereich der gelesenen Daten:

- Der Zielbereich muss mindestens so lang sein wie die Eingänge des ausgewählten Moduls. Nur die Eingänge werden übertragen; die anderen Bytes werden nicht berücksichtigt. Wenn Sie aus einem DP-Normslave mit modularem Aufbau oder mit mehreren DP-Kennungen lesen, können Sie pro Aufruf der Anweisung DPRD_DAT auf die Daten nur eines Moduls der konfigurierten Hardwarekennung zugreifen. Wenn Sie einen zu kleinen Zielbereich auswählen, wird am Parameter RET_VAL der Fehlercode W#16#80B1 ausgegeben.
- Die folgenden Datentypen sind verwendbar: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Die Verwendung dieser Datentypen in der Datenstruktur eines anwenderdefinierten Datentyps (UDT) des Typs ARRAY oder STRUCT ist zulässig.
- Der Datentyp STRING wird nicht unterstützt.
- Trat während der Datenübertragung kein Fehler auf, werden die ausgelesenen Daten in den vom Parameter RECORD angegebenen Zielbereich eingegeben.

Funktionsweise von DPWR_DAT

Mit dem Parameter LADDR wählen Sie das Modul des DP-Normslaves/PROFINET IO-Device aus. Tritt bei dem adressierten Modul ein Zugriffsfehler auf, wird der Fehlercode W#16#8090 ausgegeben.

Mit dem Parameter RECORD definieren Sie den Quellbereich der zu schreibenden Daten:

- Der Quellbereich muss mindestens so lang sein wie die Ausgänge des ausgewählten Moduls. Nur die Ausgänge werden übertragen; die anderen Bytes werden nicht berücksichtigt. Wenn der Quellbereich am Parameter RECORD länger ist als die Ausgänge des konfigurierten Moduls, werden nur die Daten bis zur maximalen Länge der Ausgänge übertragen. Wenn der Quellbereich am Parameter RECORD kürzer ist als die Ausgänge des konfigurierten Moduls, wird am Parameter RET_VAL der Fehlercode W#16#80B1 ausgegeben.
- Die folgenden Datentypen sind verwendbar: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt. Die Verwendung dieser Datentypen in der Datenstruktur eines anwenderdefinierten Datentyps (UDT) des Typs ARRAY oder STRUCT ist zulässig.
- Der Datentyp STRING wird nicht unterstützt.
- Die Daten werden synchron übertragen, d.h. der Schreibvorgang wird zusammen mit der Anweisung beendet.

Fehlercodes

Tabelle 9-90 Fehlercodes von DPRD_DAT und DPWR_DAT

Fehlercode ¹	Beschreibung
0000	Kein Fehler
8090	Es gilt einer der folgenden Fälle: <ul style="list-style-type: none"> • Sie haben für die angegebene logische Basisadresse kein Modul konfiguriert. • Sie haben die Einschränkung bezüglich der Länge konsistenter Daten ignoriert. • Sie haben die Anfangsadresse im Parameter LADDR nicht im Hexadezimalformat eingegeben.
8092	Der Parameter RECORD unterstützt die folgenden Datentypen: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.
8093	An der in LADDR angegebenen logischen Adresse existiert kein DP-Modul/PROFINET IO-Device, aus dem konsistente Daten gelesen (DPRD_DAT) oder in das konsistente Daten geschrieben (DPWR_DAT) werden können.
80A0	Beim Zugriff auf die E/A-Geräte wurde ein Zugriffsfehler erkannt (DPRD_DAT).
80B1	Die Länge des angegebenen Zielbereichs (DPRD_DAT) oder Quellbereichs (DPWR_DAT) ist nicht mit der in STEP 7 konfigurierten Länge der Anwenderdaten identisch.
80B2	Systemfehler mit externem DP-Schnittstellenmodul (DPRD_DAT) und (DPWR_DAT)

¹ Verwenden Sie bei der Anzeige der Fehlercodes von DPRD_DAT und DPWR_DAT den Datentyp Word.

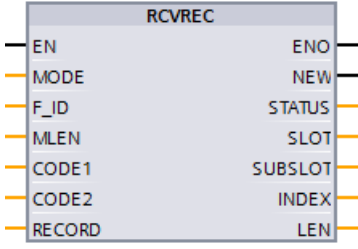
Hinweis

Wenn Sie auf DPV1-Slaves zugreifen, können Fehlerinformationen von diesen Slaves vom DP-Master an die Anweisung weitergeleitet werden.

9.3.10.2 RCVREC (Datensatz empfangen)

Ein I-Device kann einen Datensatz von einem übergeordneten Controller empfangen. Der Empfang erfolgt im Anwenderprogramm mit der Anweisung RCVREC (Datensatz empfangen).

Tabelle 9-91 Anweisung RCVREC

KOP/FUP	SCL	Beschreibung
<p>%DB1 "RCVREC_SFB_DB"</p> 	<pre>"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_);</pre>	<p>Mit der Anweisung RCVREC kann ein Datensatz von einem übergeordneten Controller empfangen werden.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "RCVREC_SFB_DB" der Name des Instanz-DBs.

Die Anweisung hat die folgenden Betriebszustände:

- Prüfen, ob das I-Device eine Anforderung für den Empfang eines Datensatzes hat
- Den Datensatz den Ausgangsparametern zur Verfügung stellen
- Eine Antwort an den übergeordneten Controller senden

Sie können den von der Anweisung ausgeführten Betriebszustand mit dem Eingangsparameter MODE ermitteln.

Das I-Device muss sich im Betriebszustand RUN oder ANLAUF befinden.

Mit MLEN geben Sie die maximale Anzahl Bytes an, die empfangen werden sollen. Die ausgewählte Länge des Zielbereichs RECORD muss mindestens die Länge der von MLEN angegebenen Bytes haben.

Wenn ein Datensatz empfangen wird (MODE = 1 oder MODE = 2), meldet der Ausgangsparameter NEW, dass der Datensatz in RECORD gespeichert ist. Achten Sie darauf, dass RECORD über eine ausreichende Länge verfügt. Der Ausgangsparameter LEN enthält die tatsächliche Länge des empfangenen Datensatzes in Bytes.

Setzen Sie CODE1 und CODE2 auf 0, um dem übergeordneten Controller eine positive Antwort zu geben. Wenn der empfangene Datensatz abgelehnt werden soll, geben Sie die negative

Antwort an den übergeordneten Controller in Fehlercode 1 von CODE1 und in Fehlercode 2 von CODE2 ein.

Hinweis

Wenn das I-Device eine Anforderung für den Empfang eines Datensatzes empfangen hat, müssen Sie die Zustellung dieser Anforderung innerhalb eines bestimmten Zeitraums anerkennen. Nach der Erkennung müssen Sie innerhalb dieses Zeitraums eine Antwort an den übergeordneten Controller senden. Ansonsten kommt es auf dem I-Device zu einem Zeitüberschreitungsfehler, woraufhin das Betriebssystem des I-Device eine negative Antwort an den übergeordneten Controller sendet. Informationen zum Wert für den Zeitraum finden Sie in den technischen Daten Ihrer CPU.

Der Ausgangsparameter STATUS empfängt die Fehlerinformationen nach Auftreten eines Fehlers.

Betriebszustände

Sie können den Betriebszustand der Anweisung RCVREC mit dem Eingangsparameter MODE ermitteln. Dieser Schritt wird in der folgenden Tabelle erläutert:

MODE	Bedeutung
0	Prüfen, ob eine Anforderung für den Empfang eines Datensatzes vorhanden ist Wenn ein Datensatz von einem übergeordneten Controller auf dem I-Device vorhanden ist, schreibt die Anweisung nur in die Ausgangsparameter NEW, SLOT, SUBSLOT, INDEX und LEN. Wenn Sie die Anweisung mehrmals mit MODE = 0 aufrufen, bezieht sich der Ausgangsparameter nur auf ein und dieselbe Anforderung.
1	Empfangen eines Datensatzes für einen beliebigen Untersteckplatz des I-Device Wenn auf dem I-Device ein Datensatz von einem übergeordneten Controller für einen beliebigen Untersteckplatz des I-Device vorhanden ist, schreibt die Anweisung in die Ausgangsparameter und überträgt den Datensatz in den Parameter RECORD.
2	Empfangen eines Datensatzes für einen spezifischen Untersteckplatz des I-Device Wenn auf dem I-Device ein Datensatz von einem übergeordneten Controller für einen spezifischen Untersteckplatz des I-Device vorhanden ist, schreibt die Anweisung in die Ausgangsparameter und überträgt den Datensatz in den Parameter RECORD.
3	Senden einer positiven Antwort an den übergeordneten Controller Die Anweisung prüft die Anforderung des übergeordneten Controllers zum Empfangen eines Datensatzes, akzeptiert den vorhandenen Datensatz und sendet eine positive Quittierung an den übergeordneten Controller.
4	Senden einer negativen Antwort an den übergeordneten Controller Die Anweisung prüft die Anforderung des übergeordneten Controllers zum Empfangen eines Datensatzes, lehnt den vorhandenen Datensatz ab und sendet eine negative Quittierung an den übergeordneten Controller. Den Grund für die Ablehnung geben Sie in die Eingangsparameter CODE1 und CODE2 ein.

Hinweis

Nach dem Empfang des Datensatzes (NEW = 1) müssen Sie die Anweisung RCVREC zweimal aufrufen, um vollständige Bearbeitung zu gewährleisten. Beachten Sie hierbei die folgende Reihenfolge:

- Erster Aufruf mit MODE = 1 oder MODE = 2
 - Zweiter Aufruf mit MODE = 3 oder MODE = 4
-

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung RCVREC:

Parameter	Deklaration	Datentyp	Beschreibung
MODE	IN	Int	Betriebsart
F_ID	IN	HW_SUBMODULE	Untersteckplatz im Übertragungsbereich des I-Device für den zu empfangenden Datensatz (nur relevant bei MODE = 2). Das hohe Wort ist immer auf null gesetzt.
MLEN	IN	Int	Maximale Länge des zu empfangenden Datensatzes in Bytes
CODE1	IN	Byte	Null (bei MODE = 3) und/oder Fehlercode 1 (bei MODE = 4)
CODE2	IN	Byte	Null (bei MODE = 3) und/oder Fehlercode 2 (bei MODE = 4)
NEW	OUT	Bool	<ul style="list-style-type: none"> MODE = 0: Neuer Datensatz wurde empfangen MODE = 1 oder 2: Datensatz wurde übertragen an RECORD
STATUS	OUT	DWord	Fehlerinformation. Weitere Informationen finden Sie unter "STATUS-Parameter" (Seite 388).
SLOT	OUT	HW_SUBMODULE	Identisch mit F_ID
SUBSLOT	OUT	HW_SUBMODULE	Identisch mit F_ID
INDEX	OUT	UInt	Nummer des empfangenen Datensatzes
LEN	OUT	UInt	Länge des empfangenen Datensatzes
RECORD	IN_OUT	Variante	Zielbereich für den empfangenen Datensatz

9.3.10.3 PRVREC (Datensatz bereitstellen)

Ein I-Device kann eine Anforderung zur Bereitstellung eines Datensatzes von einem übergeordneten Controller empfangen. Das I-Device stellt mit der Anweisung PRVREC (Datensatz bereitstellen) den Datensatz im Anwenderprogramm bereit.

Tabelle 9-92 Anweisung PRVREC

KOP/FUP	SCL	Beschreibung
<p style="text-align: center;">%DB2 *PRVREC_SFB_DB*</p>	<pre>"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_);</pre>	<p>Mit der Anweisung PRVREC kann eine Anforderung zur Bereitstellung eines Datensatzes von einem übergeordneten Controller empfangen werden.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "PRVREC_SFB_DB" der Name des Instanz-DBs.

Die Anweisung hat die folgenden Betriebszustände:

- Prüfen, ob das I-Device eine Anforderung für die Bereitstellung eines Datensatzes hat
- Den angeforderten Datensatz an den übergeordneten Controller übertragen
- Eine Antwort an den übergeordneten Controller senden

Sie können den von der Anweisung ausgeführten Betriebszustand mit dem Eingangsparameter MODE ermitteln.

Das I-Device muss sich im Betriebszustand RUN oder ANLAUF befinden.

Geben Sie mit LEN die maximale Anzahl von Bytes ein, die der zu sendende Datensatz haben soll. Die ausgewählte Länge des Zielbereichs RECORD muss mindestens die Länge der von LEN angegebenen Bytes haben.

Wenn eine Anforderung zur Bereitstellung eines Datensatzes vorhanden ist, (MODE = 0), wird der Ausgangsparameter NEW auf TRUE gesetzt.

Wenn die Anforderung zur Bereitstellung eines Datensatzes angenommen wird, schreiben Sie den angeforderten Datensatz in RECORD, um dem übergeordneten Controller eine positive Antwort zu senden, und schreiben den Wert 0 in CODE1 und CODE2. Wenn die Anforderung zur Bereitstellung eines Datensatzes abgelehnt werden soll, geben Sie die negative Antwort an den übergeordneten Controller in Fehlercode 1 von CODE1 und in Fehlercode 2 von CODE2 ein.

Hinweis

Wenn das I-Device eine Anforderung zur Bereitstellung eines Datensatzes empfangen hat, müssen Sie die Zustellung dieser Anforderung innerhalb eines bestimmten Zeitraums anerkennen. Nach der Erkennung müssen Sie innerhalb dieses Zeitraums eine Antwort an den übergeordneten Controller senden. Ansonsten kommt es auf dem I-Device zu einem Zeitüberschreitungsfehler, woraufhin das Betriebssystem des I-Device eine negative Antwort an den übergeordneten Controller sendet. Informationen zum Wert für den Zeitraum finden Sie in den technischen Daten Ihrer CPU.

Der Ausgangsparameter STATUS empfängt die Fehlerinformationen nach Auftreten eines Fehlers.

Betriebszustände

Sie können den Betriebszustand der Anweisung PRVREC mit dem Eingangsparameter MODE ermitteln. Dieser Schritt wird in der folgenden Tabelle erläutert:

MODE	Bedeutung
0	Prüfen, ob eine Anforderung für die Bereitstellung eines Datensatzes vorhanden ist Wenn auf dem I-Device eine Anforderung von einem übergeordneten Controller zur Bereitstellung eines Datensatzes vorhanden ist, schreibt die Anweisung nur in die Ausgangsparameter NEW, SLOT, SUBSLOT, INDEX und RLEN. Wenn Sie die Anweisung mehrmals mit MODE = 0 aufrufen, bezieht sich der Ausgangsparameter nur auf ein und dieselbe Anforderung.
1	Empfangen einer Anforderung zur Bereitstellung eines Datensatzes für einen beliebigen Untersteckplatz des I-Device Wenn auf dem I-Device eine solche Anforderung von einem übergeordneten Controller für einen beliebigen Untersteckplatz des I-Device vorhanden ist, schreibt die Anweisung in den Ausgangsparameter.

MODE	Bedeutung
2	Empfangen einer Anforderung zur Bereitstellung eines Datensatzes für einen spezifischen Untersteckplatz des I-Device Wenn auf dem I-Device eine solche Anforderung von einem übergeordneten Controller für einen spezifischen Untersteckplatz des I-Device vorhanden ist, schreibt die Anweisung in den Ausgangsparameter.
3	Bereitstellen des Datensatzes und Senden einer positiven Antwort an den übergeordneten Controller Die Anweisung prüft die Anforderung des übergeordneten Controllers zur Bereitstellung eines Datensatzes, stellt den Datensatz in RECORD bereit und sendet eine positive Quittierung an den übergeordneten Controller.
4	Senden einer negative Antwort an den übergeordneten Controller Die Anweisung prüft die Anforderung des übergeordneten Controllers zur Bereitstellung eines Datensatzes, lehnt diese Anforderung ab und sendet eine negative Quittierung an den übergeordneten Controller. Den Grund für die Ablehnung geben Sie in die Eingangsparameter CODE1 und CODE2 ein.

Hinweis

Nach dem Empfang einer Anforderung (NEW = 1) müssen Sie die Anweisung PRVREC zweimal aufrufen, um vollständige Bearbeitung zu gewährleisten. Beachten Sie hierbei die folgende Reihenfolge:

- Erster Aufruf mit MODE = 1 oder MODE = 2
- Zweiter Aufruf mit MODE = 3 oder MODE = 4

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung PRVREC:

Parameter	Deklaration	Datentyp	Beschreibung
MODE	IN	Int	Betriebsart
F_ID	IN	HW_SUBMODULE	Untersteckplatz im Übertragungsbereich des I-Device für den zu sendenden Datensatz (nur relevant bei MODE = 2). Das hohe Wort ist immer auf null gesetzt.
CODE1	IN	Byte	Null (bei MODE = 3) und/oder Fehlercode 1 (bei MODE = 4)
CODE2	IN	Byte	Null (bei MODE = 3) und/oder Fehlercode 2 (bei MODE = 4)
LEN	IN	UInt	Maximale Länge des zu sendenden Datensatzes in Bytes
NEW	OUT	Bool	Der neue Datensatz wurde vom übergeordneten Controller angefordert.
STATUS	OUT	DWord	Fehlerinformation. Weitere Informationen finden Sie unter "STATUS-Parameter" (Seite 388).
SLOT	OUT	HW_SUBMODULE	Identisch mit F_ID
SUBSLOT	OUT	HW_SUBMODULE	Identisch mit F_ID
INDEX	OUT	UInt	Nummer des zu sendenden Datensatzes
RLEN	OUT	UInt	Länge des zu sendenden Datensatzes
RECORD	IN_OUT	Variante	Bereitgestellter Datensatz

9.3.10.4 DPNRM_DG (Diagnosedaten eines DP-Slaves lesen)

Die Anweisung DPNRM_DG (Diagnosedaten lesen) können Sie für PROFIBUS einsetzen.

Tabelle 9-93 Anweisung DPNRM_DG


KOP/FUP	SCL	Beschreibung
	<pre>ret_val := DPNRM_DG(req:=_bool_in_, laddr:=_word_in_, record=>_variant_out_, busy=>_bool_out_);</pre>	<p>Mit der Anweisung DPNRM_DG können Sie die aktuellen Diagnosedaten eines DP-Slaves in dem Format, das in EN 50 170 Band 2, PROFIBUS angegeben ist, lesen. Die ausgelesenen Daten werden nach einer fehlerfreien Datenübertragung in den von RECORD angegebenen Zielbereich eingegeben.</p>

Tabelle 9-94 Datentypen für die Parameter von DPNRM_DG

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool REQ=1: Leseanforderung
LADDR	IN	HW_DPSLAVE Konfigurierte Diagnoseadresse des DP-Slaves: muss die Adresse der Station und nicht des E/A-Geräts sein. Um die Diagnoseadresse zu ermitteln, wählen Sie in der Netzsicht der "Gerätekonfiguration" die Station (und nicht das Abbild des Geräts). Geben Sie die Adressen im Hexadezimalformat ein. Beispiel: Diagnoseadresse 1022 bedeutet LADDR=W#16#3FE.
RET_VAL	OUT	Int Tritt ein Fehler auf, während die Funktion aktiv ist, enthält der Rückgabewert einen Fehlercode. Tritt kein Fehler auf, wird die Länge der tatsächlich übertragenen Daten in RET_VAL eingegeben.
RECORD	OUT	Variant Zielbereich für die Diagnosedaten, die gelesen wurden. Die Mindestlänge des zu lesenden Datensatzes (bzw. der Zielbereich) beträgt 6 Byte. Die maximale Länge des zu sendenden Datensatzes beträgt 240 Byte. Normslaves können mehr als 240 Bytes an Diagnosedaten liefern, maximal 244 Bytes. Dann werden nur die ersten 240 Bytes in den Zielbereich übertragen, und das Überlaufbit wird in den Daten gesetzt.
BUSY	OUT	Bool BUSY=1: Der Leseauftrag ist noch nicht beendet.

Sie starten den Leseauftrag, indem Sie dem Eingangsparameter REQ im Aufruf der Anweisung DPNRM_DG den Wert 1 zuweisen. Der Leseauftrag wird asynchron ausgeführt, in anderen Worten, es sind dafür mehrere Aufrufe der Anweisung DPNRM_DG erforderlich. Der Status des Auftrags wird von den Ausgangsparametern RET_VAL und BUSY angegeben.

Tabelle 9-95 Struktur der Slave-Diagnosedaten

Byte	Beschreibung
0	Teilnehmerstatus 1
1	Teilnehmerstatus 2
2	Teilnehmerstatus 3
3	Master-Teilnehmernummer
4	Hersteller-ID (oberes Byte)
5	Hersteller-ID (unteres Byte)
6 ...	Weitere slave-spezifische Diagnoseinformationen

Tabelle 9-96 Fehlercodes der Anweisung DPNRM_DG

Fehlercode	Beschreibung	Einschränkung
0000	Kein Fehler	-
7000	Erster Aufruf mit REQ = 0: Keine Datenübertragung aktiv; BUSY hat den Wert 0.	-
7001	Erster Aufruf mit REQ = 1: Keine Datenübertragung aktiv; BUSY hat den Wert 1.	Dezentrale E/A
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Datenübertragung ist bereits aktiv; BUSY hat den Wert 1.	Dezentrale E/A
8090	Angegebene logische Basisadresse ist ungültig: Keine Basisadresse vorhanden.	-
8092	Der Parameter RECORD unterstützt die folgenden Datentypen: Byte, Char, Word, DWord, Int, UInt, USInt, SInt, DInt, UDInt, and arrays of these types.	-
8093	<ul style="list-style-type: none"> Diese Anweisung ist für das von LADDR angegebene Modul nicht zulässig (S7-DP-Module für die S7-1200 sind zulässig). LADDR gibt statt der Station das E/A-Gerät an. Um die Diagnoseadresse für LADDR zu ermitteln, wählen Sie in der Netzsicht der "Gerätekonfiguration" die Station (und nicht das Abbild des Geräts). 	-
80A2	<ul style="list-style-type: none"> DP-Protokollfehler in Schicht 2 (Beispiel: Slave-Ausfall oder Busprobleme) Bei ET200S kann der Datensatz nicht im DPV0-Modus gelesen werden. 	Dezentrale E/A
80A3	DP-Protokollfehler mit User-Interface/User	Dezentrale E/A
80A4	Kommunikationsproblem auf dem Kommunikationsbus	Der Fehler tritt zwischen der CPU und dem externen DP-Schnittstellenmodul auf.
80B0	<ul style="list-style-type: none"> Die Anweisung ist für den Modultyp nicht zulässig. Das Modul erkennt den Datensatz nicht. Datensatznummer 241 ist nicht zulässig. 	-
80B1	Die im Parameter RECORD angegebene Länge ist falsch.	Angegebene Länge > Datensatzlänge
80B2	Der konfigurierte Steckplatz ist nicht belegt.	-
80B3	Tatsächlicher Modultyp entspricht nicht dem erforderlichen Modultyp.	-
80C0	Keine Diagnoseinformationen vorhanden.	-
80C1	Die Daten des vorherigen Schreibauftrags für denselben Datensatz auf dem Modul wurden vom Modul noch nicht verarbeitet.	-
80C2	Das Modul verarbeitet derzeit die maximal mögliche Anzahl von Aufträgen für eine CPU.	-
80C3	Die erforderlichen Ressourcen (Speicher usw.) sind derzeit ausgelastet.	-
80C4	Interner temporärer Fehler. Der Auftrag konnte nicht verarbeitet werden. Wiederholen Sie den Auftrag. Tritt dieser Fehler häufig auf, prüfen Sie Ihr System auf Quellen elektrischer Störung.	-
80C5	Dezentrale E/A nicht verfügbar	Dezentrale E/A
80C6	Datensatzübertragung wurde wegen Prioritätsklasseneinordnung abgebrochen (Neustart oder Hintergrund)	Dezentrale E/A
8xyy ¹	Allgemeine Fehlercodes	

Weitere Informationen über allgemeine Fehlercodes finden Sie unter "Erweiterte Anweisungen, Dezentrale Peripherie: Fehlerinformationen für RDREC, WRREC und RALRM" (Seite 388).

9.4 PROFlenergy

PROFlenergy ist ein hersteller- und geräteneutrales Profil für die Energieverwaltung mit PROFINET. Um den Elektrizitätsverbrauch bei Pausen in der Produktion und während ungeplanter Unterbrechungen zu reduzieren, ist es möglich, Geräte mit PROFlenergy koordiniert und zentralisiert herunterzufahren.

Der PROFINET IO-Controller schaltet die PROFINET-Geräte/Powermodule über besondere Befehle im Anwenderprogramm aus. Sie benötigen keine zusätzliche Hardware. Die PROFINET-Geräte interpretieren die PROFlenergy-Befehle direkt.

Die S7-1200 CPU unterstützt die PE-Controller-Funktionalität nicht. Die S7-1200 CPU kann nur als PROFlenergy-Einheit (mit I-Device-Funktionalität) fungieren.

PROFlenergy-Controller (PE-Controller)

Der PE-Controller ist eine höhere CPU (z. B. eine S7-1500), die den Ruhezustand von untergeordneten Geräten aktiviert bzw. deaktiviert. Der PE-Controller deaktiviert und reaktiviert bestimmte Produktionskomponenten oder ganze Produktionslinien über das Anwenderprogramm. Untergeordnete Geräte empfangen Befehle vom Anwenderprogramm über entsprechende Anweisungen (Funktionsbausteine).

Das Anwenderprogramm sendet die Befehle über das Kommunikationsprotokoll PROFINET. Der PE-Befehl kann entweder ein Steuerbefehl sein, um ein PE-Device in den Energiesparmodus zu versetzen, oder ein Befehl, um einen Status oder Messwert auszulesen.

Mit der Anweisung PE_I_DEV fordern Sie Daten von einem Modul an. Das Anwenderprogramm muss ermitteln, welche Informationen vom PE-Controller angefordert werden, und diese mit Hilfe von Datensätzen aus dem Energiemodul abrufen. Das Modul selbst unterstützt die PE-Befehle nicht direkt. Das Modul speichert die Informationen der Energiemessung in einem gemeinsam genutzten Bereich, und die niedere CPU (z. B. löst eine S7-1200) die Anweisung PE_I_DEV aus, um die Informationen an den PE-Controller auszugeben.

PROFlenergy-Einheit (PE-Device)

Das PE-Device (z. B. eine S7-1200) empfängt die PROFlenergy-Befehle des PE-Controllers (z. B. einer S7-1500) und führt diese entsprechend aus (indem sie beispielsweise einen Messwert ausgibt oder einen Energiesparmodus aktiviert). Die Implementierung des PE-Device in einem PROFlenergy-fähigen Gerät ist geräte- und herstellerabhängig.

Referenzinformationen

Weitere Informationen zu PROFlenergy finden Sie in der STEP 7-Onlinehilfe im TIA Portal. Beispiele, in denen die PROFlenergy-Anweisungen verwendet werden, finden Sie im Industry Online Support im Eintrag "PROFlenergy - Energie sparen mit SIMATIC S7 (<http://support.automation.siemens.com/WW/view/de/41986454>)".

9.5 Alarme

9.5.1 Anweisungen ATTACH und DETACH (OB und Alarmereignis einander zuweisen/Zuweisung aufheben)

Mit den Anweisungen ATTACH und DETACH können Sie Unterprogramme, die durch Alarmereignisse angestoßen werden, aktivieren und deaktivieren.

Tabelle 9-97 Anweisungen ATTACH und DETACH

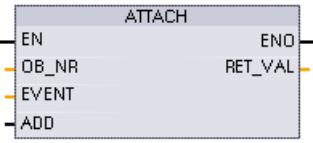
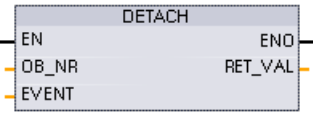
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ATTACH(ob_nr:=_int_in_, event:=_event_att_in_, add:=_bool_in_);</pre>	Die Anweisung ATTACH aktiviert die Ausführung des Unterprogramms eines Alarm-OBs bei einem Prozessereignis.
	<pre>ret_val := DETACH(ob_nr:=_int_in_, event:=_event_att_in_);</pre>	Die Anweisung DETACH deaktiviert die Ausführung des Unterprogramms eines Alarm-OBs bei einem Prozessereignis.

Tabelle 9-98 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_ATT	Organisationsbausteinkennung: Wählen Sie einen der verfügbaren Prozessalarm-OBs aus, die über die Funktion "Neuen Baustein hinzufügen" angelegt wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren OBs anzuzeigen.
EVENT	IN	EVENT_ATT	Ereigniskennung: Wählen Sie eines der verfügbaren Prozessalarmereignisse aus, die in der PLC-Gerätekonfiguration für digitale Eingänge oder schnelle Zähler aktiviert wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren Ereignisse anzuzeigen.
ADD (nur ATTACH)	IN	Bool	<ul style="list-style-type: none"> • ADD = 0 (Standard): Dieses Ereignis ersetzt alle bisherigen Ereigniszuordnungen für diesen OB. • ADD = 1: Dieses Ereignis wird zu allen bisherigen Ereigniszuordnungen für diesen OB hinzugefügt.
RET_VAL	OUT	Int	Ausführungsbedingung

Prozessalarmereignisse

Die folgenden Prozessalarmereignisse werden von der CPU unterstützt:

- Ereignisse steigende Flanke: die ersten 12 integrierten Digitaleingänge der CPU (DEa.0 bis DEb.3) und alle SB-Digitaleingänge
 - Eine steigende Flanke tritt auf, wenn der digitale Eingang in Reaktion auf den Signalwechsel eines an den Eingang angeschlossenen Geräts von AUS nach EIN wechselt.
- Ereignisse fallende Flanke: die ersten 12 integrierten Digitaleingänge der CPU (DEa.0 bis DEb.3) und alle SB-Digitaleingänge
 - Eine fallende Flanke tritt auf, wenn der digitale Eingang von EIN nach AUS wechselt.
- Ereignisse Aktueller Wert des schnellen Zählers (HSC) = Referenzwert (CV = RV) (HSC 1 bis 6)
 - Ein Alarm CV = RV wird für einen HSC generiert, wenn der aktuelle Zählwert von einem benachbarten Wert zu dem Wert wechselt, der genau dem zuvor eingerichteten Referenzwert entspricht.
- Ereignisse HSC-Richtungswechsel (HSC 1 bis 6)
 - Ein Richtungswechselereignis tritt auf, wenn erkannt wird, dass der HSC von Erhöhen zu Verringern oder von Verringern zu Erhöhen wechselt.
- Ereignisse Externes Rücksetzen HSC (HSC 1 bis 6)
 - Bestimmte HSC-Betriebsarten gestatten die Zuweisung eines digitalen Eingangs für externes Rücksetzen, um den Zählwert des HSC auf Null zurückzusetzen. Ein Ereignis Externes Rücksetzen tritt für einen solchen HSC auf, wenn dieser Eingang von AUS nach EIN wechselt.

Aktivieren von Prozessalarmereignissen in der Gerätekonfiguration

Prozessalarmer müssen während der Gerätekonfiguration aktiviert werden. Sie müssen das Kästchen zum Aktivieren der Ereignisse in der Gerätekonfiguration für einen digitalen Eingangskanal oder einen HSC selektieren, wenn Sie dieses Ereignis während der Konfiguration oder zur Laufzeit zuweisen möchten.

Mögliche Optionskästchen in der PLC-Gerätekonfiguration:

- Digitaleingang
 - Erkennung steigender Flanken aktivieren
 - Erkennung fallender Flanken aktivieren
- Schneller Zähler (HSC)
 - Diesen schnellen Zähler aktivieren
 - Alarm für Zählerwert gleich Referenzwert generieren
 - Alarm für Ereignis Externes Rücksetzen generieren
 - Alarm für Richtungswechselereignis generieren

Neue Prozessalarm-OBs in Ihr Programm einfügen

Standardmäßig wird bei der ersten Aktivierung des Ereignisses dem Ereignis kein OB zugeordnet. Dies wird durch die Kennung "<nicht verbunden>" in der Gerätekonfiguration "Prozessalarm:" gekennzeichnet. Nur Prozessalarm-OBs können einem Prozessalarmereignis zugeordnet werden. Alle vorhandenen Prozessalarm-OBs werden in der Klappliste "Prozessalarm:" aufgeführt. Wird kein OB aufgeführt, müssen Sie einen OB vom Typ "Prozessalarm" wie folgt anlegen. Gehen Sie in der Projektnavigation unter "Programmbausteine" folgendermaßen vor:

1. Doppelklicken Sie auf "Neuen Baustein hinzufügen", wählen Sie "Organisationsbaustein (OB)" und dann "Prozessalarm".
2. Optional können Sie den OB umbenennen, die Programmiersprache (KOP, FUP oder SCL) festlegen und die Bausteinnummer auswählen (schalten Sie auf manuelle Bearbeitung um und wählen Sie eine andere als die vorgeschlagene Bausteinnummer aus).
3. Bearbeiten Sie den OB und fügen Sie die Reaktion des Programms ein, die bei Auftreten des Ereignisses ausgeführt werden soll. Sie können aus diesem OB FCs und FBs bis zur maximalen Schachtelungstiefe aufrufen. Die maximale Schachtelungstiefe für Sicherheitsprogramme sind vier Ebenen. Für andere Programme beträgt die maximale Schachtelungstiefe sechs Ebenen.

Parameter OB_NR

Die Namen aller vorhandenen Prozessalarm-OBs werden in der Klappliste "Prozessalarm:" in der Gerätekonfiguration und in der Klappliste für den Parameter "OB_NR" der Anweisungen "ATTACH /DETACH" aufgeführt.

Parameter EVENT

Wenn ein Prozessalarmereignis aktiviert wird, wird diesem bestimmten Ereignis ein eindeutiger Standardereignisname zugewiesen. Diesen Ereignisnamen können Sie über das Bearbeitungsfeld "Ereignisname:" ändern, der Name muss jedoch eindeutig sein. Die Ereignisnamen werden in Variablennamen in der Variablen-tabelle "Konstanten" umgewandelt und erscheinen in der Klappliste des Parameters "EVENT" in den Anweisungsfeldern ATTACH und DETACH. Der Wert dieser Variablen ist eine interne Nummer, die das Ereignis kennzeichnet.

Allgemeine Funktionsweise

Jeder Prozessalarm kann einem Prozessalarm-OB zugeordnet werden, der bei Auftreten des Prozessalarmereignisses in die Warteschlange für die Bearbeitung gestellt wird. Die Zuordnung von OB und Ereignis kann während der Konfiguration oder zur Laufzeit erfolgen.

Sie haben die Möglichkeit, zur Zeit der Konfiguration einem aktivierten Ereignis einen OB zuzuordnen bzw. die Zuordnung aufzuheben. Um einem OB zum Zeitpunkt der Konfiguration einem Ereignis zuzuordnen, wählen Sie in der Klappliste der verfügbaren Prozessalarm-OBs unter "Prozessalarm:" einen OB aus (klicken Sie auf den abwärts gerichteten Pfeil an der Klappliste). Wählen Sie in dieser Liste den Namen des gewünschten OBs aus, oder wählen Sie "<nicht verbunden>", um die Zuordnung aufzuheben.

Sie können ein aktiviertes Prozessalarmereignis auch während der Laufzeit zuordnen bzw. die Zuordnung aufheben. Mit den Anweisungen ATTACH und DETACH können Sie dem entsprechenden OB während der Laufzeit (bei Bedarf mehrmals) ein aktiviertes

Prozessalarmereignis zuordnen bzw. die Zuordnung aufheben. Wenn gegenwärtig kein OB zugeordnet ist (entweder aufgrund der Auswahl "<nicht verbunden>" in der Gerätekonfiguration oder als Ergebnis der Ausführung von DETACH), wird der aktivierte Prozessalarm ignoriert.

Funktionsweise von DETACH

Mit der Anweisung DETACH heben Sie die Zuordnung eines bestimmten Ereignisses oder aller Ereignisse zu einem bestimmten OB auf. Ist ein EVENT angegeben, wird nur dieses eine Ereignis von dem angegebenen Organisationsbaustein OB_NR getrennt. Alle anderen gegenwärtig diesem Organisationsbaustein OB_NR zugeordneten Ereignisse bleiben zugeordnet. Ist kein EVENT angegeben, werden alle gegenwärtig diesem Organisationsbaustein OB_NR zugeordneten Ereignisse abgetrennt.

Bedingungscode

Tabelle 9-99 Bedingungscode

RET_VAL (W#16#...)	ENO	Beschreibung
0000	1	Kein Fehler
0001	1	Keine Zuordnung vorhanden (nur DETACH)
8090	0	OB ist nicht vorhanden
8091	0	OB ist vom falschen Typ
8093	0	Ereignis ist nicht vorhanden

9.5.2 Weckalarmer

9.5.2.1 SET_CINT (Weckalarm parametrieren)

Tabelle 9-100 SET_CINT (Weckalarm parametrieren)


KOP/FUP	SCL	Beschreibung
	<pre>ret_val := SET_CINT(ob_nr:=_int_in_, cycle:=_udint_in_, phase:=_udint_in_);</pre>	Der angegebene Alarm-OB wird gesetzt, um die zyklische Ausführung zu starten, die den Programmzyklus unterbricht.

Tabelle 9-101 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
OB_NR	IN	OB_CYCLIC
CYCLE	IN	UDInt
		Zeitintervall in Mikrosekunden

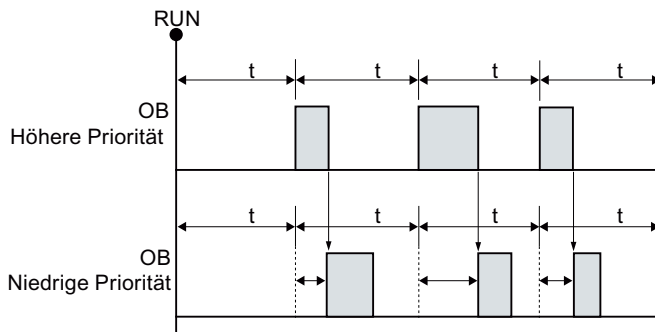
Parameter und Datentyp		Datentyp	Beschreibung
PHASE	IN	UDInt	Phasenverschiebung in Mikrosekunden
RET_VAL	OUT	Int	Ausführungsbedingung

Beispiele: Zeitparameter

- Ist die CYCLE-Zeit = 100 us, unterbricht der von OB_NR angegebene Alarm-OB den Programmzyklus alle 100 us. Der Alarm-OB wird ausgeführt und übergibt anschließend die Ausführungssteuerung wieder an den Programmzyklus an der Stelle, an der die Unterbrechung auftrat.
- Ist die CYCLE-Zeit = 0, wird das Alarmereignis deaktiviert und der Alarm-OB wird nicht ausgeführt.
- Die PHASE-Zeit (Phasenverschiebung) ist eine angegebene Verzögerungszeit, die auftritt, bevor das CYCLE-Zeitintervall beginnt. Mit der Phasenverschiebung können Sie die Ausführung von OBs mit niedrigerer Priorität zeitlich steuern.

Wenn OBs niedrigerer und höherer Priorität in demselben Zeitintervall aufgerufen werden, wird der OB niedrigerer Priorität erst aufgerufen, wenn die Bearbeitung des OBs höherer Priorität beendet ist. Der Ausführungsbeginn des OBs niedrigerer Priorität kann sich je nach Verarbeitungszeit des OBs höherer Priorität verschieben.

OB-Aufruf ohne Phasenverschiebung



Wenn Sie die Ausführung eines OBs niedrigerer Priorität mit fester Zykluszeit starten möchten, muss die Phasenverschiebung größer sein als die Verarbeitungszeit des OBs höherer Priorität.

OB-Aufruf mit Phasenverschiebung

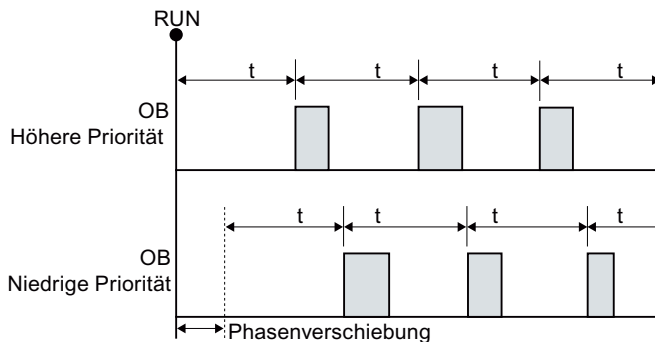


Tabelle 9-102 Bedingungscode

RET_VAL (W#16#....)	Beschreibung
0000	Kein Fehler
8090	OB ist nicht vorhanden oder hat den falschen Typ
8091	Ungültige Zykluszeit
8092	Ungültige Phasenverschiebungszeit
80B2	OB hat kein zugehöriges Ereignis

9.5.2.2 QRY_CINT (Weckalarmparameter abfragen)

Tabelle 9-103 QRY_CINT (Weckalarm abfragen)

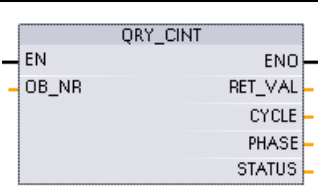
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := QRY_CINT(ob_nr:=_int_in_, cycle=>_udint_out_, phase=>_udint_out_, status=>_word_out_);</pre>	Parameter und Ausführungszustand eines Weckalarm-OBs werden abgerufen. Die zurückgegebenen Werte existierten zur Zeit der Ausführung von QRY_CINT.

Tabelle 9-104 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
OB_NR	IN	OB_CYCLIC
RET_VAL	OUT	Int
CYCLE	OUT	UDInt
PHASE	OUT	UDInt
STATUS	OUT	Word
		Beschreibung: OB-Nummer (symbolisch adressierbar, z. B. OB_MyOBName) Ausführungsbedingung Zeitintervall in Mikrosekunden Phasenverschiebung in Mikrosekunden Statuscode des Weckalarms: <ul style="list-style-type: none"> • Bits 0 bis 4, siehe STATUS-Tabelle unten • Sonstige Bits, immer 0

Tabelle 9-105 Parameter STATUS

Bit	Wert	Beschreibung
0	0	Während CPU RUN
	1	Beim Anlauf
1	0	Der Alarm ist aktiviert.
	1	Der Alarm wird über die Anweisung DIS_IRT deaktiviert.
2	0	Der Alarm ist nicht aktiv oder abgelaufen.
	1	Der Alarm ist aktiv.
4	0	Der von OB_NR angegebene OB existiert nicht.
	1	Der von OB_NR angegebene OB existiert.
Sonstige Bits		Immer 0

Tritt ein Fehler auf, zeigt RET_VAL den entsprechenden Fehlercode und den Parameter STATUS = 0 an.

Tabelle 9-106 Parameter RET_VAL

RET_VAL (W#16#....)	Beschreibung
0000	Kein Fehler
8090	OB ist nicht vorhanden oder hat den falschen Typ.
80B2	OB hat kein zugehöriges Ereignis.

9.5.3 Uhrzeitalarme

WARNUNG

Gefahr unbefugter Zugriffe auf Ihre Netzwerke durch NTP-Synchronisation (Network Time Protocol)

Wenn ein Angreifer über die NTP-Synchronisation (Network Time Protocol) auf Ihre Netzwerke zugreifen kann, kann er Ihnen möglicherweise durch Veränderung der CPU-Systemzeit die Kontrolle über Ihren Prozess entziehen. Durch eine solche Übernahme der Prozesskontrolle kann es zu Tod, schweren Verletzungen oder Sachschäden kommen.

Die NTP-Client-Funktion der S7-1200 CPU ist standardmäßig deaktiviert und ermöglicht im aktivierten Zustand lediglich konfigurierten IP-Adressen, als NTP-Server zu fungieren. Die CPU deaktiviert diese Funktion standardmäßig und Sie müssen die Funktion konfigurieren, um ferngesteuerte Korrekturen der CPU-Systemzeit zu gestatten.

Die S7-1200 CPU unterstützt Uhrzeitalarme und Uhrzeitoperationen, die von einer korrekten CPU-Systemzeit abhängig sind. Wenn Sie NTP konfigurieren und die Uhrzeitsynchronisierung von einem Server akzeptieren, müssen Sie sicherstellen, dass es sich bei dem Server um eine vertrauenswürdige Quelle handelt. Ist dies nicht der Fall, kann es zu einer Sicherheitsverletzung kommen, bei der ein unbekannter Benutzer die CPU-Systemzeit verändern und dadurch in gewissem Rahmen Ihren Prozess kontrollieren kann.

Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren "Operational Guidelines für Industrial Security" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) auf der Website "Service & Support" von Siemens.

9.5.3.1 SET_TINTL (Uhrzeitalarm festlegen)

Tabelle 9-107 SET_TINTL (Datum und Uhrzeit für Uhrzeitalarm mit Datentyp DTL festlegen)

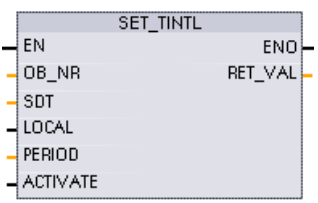
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := SET_TINTL(OB_NR:= int_in_, SDT:= dtl_in_, LOCAL:= bool_in_ PERIOD:= word_in_ ACTIVATE:= _bool_in_);</pre>	Festlegen eines Uhrzeitalarms. Der OB kann für eine Ausführung oder für wiederholte Ausführungen in einem zugewiesenen Zeitraum festgelegt werden.

Tabelle 9-108 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
OB_NR	IN	OB_TOD (INT)	OB-Nummer (symbolisch adressierbar)
SDT	IN	DTL	Anfangsdatum und -uhrzeit: Sekunden und Millisekunden werden ignoriert und können auf 0 gesetzt werden.
LOCAL	IN	Bool	0 = Systemzeit verwenden 1 = Ortszeit verwenden (sofern für die CPU die Ortszeit eingestellt ist, ansonsten verwenden Sie die Systemzeit)
PERIOD	IN	Word	Der Zeitraum ab Anfangsdatum und -uhrzeit für das wiederholt auftretende Alarmereignis. <ul style="list-style-type: none"> • W#16#0000 = einmalig • W#16#0201 = jede Minute • W#16#0401 = jede Stunde • W#16#1001 = täglich • W#16#1201 = wöchentlich • W#16#1401 = monatlich • W#16#1801 = jährlich • W#16#2001 = Monatsende
ACTIVATE	IN	Bool	0 = ACT_TINT muss ausgeführt werden, um das Alarmereignis zu aktivieren. 1 = Das Alarmereignis ist aktiviert.
RET_VAL	OUT	Int	Ausführungsbedingung

Ihr Programm kann mit der Anweisung SET_TINTL ein Datums- und Uhrzeit-Alarmereignis festlegen, das den zugewiesenen Alarm-OB ausführt. Startdatum und Startuhrzeit werden vom Parameter SDT vorgegeben, und der Zeitraum für wiederholte Alarmer (beispielsweise täglich oder wöchentlich) wird vom Parameter PERIOD festgelegt. Wenn Sie eine monatlich wiederholte Uhrzeit einstellen, müssen Sie für das Startdatum einen Tag von 1 bis 28 angeben. Die Tage 29 bis 31 dürfen nicht verwendet werden, da sie im Monat Februar nicht vorkommen. Wenn Sie ein Alarmereignis am Monatsende programmieren möchten, verwenden Sie für den Parameter PERIOD die Monatsende-Einstellung.

Der Wochentagswert der DTL-Daten im Parameter SDT wird ignoriert. Das aktuelle Datum und die aktuelle Uhrzeit einer CPU stellen Sie über die Funktion "Uhrzeit stellen" in der Ansicht

"Online & Diagnose" einer Online-CPU ein. Sie müssen Tag, Monat und Jahr einstellen. STEP 7 berechnet den Alarmzeitraum basierend auf Datum und Uhrzeit der CPU.

Hinweis

Bei der Umstellung von Sommer- nach Winterzeit ist die erste Stunde des Tages nicht vorhanden. Verwenden Sie deshalb eine Anfangszeit in der zweiten Stunde oder verwenden Sie einen zusätzlichen Verzögerungsalarm innerhalb der ersten Stunde.

Tabelle 9-109 Bedingungscode

RET_VAL (W#16#....)	Beschreibung
0000	Kein Fehler
8090	Ungültiger Parameter OB_NR
8091	Ungültiger Parameter der SDT-Anfangszeit: (beispielsweise eine Anfangszeit in der übersprungenen Stunde zu Beginn der Sommerzeit)
8092	Ungültiger PERIOD-Parameter
80A1	Die Anfangszeit liegt in der Vergangenheit. (Dieser Fehlercode tritt nur bei PERIOD = W #16#0000 auf.)

9.5.3.2 CAN_TINT (Uhrzeitalarm löschen)

Tabelle 9-110 CAN_TINT (Uhrzeitalarm löschen)

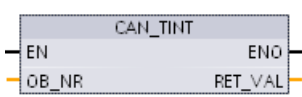
KOP/FUP	SCL	Beschreibung
	<pre>ret_val:=CAN_TINT(_int_in);</pre>	Löscht das Uhrzeitalarmereignis für den angegebenen Alarm-OB.

Tabelle 9-111 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
OB_NR	IN	OB_TOD (INT)
RET_VAL	OUT	Int

Tabelle 9-112 Bedingungscode

RET_VAL (W#16#....)	Beschreibung
0000	Kein Fehler
8090	Ungültiger Parameter OB_NR
80A0	Kein(e) Startdatum/-uhrzeit für diesen Alarm-OB eingestellt

9.5.3.3 ACT_TINT (Uhrzeitalarm aktivieren)

Tabelle 9-113 ACT_TINT (Uhrzeitalarm aktivieren)

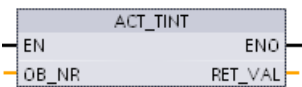
KOP/FUP	SCL	Beschreibung
	<pre>ret_val:=ACT_TINT(_int_in_);</pre>	Aktiviert das Uhrzeitalarmereignis für den angegebenen Alarm-OB.

Tabelle 9-114 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_TOD (INT)	OB-Nummer (symbolisch adressierbar)
RET_VAL	OUT	Int	Ausführungsbedingung

Tabelle 9-115 BedingungsCodes

RET_VAL (W#16#....)	Beschreibung
0000	Kein Fehler
8090	Ungültiger Parameter OB_NR
80A0	Startdatum und -uhrzeit für den Uhrzeitalarm-OB nicht eingestellt
80A1	Die aktivierte Zeit liegt in der Vergangenheit. Der Fehler tritt nur auf, wenn der Alarm-OB nur für die einmalige Ausführung eingestellt ist.

9.5.3.4 QRY_TINT (Status des Uhrzeitalarms abfragen)

Tabelle 9-116 QRY_TINT (Uhrzeitalarm abfragen)

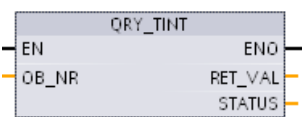
KOP/FUP	SCL	Beschreibung
	<pre>ret_val:=QRY_TINT(OB_NR:=_int_in_, STATUS=>_word_out_);</pre>	Fragt den Uhrzeitalarmstatus für den angegebenen Alarm-OB ab.

Tabelle 9-117 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_TOD (INT)	OB-Nummer (symbolisch adressierbar) des Alarm-OBs für die Abfrage
RET_VAL	OUT	Int	Ausführungsbedingung
STATUS	OUT	Word	Status des angegebenen Alarm-OBs

Tabelle 9-118 Parameter STATUS

Bit	Wert	Beschreibung
0	0	In Run
	1	Beim Anlauf
1	0	Der Alarm ist aktiviert.
	1	Der Alarm ist deaktiviert.
2	0	Der Alarm ist nicht aktiv oder abgelaufen.
	1	Der Alarm ist aktiv.
4	0	Die zugewiesene OB_NR ist nicht vorhanden.
	1	Ein OB mit der zugewiesenen OB_NR ist vorhanden.
6	1	Uhrzeitalarm verwendet Ortszeit.
	0	Uhrzeitalarm verwendet Systemzeit.
Sonstiges		Immer 0

Tabelle 9-119 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Ungültiger Parameter OB_NR

9.5.4 Verzögerungsalarme

Sie können die Bearbeitung des Verzögerungsalarms mit den Anweisungen SRT_DINT und CAN_DINT starten und abbrechen und den Alarmzustand mit der Anweisung QRY_DINT abfragen. Jeder Verzögerungsalarm ist ein einmaliges Ereignis, das nach einer vorgegebenen Verzögerungszeit auftritt. Wird das Zeitverzögerungsereignis gelöscht, bevor die Zeitverzögerung abgelaufen ist, tritt der Alarm nicht im Programm auf.

Tabelle 9-120 Anweisungen SRT_DINT, CAN_DINT und QRY_DINT

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := SRT_DINT(ob_nr:=_int_in_, dtime:=_time_in_, sign:=_word_in_);</pre>	SRT_DINT startet einen Verzögerungsalarm, der einen OB ausführt, wenn die im Parameter DTIME festgelegte Verzögerungszeit abgelaufen ist.
	<pre>ret_val := CAN_DINT(ob_nr:=_int_in_);</pre>	CAN_DINT löscht einen Verzögerungsalarm, der bereits gestartet ist. Der Verzögerungsalarm-OB wird dann nicht ausgeführt.
	<pre>ret_val := QRY_DINT(ob_nr:=_int_in_, status=>_word_out_) ;</pre>	QRY_DINT fragt den Zustand des vom Parameter OB_NR angegebenen Verzögerungsalarms ab.

Tabelle 9-121 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
OB_NR	IN	OB_DELAY	Organisationsbaustein (OB), der nach einer Zeitverzögerung gestartet werden soll: Wählen Sie einen der verfügbaren Verzögerungsalarm-OBs aus, die über die Funktion "Neuen Baustein hinzufügen" in der Projektnavigation angelegt wurden. Doppelklicken Sie auf das Parameterfeld und dann auf das Symbol, um die verfügbaren OBs anzuzeigen.
DTIME ¹	IN	Time	Verzögerungswert (1 bis 60000 ms)
SIGN ¹	IN	Word	Von der S7-1200 nicht verwendet: Jeder Wert wird angenommen. Um Fehler zu verhindern, muss ein Wert zugewiesen werden.
RET_VAL	OUT	Int	Ausführungsbedingung
STATUS	OUT	Word	Anweisung QRY_DINT: Zustand des angegebenen Verzögerungsalarm-OBs, siehe Tabelle unten

¹ Nur bei SRT_DINT

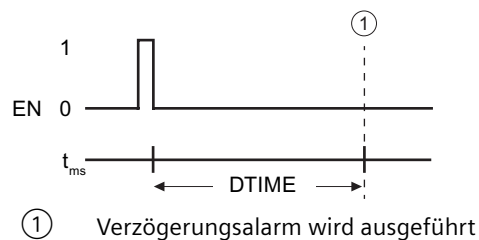
Bedienung

Ist EN=1, startet Anweisung SRT_DINT die interne Zeitverzögerung (DTIME). Wenn die Zeitverzögerung abläuft, generiert die CPU eine Programmunterbrechung, die die Ausführung des zugeordneten Verzögerungsalarm-OBs anstößt. Mit der Anweisung CAN_DINT können Sie einen gestarteten Verzögerungsalarm abbrechen, bevor die angegebene Verzögerungszeit auftritt. Insgesamt dürfen maximal vier Verzögerungsalarmereignisse aktiv sein.

Hinweis

SRT_DINT startet die Zeit für die Zeitverzögerung in jedem Zyklus, wenn EN = 1. Bestätigen Sie EN = 1 einmalig, statt EN = 1 einfach zu setzen, um die Zeitverzögerung zu starten.

Zeitdiagramm für Anweisung SRT_DINT:



Verzögerungsalarm-OBs in Ihr Projekt einfügen

Sie können Verzögerungsalarm-OBs nur in den Anweisungen SRT_DINT und CAN_DINT hinzufügen. In einem neuen Projekt ist noch kein Verzögerungsalarm-OB vorhanden. Sie müssen Verzögerungsalarm-OBs in Ihr Projekt einfügen. Um einen Verzögerungsalarm-OB anzulegen, gehen Sie folgendermaßen vor:

1. Doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", wählen Sie "Organisationsbaustein (OB)" und dann "Verzögerungsalarm".
2. Sie können den OB umbenennen, die Programmiersprache festlegen und die Bausteinnummer auswählen. Schalten Sie auf manuelle Bearbeitung um, wenn Sie eine andere als die automatisch zugeordnete Bausteinnummer auswählen möchten.
3. Bearbeiten Sie das Unterprogramm des Verzögerungsalarm-OBs und fügen Sie die Reaktion des Programms ein, die bei Auftreten des Verzögerungsereignisses ausgeführt werden soll. Sie können andere FC- und FB-Codebausteine aus dem Verzögerungsalarm-OB aufrufen. Die maximale Schachtelungstiefe für Sicherheitsprogramme sind vier Ebenen. Für andere Programme beträgt die maximale Schachtelungstiefe sechs Ebenen.
4. Die Namen der neu zugewiesenen Verzögerungsalarm-OBs stehen zur Verfügung, wenn Sie den Parameter OB_NR der Anweisungen SRT_DINT und CAN_DINT bearbeiten.

Parameter STATUS von QRY_DINT

Tabelle 9-122 Wenn ein Fehler (REL_VAL \neq 0) vorliegt, ist STATUS = 0.

Bit	Wert	Beschreibung
0	0	In RUN
	1	Beim Anlauf
1	0	Der Alarm ist aktiviert.
	1	Der Alarm ist deaktiviert.
2	0	Der Alarm ist nicht aktiv oder abgelaufen.
	1	Der Alarm ist aktiv.
4	0	Ein OB mit der in OB_NR angegebenen OB-Nummer existiert nicht.
	1	Ein OB mit der in OB_NR angegebenen OB-Nummer existiert.
Sonstige Bits		Immer 0

Bedingungscode

Tabelle 9-123 Bedingungscode für SRT_DINT, CAN_DINT und QRY_DINT

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Falscher Parameter OB_NR
8091	Falscher Parameter DTIME
80A0	Verzögerungsalarm noch nicht gestartet.

9.5.5 Anweisungen DIS_AIRT und EN_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen verzögern/aktivieren)

Die Alarmbearbeitung kann mit den Anweisungen DIS_AIRT und EN_AIRT aktiviert und deaktiviert werden.

Tabelle 9-124 Anweisungen DIS_AIRT und EN_AIRT

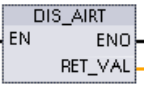
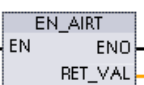
KOP/FUP	SCL	Beschreibung
	DIS_AIRT () ;	DIS_AIRT verzögert die Bearbeitung neuer Alarmereignisse. Sie können DIS_AIRT in einem OB mehrmals ausführen.
	EN_AIRT () ;	EN_AIRT aktiviert die Bearbeitung von Alarmereignissen, die zuvor mit der Anweisung DIS_AIRT deaktiviert wurden. Jede Ausführung von DIS_AIRT muss mit einer Ausführung von EN_AIRT gelöscht werden. Die Anweisungen EN_AIRT müssen in demselben OB oder in einer aus demselben OB aufgerufenen FC bzw. FB ausgeführt werden, bevor für diesen OB erneut Alarme aktiviert werden.

Tabelle 9-125 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
RET_VAL	OUT	Int
		Anzahl Verzögerungen = Anzahl ausgeführter DIS_AIRT in der Warteschlange.

Die Anzahl der Ausführungen von DIS_AIRT werden vom Betriebssystem gezählt. Jede dieser Anweisungen bleibt gültig, bis sie mit einer Anweisung EN_AIRT gelöscht wird oder bis der aktuelle OB vollständig abgearbeitet ist. Beispiel: Wenn Sie die Alarmbearbeitung fünfmal mit fünf Ausführungen von DIS_AIRT deaktiviert haben, müssen Sie diese mit fünf Ausführungen von EN_AIRT stornieren, damit die Alarmbearbeitung wieder aktiviert ist.

Nach der erneuten Aktivierung der Alarmereignisse werden die Alarme, die während der Ausführung von DIS_AIRT auftraten, bearbeitet bzw. die Alarme werden bearbeitet, sobald der aktuelle OB ausgeführt wurde.

Der Parameter RET_VAL zeigt an, wie häufig die Alarmverarbeitung deaktiviert wurde, also wie viele Ausführungen von DIS_AIRT in die Warteschlange gestellt wurden. Die Alarmverarbeitung wird erst wieder aktiviert, wenn der Parameter RET_VAL = 0 ist.

9.6 Alarme

9.6.1 Gen_UsrMsg (Anwenderdiagnosemeldungen erzeugen)

Tabelle 9-126 Anweisung Gen_UsrMsg

KOP/FUP	SCL	Beschreibung
<p>Gen_UsrMsg</p> <ul style="list-style-type: none"> - EN - Mode - TextID - TextListID - AssocValues <p style="text-align: right;">ENO Ret_Val</p>	<pre>ret_val :=Gen_UsrMsg(Mode:=_uint_in_, TextID:=_uint_in_, TextListID:=_uint_in_, AssocValues:=_struct_inout_);</pre>	<p>Mit der Anweisung "Gen_UsrMsg" erzeugen Sie einen Anwenderdiagnosealarm, bei dem es sich entweder um einen kommenden oder gehenden Alarm handeln kann. Mit Hilfe von Anwenderdiagnosealarmen können Sie einen Anwendereintrag in den Diagnosepuffer schreiben und einen entsprechenden Alarm senden. Der Eintrag im Diagnosepuffer wird synchron erstellt. Die Alarmübertragung erfolgt asynchron.</p> <p>Tritt während der Ausführung der Anweisung ein Fehler auf, wird er über den Parameter RET_VAL ausgegeben.</p>

Inhalt des Alarms

Eine Textliste definiert den Inhalt des Alarms:

- Sie definieren die zu verwendende Textliste mit dem Parameter TextListID. Zu diesem Zweck öffnen Sie in der Projektnavigation den Dialog "Textlisten". Zeigen Sie im Dialog "Textlisten" die Spalte "ID" an. Übernehmen Sie die ID am Parameter TextListID.
- Wählen Sie über den Parameter TextID den Eintrag in der Textliste aus, den Sie in den Diagnosepuffer schreiben möchten. Wählen Sie hierzu im Dialog "Textlisteneinträge" einen Eintrag aus, indem Sie am Parameter TextID eine Nummer aus den Spalten "Bereich von/ Bereich bis" verwenden. Sie müssen für beide Spalten "Bereich von" und "Bereich bis" der Textlisteneinträge die gleiche Nummer verwenden.

Weitere Informationen zu Textlisten finden Sie im STEP 7 Informationssystem.

Begleitwerte definieren

Der Textlisteneintrag definiert neue Begleitwerte, die dem Alarm hinzugefügt werden sollen:

- Fügen Sie dem Textlisteneintrag die folgenden Informationen zu, um Begleitwerte zu definieren:
@<Nr. des Begleitwerts><Elementtyp><Formatangabe>@
- Geben Sie über den Systemdatentyp AssocValues an, welcher Begleitwert beim Erzeugen des Alarms hinzugefügt werden soll.

Weitere Informationen zur Struktur von Begleitwerten finden Sie im STEP 7 Informationssystem.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "Gen_UsrMsg":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
Mode	Input	UInt	E, A, M, D, L oder Konstante	Parameter zum Auswählen des Status des Alarms: <ul style="list-style-type: none"> • 1: kommender Alarm • 2: gehender Alarm
TextID	Input	UInt	E, A, M, D, L oder Konstante	ID des Textlisteneintrags, der für den Alarmtext verwendet werden soll.
TextListID	Input	UInt	E, A, M, D, L oder Konstante	ID der Textliste, die den Textlisteneintrag enthält.
Ret_Val	Return	Int	E, A, M, D, L	Fehlercode der Anweisung.
AssocValues	InOut	VARIANT	D, L	Pointer auf den Systemdatentyp AssocValues, mit dem Sie die Begleitwerte definieren können.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Parameter AssocValues

Mit dem Systemdatentyp AssocValues definieren Sie, welche Begleitwerte gesendet werden. Maximal sind acht Begleitwerte möglich. Geben Sie den Datentyp "AssocValues" als Datenbaustein ein, um die Struktur anzulegen.

Sie wählen Begleitwerte aus, indem Sie die Nummern der Begleitwerte für die Parameter Value[x] eingeben. Beachten Sie bitte den folgenden Hinweis:

- Die Anweisung Gen_UsrMsg behandelt die Werte für TextID und TextListID als zu sendende Begleitwerte. Deshalb sind die Zahlen "1" und "2" vorab für die Adressierung von Begleitwerten zugewiesen. Verwenden Sie die Zahlen "1" und "2" nicht zum Adressieren von Begleitwerten.
- Adressieren Sie den Begleitwert am Parameter Value [1] als Nummer "3", an Parameter Value [2] als Nummer "4" usw.

Byte	Parameter	Datentyp	Startwert	Beschreibung	Nummer des Begleitwerts
0..1	Value[1]	UINT	0	Erster Begleitwert des Alarms	3
2..3	Value[2]	UINT	0	Zweiter Begleitwert des Alarms	4
4..5	Value[3]	UINT	0	...	5
6..7	Value[4]	UINT	0	...	6
8..9	Value[5]	UINT	0	...	7
10..11	Value[6]	UINT	0	...	8

Byte	Parameter	Datentyp	Startwert	Beschreibung	Nummer des Begleitwerts
12..13	Value[7]	UINT	0	...	9
14..15	Value[8]	UINT	0	Achter Begleitwert des Alarms	10

Parameter RET_VAL

Die folgende Tabelle definiert die Ausgangswerte für den Parameter RET_VAL . Siehe auch Gemeinsame Fehlercodes für die erweiterten Anweisungen (Seite 535).

Fehlercode* (W#16#...)	Bedeutung
0000	Kein Fehler
8080	Der Wert im Parameter MODE wird nicht unterstützt.
80C1	Ressourcen-Engpass aufgrund zu vieler paralleler Aufrufe.
8528	Parameter 5 (AssocValues) entspricht nicht den Bytegrenzen.
853A	Parameter 5 (AssocValues) referenziert einen ungültigen Punkt.
*Fehlercodes können im Programmierer als Ganzzahlen oder Hexadezimalwerte angezeigt werden.	

9.7 Diagnose (PROFINET oder PROFIBUS)

9.7.1 Diagnoseanweisungen

Die folgenden Diagnoseanweisungen können entweder mit PROFINET oder PROFIBUS verwendet werden:

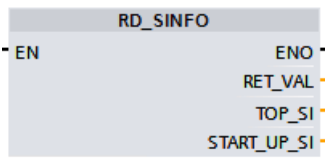
- Anweisung RD_SINFO (Seite 421): Liest die Startinformation des aktuellen OBs
- Anweisung LED (Seite 430): Liest den Zustand der LEDs eines dezentralen Peripheriegeräts
- Anweisung Get_IM_Data (Seite 431): Prüft die Identifikations- und Wartungsdaten (I&M) für ein angegebenes Modul oder Submodul
- Anweisung Get_Name (Seite 433): Liest den Namen eines PROFINET IO-Device, PROFIBUS-Slaves oder AS-i-Slaves
- Anweisung GetStationInfo (Seite 439): Liest die IP- oder MAC-Adresse eines PROFINET IO-Device im lokalen IO-System oder eines PROFINET IO-Device in einem untergeordneten IO-System (angeschlossen über CP/CM-Module).
- Anweisung DeviceStates (Seite 449): Ruft die Betriebszustände eines dezentralen Peripheriegeräts in einem E/A-Untersystem ab
- Anweisung ModuleStates (Seite 454): Ruft die Betriebszustände der Module in einem dezentralen Peripheriegerät ab
- Anweisung GET_DIAG (Seite 459): Liest die Diagnoseinformationen aus einem angegebenen Gerät aus

Hinweis

Die Anweisung GetStationInfo können Sie nur bei PROFINET IO-Devices verwenden. Sie können die Anweisung nicht mit PROFIBUS DP-Slaves verwenden.

9.7.2 RD_SINFO (Startinformation des aktuellen OBs auslesen)**Beschreibung**

Tabelle 9-127 Anweisung RD_SINFO

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := RD_SINFO(TOP_SI=>_variant_out_, START_UP_SI=>_variant_out_) ;</pre>	<p>Mit der Anweisung "RD_SINFO" lesen Sie die Startinformation der folgenden OBs:</p> <ul style="list-style-type: none"> • Letzter aufgerufener OB, der noch nicht vollständig ausgeführt ist • Letzter Anlauf-OB, den die CPU gestartet hat <p>In beiden Fällen gibt es keinen Zeitstempel. Wenn der Aufruf in OB 100, OB 101 oder OB 102 stattfindet, werden zwei identische Startinformationsmeldungen ausgegeben.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "RD_SINFO":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
RET_VAL	Return	INT	E, A, M, D, L	Fehlerinformationen
TOP_SI	Output	VARIANT	D, L	Startinformation des aktuellen OBs
START_UP_SI	Output	VARIANT	D, L	Startinformation des zuletzt gestarteten Anlauf-OBs

Ausführlichere Informationen zu den gültigen Datentypen enthält das Kapitel "Datentypen (Seite 105)".

SDTs des Parameters TOP_SI

In der folgenden Tabelle werden die möglichen SDTs für den Parameter TOP_SI gezeigt:

Organisationsbausteine (OB)	Systemdatentypen (SDT)	Nummern der Systemdatentypen
Beliebig	SI_classic*	592*
	SI_none	593
ProgramCycleOB	ProgramCycleOB	594
TimeOfDay	TimeOfDay	595

Organisationsbausteine (OB)	Systemdatentypen (SDT)	Nummern der Systemdatentypen
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

*Der SDT SI_classic gilt nicht für die S7-1200. Die S7-1200 CPU gibt einen RET_VAL von #16#8081 aus, wenn der Parameter TOP_SI vom Typ SI_classic ist.

SDTs des Parameters START_UP_SI

In der folgenden Tabelle werden die möglichen SDTs für den Parameter START_UP_SI gezeigt:

Systemdatentypen (SDT)	Nummern der Systemdatentypen
SI_classic*	592
SI_none	593
SI_Startup	613

*Der SDT SI_classic gilt nicht für die S7-1200. Die S7-1200 CPU gibt einen RET_VAL von #16#8083 aus, wenn der Parameter START_UP_SI vom Typ SI_classic ist.

Strukturen

Die folgenden Tabellen definieren die Strukturelemente der einzelnen Strukturen:

Tabelle 9-128 Struktur SI_classic

Strukturelement	Datentyp	Beschreibung
EV_CLASS	BYTE	<ul style="list-style-type: none"> Bits 0 bis 3: Ereignis-ID Bits 4 bis 7: Ereignisklasse
EV_NUM	BYTE	Ereignisnummer

Strukturelement	Datentyp	Beschreibung
PRIORITY	BYTE	Nummer der Prioritätsklasse (Bedeutung von B#16#FE: OB nicht verfügbar oder deaktiviert oder kann nicht im aktuellen Betriebszustand gestartet werden)
NUM	BYTE	OB-Nummer
TYP2_3	BYTE	Daten-ID 2_3: Identifiziert die in ZI2_3 eingegebenen Informationen
TYP1	BYTE	Daten-ID 1: Identifiziert die in ZI1 eingegebenen Informationen
ZI1	WORD	Weitere Informationen 1
ZI2_3	DWORD	Weitere Informationen 2_3

Tabelle 9-129 Struktur SI_none

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)

Tabelle 9-130 Struktur SI_ProgramCycle

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 1	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Initial_Call	BOOL	Für OB_Class = 1, 30, 52, 61, 65
Remanence	BOOL	Für OB_Class = 1

Tabelle 9-131 Struktur SI_TimeOfDay

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 10	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
CaughtUp	BOOL	Für OB_Class = 10
SecondTime	BOOL	Für OB_Class = 10

Tabelle 9-132 Struktur SI_Delay

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 20	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Sign	WORD	Für OB_Class = 20

Tabelle 9-133 Struktur SI_Cyclic

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 30	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Initial_Call	BOOL	Für OB_Class = 1, 30, 52, 61, 65
Event_Count	INT	Für OB_Class = 30, 51, 52, 61, 65, 91, 92

Tabelle 9-134 Struktur SI_HWInterrupt

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 40	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_IO	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
USI	WORD	Für OB_Class = 40
IChannel	USINT	Für OB_Class = 40
EventType	BYTE	Für OB_Class = 40

Tabelle 9-135 Struktur SI_Submodule

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_IO	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Slot	UINT	Für OB_Class = 55, 56, 57
Specifier	WORD	Für OB_Class = 55, 56, 57

Tabelle 9-136 Struktur SI_SynchCycle

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 61	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Initial_Call	BOOL	Für OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Für OB_Class = 61, 91, 92
PIP_Output	BOOL	Für OB_Class = 61, 91, 92
IO_System	USINT	Für OB_Class = 61, 91, 92
Event_Count	INT	Für OB_Class = 30, 51, 52, 61, 65, 91, 92
SyncCycleTime	LTIME	Berechnete Zykluszeit

Tabelle 9-137 Struktur SI_IORedundancyError

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 70	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_ANY	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Für OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Für OB_Class = 70, 80, 83, 85, 86

Tabelle 9-138 Struktur SI_CPURedundancyError

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 72	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Switch_Over	BOOL	Für OB_Class = 72

Tabelle 9-139 Struktur SI_TimeError

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 80	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Fault_ID	BYTE	Für OB_Class = 70, 80, 83, 85, 86

Strukturelement	Datentyp	Beschreibung
Csg_OBnr	OB_ANY	Für OB_Class = 80
Csg_Prio	UINT	Für OB_Class = 80

Tabelle 9-140 Struktur SI_DiagnosticInterrupt

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 82	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
IO_State	WORD	Für OB_Class = 82
LADDR	HW_ANY	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Channel	UINT	Für OB_Class = 82
MultiError	BOOL	Für OB_Class = 82

Tabelle 9-141 Struktur SI_PlugPullModule

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 83	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_IO	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Für OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Für OB_Class = 70, 80, 83, 85, 86

Tabelle 9-142 Struktur SI_AccessError

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 85	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_IO	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Für OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Für OB_Class = 70, 80, 83, 85, 86
IO_Addr	UINT	Für OB_Class = 85
IO_LEN	UINT	Für OB_Class = 85

Tabelle 9-143 Struktur SI_StationFailure

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 86	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LADDR	HW_IO	Für OB_Class = 40, 51, 55, 56, 57, 70, 82, 83, 85, 86, 91, 92
Event_Class	BYTE	Für OB_Class = 70, 83, 85, 86
Fault_ID	BYTE	Für OB_Class = 70, 80, 83, 85, 86

Tabelle 9-144 Struktur SI_Servo

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 91	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Initial_Call	BOOL	Für OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Für OB_Class = 61, 91, 92
PIP_Output	BOOL	Für OB_Class = 61, 91, 92
IO_System	USINT	Für OB_Class = 61, 91, 92
Event_Count	INT	Für OB_Class = 30, 51, 52, 61, 65, 91, 92
Synchronous	BOOL	

Tabelle 9-145 SI_Ipo SI_Ipo

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 92	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
Initial_Call	BOOL	Für OB_Class = 1, 30, 52, 61, 65
PIP_Input	BOOL	Für OB_Class = 61, 91, 92
PIP_Output	BOOL	Für OB_Class = 61, 91, 92
IO_System	USINT	Für OB_Class = 61, 91, 92
Event_Count	INT	Für OB_Class = 30, 51, 52, 61, 65, 91, 92
Reduction	UINT	Für OB_Class = 92

Tabelle 9-146 Struktur SI_Startup

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT := 100	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
LostRetentive	BOOL	Für OB_Class = 100
LostRTC	BOOL	Für OB_Class = 100

Tabelle 9-147 Struktur SI_ProgIOAccessError

Strukturelement	Datentyp	Beschreibung
SI_Format	USINT	<ul style="list-style-type: none"> 16#FF = Keine Informationen 16#FE = Optimierte Startinformation
OB_Class	USINT	OB-Klasse für "Keine Informationen" oder "Optimierte Startinformation"
OB_Nr	UINT	OB-Nummer (1 ... 32767)
BlockNr	UINT	Für OB_Class = 121, 122
Reaction	USINT	Für OB_Class = 121, 122
Fault_ID	BYTE	Für OB_Class = 121, 122
BlockType	USINT	Für OB_Class = 121, 122
Area	USINT	Für OB_Class = 121, 122
DBNr	DB_ANY	Für OB_Class = 121, 122
Csg_OBNr	OB_ANY	Für OB_Class = 121, 122
Csg_Prio	USINT	Für OB_Class = 121, 122
Width	USINT	Für OB_Class = 121, 122

Hinweis

Bei Erstellung mit der Bausteineigenschaft "Standard" sind die für die Struktur SI_classic angegebenen Strukturelemente inhaltlich identisch mit den temporären Variablen eines OBs.

Beachten Sie jedoch, dass temporäre Variablen der einzelnen OBs unterschiedliche Namen und unterschiedliche Datentypen haben können. Beachten Sie auch, dass die Aufrufchnittstelle jedes OBs weitere Informationen zu Datum und Uhrzeit der OB-Anforderung enthalten.

Die Bits 4 bis 7 des Strukturelements EV_CLASS enthalten die Ereignisklasse. Die folgenden Werte sind hier zulässig:

- 1: Startereignisse von Standard-OBs
- 2: Startereignisse von synchronen Fehler-OBs
- 3: Startereignisse von asynchronen Fehler-OBs

Das Strukturelement PRIORITY liefert die Prioritätsklasse des aktuellen OBs.

Neben diesen beiden Elementen ist auch NUM relevant. NUM enthält die Nummer des aktuellen OBs oder den zuletzt gestarteten Anlauf-OB.

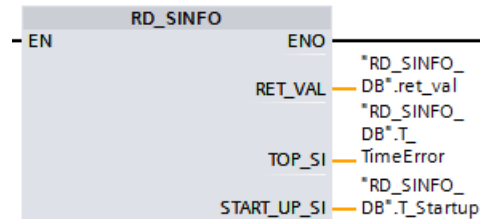
Parameter RET_VAL

Die folgende Tabelle zeigt die Bedeutung der Werte des Parameters RET_VAL:

Fehlercode* (W#16#...)	Bedeutung
8081	Startinformation des aktuellen OBs entspricht nicht dem angegebenen Systemdatentyp
8083	Startinformation des zuletzt gestarteten Anlauf-OBs entspricht nicht dem angegebenen Systemdatentyp
*Fehlercodes können im Programmierer als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden.	

Beispiel

Ein Zeitfehleralarm-OB (OB 80) ist der OB, der zuletzt aufgerufen wurde und der noch nicht vollständig verarbeitet wurde. Der Anlauf-OB (OB 100) ist der Anlauf-OB, der zuletzt gestartet wurde. Der Anweisungsaufzurum Auslesen der Anlaufinformation lautet wie folgt, wobei RD_SINFO_DB der Datenbaustein ist, der Variablen der SDTs für die OB-Typen enthält:



Die folgende Tabelle zeigt die Zuweisung der Strukturelemente des Parameters TOP_SI der Anweisung RD_SINFO zu den zugehörigen lokalen Variablen von OB 80.

Strukturelement TOP_SI	Datentyp	OB 80 - Zugehörige lokale Variable	Datentyp
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

Die folgende Tabelle zeigt die Zuweisung der Strukturelemente des Parameters START_UP_SI der Anweisung RD_SINFO zu den zugehörigen lokalen Variablen von OB 100.

Strukturelement START_UP_SI	Datentyp	OB 100 - Lokale Variable	Datentyp
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 (LED-Status lesen)

Tabelle 9-148 Anweisung LED

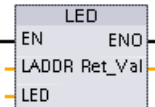
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := LED(laddr:=_word_in_, LED:=_uint_in_);</pre>	<p>Mit der Anweisung LED lesen Sie den Zustand der LEDs auf der CPU (Seite 1205) aus. Der angegebene LED-Zustand wird vom Ausgang RET_VAL zurückgegeben.</p>

Tabelle 9-149 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung		
LADDR	IN	HW_IO		
LED	IN	UInt		
		1	RUN/STOP	Farbe 1 = grün, Farbe 2 = gelb
		2	Fehler	Farbe 1 = rot
		3	Wartung	Farbe 1 = gelb
RET_VAL	OUT	Int		
		Zustand der LED		

¹ Für die Kennung der angeschlossenen CPU wählen Sie Local~Common aus der Klappliste des Parameters.

Tabelle 9-150 Zustand von RET_VAL

RET_VAL (W#16#...)	Beschreibung	
LED-Zustand 0 bis 9	0	LED ist nicht vorhanden
	1	Aus
	2	Farbe 1 ein (leuchtet ständig)
	3	Farbe 2 ein (leuchtet ständig)
	4	Farbe 1 blinkt mit 2 Hz
	5	Farbe 2 blinkt mit 2 Hz
	6	Farben 1 & 2 blinken abwechselnd mit 2 Hz
	9	Zustand der LED ist nicht verfügbar
8091	Von LADDR angegebenes Gerät ist nicht vorhanden	
8092	Von LADDR angegebenes Gerät unterstützt keine LEDs	
8093	LED-Kennung nicht definiert	
80Bx	Von LADDR angegebene CPU unterstützt die Anweisung LED nicht	

9.7.4 Get_IM_Data (Identifikations- und Wartungsdaten lesen)

Mit der Anweisung `Get_IM_Data` prüfen Sie die Identifikations- und Wartungsdaten (I&M) für das angegebene Modul oder Submodul.

Tabelle 9-151 Anweisung `Get_IM_Data`

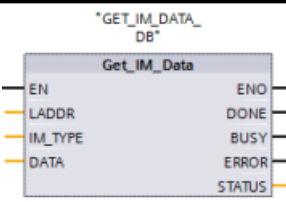
KOP/FUP	SCL	Beschreibung
	<pre>"GET_IM_DATA_DB" (LADDR:=16#0 , IM_TYPE:=0, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA:=_variant_inout_);</pre>	Mit der Anweisung <code>Get_IM_Data</code> prüfen Sie die Identifikations- und Wartungsdaten (I&M) für das angegebene Modul oder Submodul.

Tabelle 9-152 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
LADDR	Input	HW_EA
IM_TYPE	Input	UInt
RET_VAL	Output	Int
DATA	InOut	Variante

Beschreibung:

- Kennung der Baugruppe
- Nummer der Identifikations- und Wartungsdaten (I&M):
 - 0: I&M0 (MLFB, Seriennummer, Version und andere Informationen)
 - 1: I&M1 (Designatoren)
 - 2: I&M2 (Installationsdatum)
 - 3: I&M3 (Deskriptor)
 - 4: I&M4 (Signatur)
- Status (Bedingungscode)
- I&M-Daten (STRING oder ein Array aus BYTE); empfohlen wird die Verwendung des SDT "IM0_Data" für IM_TYPE = 0.

Über die Identifikations- und Wartungsdaten (I&M) können Sie die Systemkonfiguration prüfen, Hardwareänderungen erkennen oder Wartungsdaten anzeigen. Die Modulidentifikationsdaten (I-Daten) sind schreibgeschützt. Die Modulwartungsdaten (M-Daten) sind von Systeminformationen wie z. B. dem Installationsdatum abhängig. M-Daten werden während der Wartungsplanung erstellt und in das Modul geschrieben:

- Wenn es sich bei dem Datentyp des Parameters DATA um eine Zeichenkette handelt, wird die aktuelle Länge der Zeichenkette entsprechend der Länge der I&M-Daten festgelegt.
- Wenn es sich bei dem Datentyp des Parameters DATA um ein Array aus Byte oder Char handelt, werden die I&M-Daten als Folge von Bytes kopiert.
- Wenn es sich bei dem Datentyp des Parameters DATA um eine Struktur handelt, werden die I&M-Daten als Folge von Bytes kopiert.
- Wenn das jeweilige Array aus Byte/Char in DATA länger ist als die angeforderten I&M-Daten, wird der Bytewert 16#00 angehängt.
- Andere Datentypen werden nicht unterstützt, es wird der Fehler 8093 zurückgegeben.

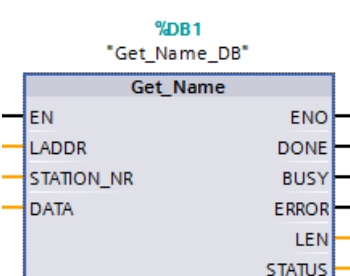
Tabelle 9-153 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0	Kein Fehler
8091	LADDR ist nicht vorhanden
8092	LADDR adressiert kein HW-Objekt, das I&M-Daten unterstützt
8093	Am Parameter DATA angegebener Datentyp wird nicht unterstützt
80B1	Die Anweisung DATA wird von der CPU für diesen Parameter LADDR nicht unterstützt
80B2	IM_TYPE wird von der CPU nicht unterstützt
8452	Die vollständigen I&M-Informationen passen nicht in die am Parameter DATA angegebene Variable. Es wird ein Teilergebnis bis zur Bytelänge der Variable zurückgegeben.

9.7.5 Get_Name (Namen eines PROFINET IO-Device lesen)

Die Anweisung "Get_Name" liest den Namen eines PROFINET IO-Device, PROFIBUS-Slaves oder AS-i-Slaves. Der Name wird in der Netzsicht und in den Eigenschaften des IO-Device angezeigt.

Tabelle 9-154 Anweisung Get_Name

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung "Get_Name" lesen Sie den Namen eines PROFINET IO-Device oder PROFIBUS-Slaves aus.</p>

- STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "Get_Name_DB" der Name des Instanz-DBs.

Sie wählen das IO-Device über die Hardwareerkennung des dezentralen IO-Systems (am Parameter LADDR) und die Gerätenummer des PROFINET IO-Device oder die PROFIBUS-Adresse des PROFIBUS-Slaves (am Parameter STATION_NR) aus.

Nach der Ausführung der Anweisung schreibt das Programm den Namen des IO-Device in den vom Parameter DATA adressierten Bereich.

Der Name, der gelesen wird, hängt vom Typ des IO-Device ab:

- DP-Slave oder IO-Device: Name des Kopfmoduls
- I-Slave oder I-Device: Name des Interfacemoduls
- HMI-Panel: Name der Schnittstelle
- PC-Station: Name des Interfacemoduls
- GSD-Geräte: Angezeigt wird der Name des Device Access Point (DAP) (Name der Schnittstelle oder des Headmoduls)

Die Anweisung schreibt die Länge des Namens in den Parameter LEN. Wenn der Name länger ist als der am Parameter DATA angegebene Bereich, schreibt das Programm nur den Abschnitt, der der maximalen Länge des adressierten Bereichs entspricht.

Die maximale Länge für einen Namen beträgt 128 Zeichen.

Hinweis

Auslesen des CPU-Namens (Version 1.1)

Wenn Sie an beiden Parametern LADDR und STATION_NR, den Wert 0 zuweisen, schreibt die Anweisung den Namen der CPU.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung Get_Name:

Parameter	Deklaration	Datentyp	Beschreibung
LADDR	IN	HW_IOSYSTEM	Hardwarekennung (HW-IOSystem) des dezentralen IO-Systems. Die Nummer wird aus den Systemkonstanten oder den Eigenschaften des IO-Systems abgerufen.
STATION_NR	IN	UInt	<ul style="list-style-type: none"> PROFINET IO-Device: Die Gerätenummer wird aus den Eigenschaften des IO-Device unter "Ethernet-Adressen" in die Netzsicht übernommen. PROFIBUS-Slave: Die PROFIBUS-Adresse wird aus den Eigenschaften des PROFIBUS-Slaves unter "PROFIBUS-Adresse" in die Netzsicht übernommen.
DATA	IN_OUT	Variant	Pointer auf den Bereich, in den der Name geschrieben wird.
DONE	OUT	Bool	Die Anweisung wird erfolgreich ausgeführt. Der Name des Moduls wird in den Bereich am Parameter DATA übertragen.
BUSY	OUT	Bool	Statusparameter: <ul style="list-style-type: none"> 0: Die Ausführung der Anweisung ist beendet. 1: Die Ausführung der Anweisung ist noch nicht beendet.
ERROR	OUT	Bool	Statusparameter: <ul style="list-style-type: none"> 0: Kein Fehler 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Der Parameter STATUS enthält ausführliche Informationen.
LEN	OUT	DInt	Länge des Namens des IO-Device (Anzahl Zeichen).
STATUS	OUT	Word	Statusparameter: Der Parameter wird nur für die Dauer eines Aufrufs gesetzt. Um den Status anzuzeigen, sollten Sie deshalb den Parameter STATUS in einen freien Datenbereich kopieren.

Parameter STATUS

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler
7000	Kein Auftrag in Bearbeitung.
7001	Erster Aufruf der asynchronen Anweisung Get_Name. Die Ausführung der Anweisung ist noch nicht beendet (BUSY = 1, DONE = 0).
7002	Zusätzlicher Aufruf der asynchronen Anweisung Get_Name. Die Ausführung der Anweisung ist noch nicht beendet (BUSY = 1, DONE = 0).
8090	Die am Parameter LADDR angegebene Hardwarekennung ist im Projekt nicht vorhanden.
8092	Der Wert am Parameter LADDR adressiert kein PROFINET IO-System.
8093	Die Anweisung unterstützt den Datentyp am Parameter DATA nicht.
8095	Die Gerätenummer (Parameter STATION_NR) ist im ausgewählten PROFINET IO-System nicht vorhanden oder adressiert kein IO-Device.
80B1	Die verwendete CPU unterstützt die Anweisung nicht.

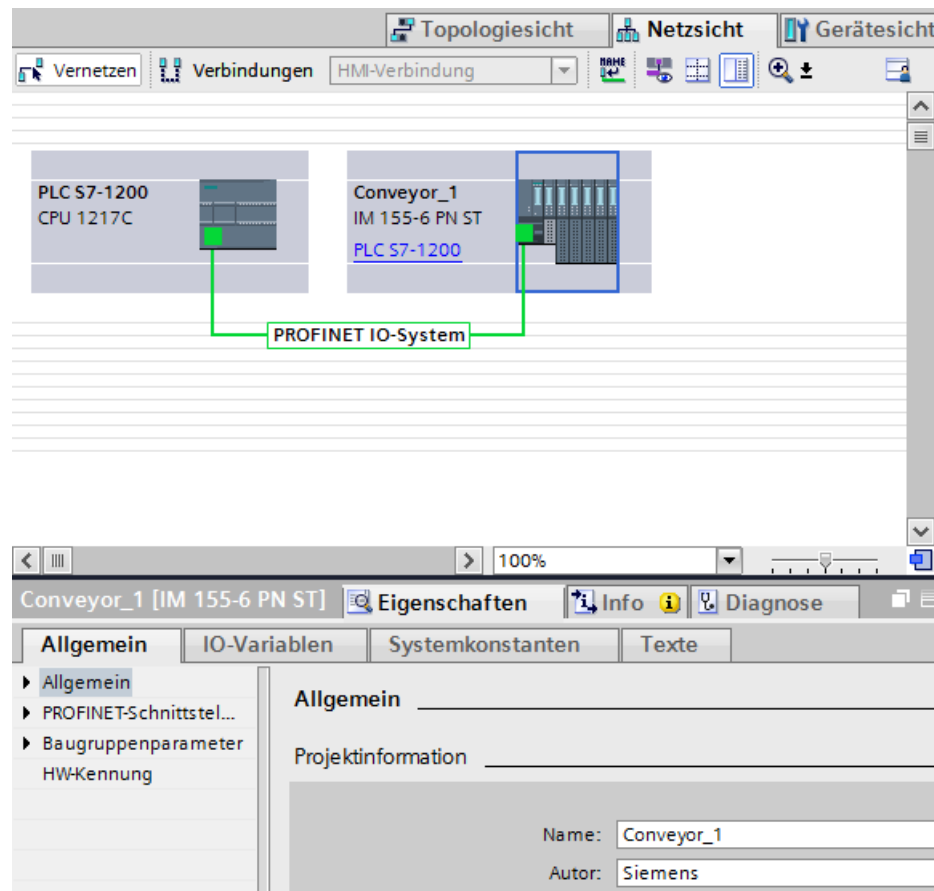
Fehlercode* (W#16#...)	Bedeutung
80C3	Temporärer Ressourcenfehler: Die CPU verarbeitet gegenwärtig die maximal mögliche Anzahl gleichzeitiger Bausteinaufrufe. Get_Name kann erst ausgeführt werden, wenn mindestens einer der Bausteinaufrufe abgearbeitet ist.
8852	Der am Parameter DATA angegebene Bereich ist zu kurz für den vollständigen Namen des IO-Device. Der Name kann bis zu maximal möglichen Länge geschrieben werden. Um den vollständigen Namen auszulesen, verwenden Sie einen längeren Datenbereich am Parameter DATA. Der Bereich muss mindestens so viele Zeichen haben, wie vom Parameter LEN vorgegeben.
* Die Fehlercodes können im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt werden.	

Beispiel

Das folgende Beispiel zeigt, wie Sie den Stationsnamen eines ET 200SP PROFINET IO-Device lesen können:

1. ET200SP konfigurieren:

- Erstellen Sie das ET 200SP mit dem Stationsnamen "Conveyor_1" in der Netzsicht und weisen Sie es dem gleichen PROFINET IO-System zu wie die CPU.
- Weisen Sie die CPU als den IO-Controller für das ET 200SP zu.
- Verwenden Sie die Standardgerätenummer "1" in den Eigenschaften unter "Ethernet-Adressen".



2. Parameter für die Anweisung Get_Name zuweisen:

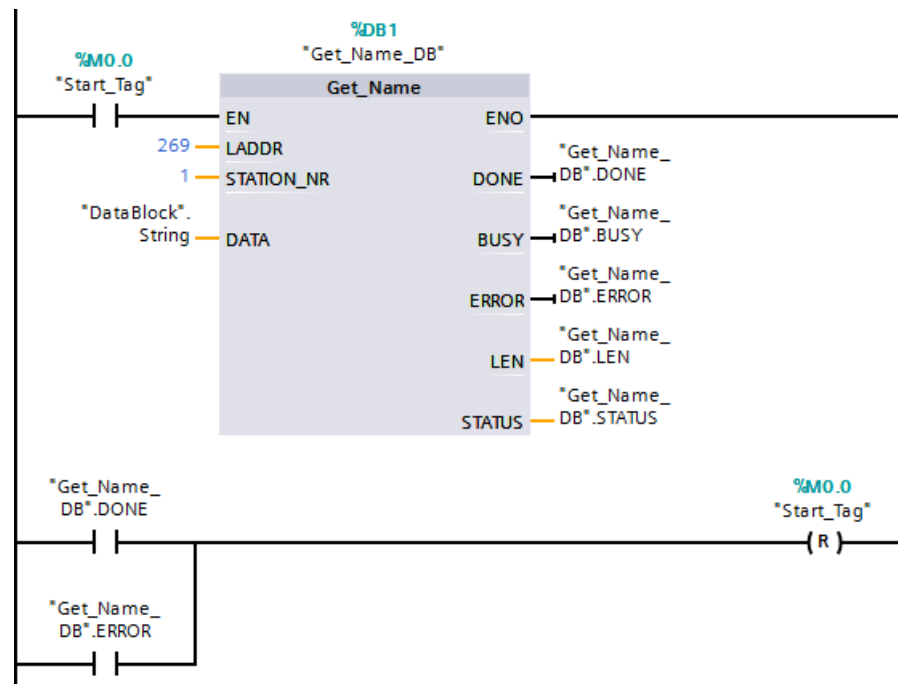
- Geben Sie die Hardware-ID des IO-Systems am Parameter LADDR ein. In diesem Beispiel lautet die Hardware-ID "269". Sie finden die Hardware-ID an folgender Stelle: PLC-Variablen > Alle Variablen anzeigen > Register "Systemkonstanten" > Lokales-PROFINET_IO-System.
- Geben Sie die Gerätenummer des ET 200SP am Parameter STATION_NR ein. In diesem Beispiel ist die Gerätenummer "1".

- Verbinden Sie eine Variable mit dem Datentyp STRING eines Datenbausteins am Parameter DATA.

Hinweis

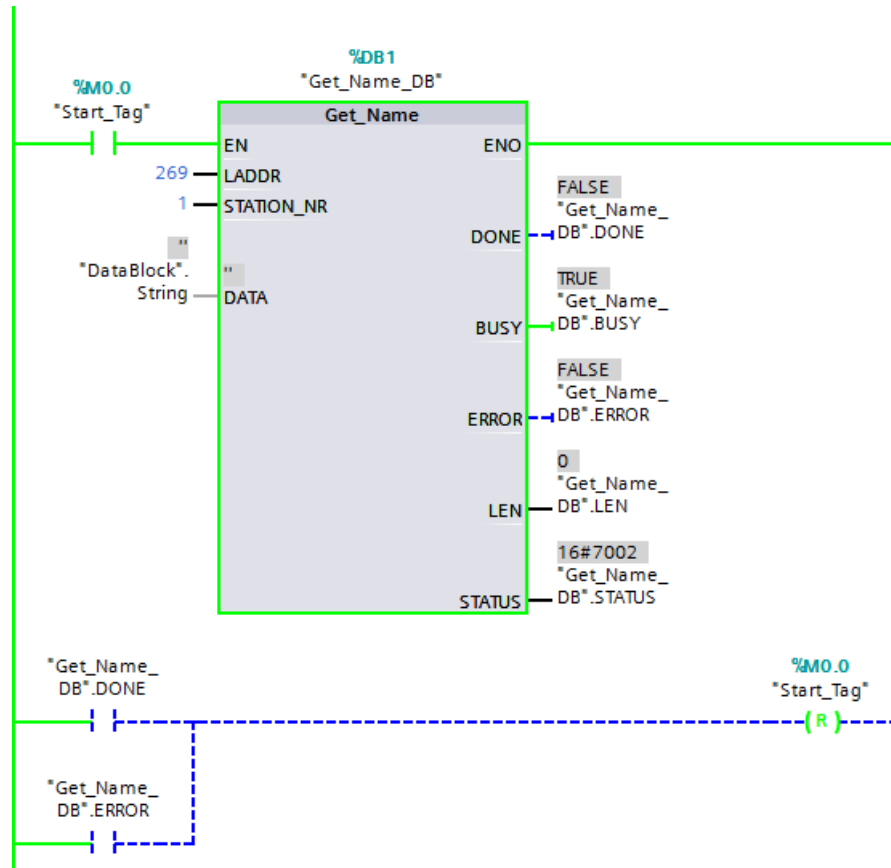
Wenn Sie bei der Konfiguration der Variablen am Parameter DATA für Ihre Auswahl die Klappliste verwenden, wählen Sie den DB (im Beispiel "Datablock") und die Variable (im Beispiel "String[]") aus. Zum Lesen des gesamten Datentyps String müssen Sie die Klammern löschen, um folgendes Endergebnis zu erzielen: "Datablock".String.

- Definieren Sie die PLC-Variablen (Speicherbereich, Merker) für die Ausgangsparameter der Anweisung.



3. Anweisung Get_Name ausführen:

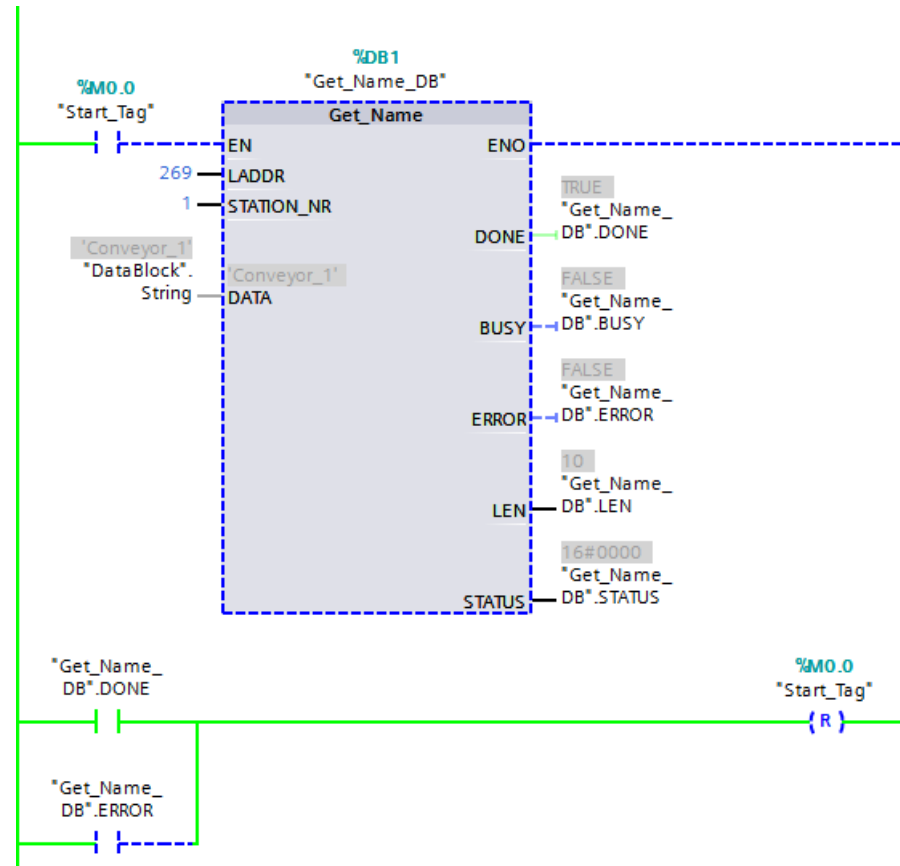
- Bei der Ausführung der Anweisung kann der Ausgangsparameter BUSY auf 1 gesetzt werden, woraufhin der Parameter DONE auf 0 gesetzt wird.
- Informationen zum Fehlercode werden am Ausgangsparameter STATUS angezeigt.



4. Ausführung der Anweisung Get_Name beenden:

- Nach der Ausführung der Anweisung schreibt das Programm "Conveyor_1", den Stationsnamen des ET 200SP, in den Datenbaustein am Parameter DATA.

- Das Programm schreibt "10", die Anzahl der Zeichen im Stationsnamen, in den Parameter LEN.



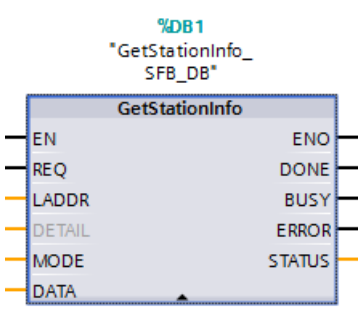
9.7.6 GetStationInfo (IP- oder MAC-Adresse eines PROFINET IO-Device lesen)

Die Anweisung "GetStationInfo" liest die IP- oder MAC-Adresse eines PROFINET IO-Device im lokalen IO-System oder eines PROFINET IO-Device in einem untergeordneten IO-System (angeschlossen über CP/CM-Module).

Hinweis

Die Anweisung GetStationInfo können Sie nur bei PROFINET IO-Devices verwenden. Sie können die Anweisung nicht mit PROFIBUS DP-Slaves verwenden.

Tabelle 9-155 Anweisung GetStationInfo

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung GetStationInfo lesen Sie die IP- oder MAC-Adresse eines PROFINET IO-Device. Mit der Anweisung können Sie auch die IP- oder MAC-Adresse eines IO-Device in einem untergeordneten IO-System (angeschlossen über CP/CM-Module) lesen.</p>

- ¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- ² Im SCL-Beispiel ist "GetStationInfo_SFB_DB" der Name des Instanz-DBs.

Sie adressieren das IO-Device über die Hardwarekennung der Station am Parameter LADDR. Sie finden die Hardware-ID an folgender Stelle:
 PLC-Variablen > Alle Variablen anzeigen > Register "Systemkonstanten". Suchen Sie in der Spalte "Name" nach dem IO-Device und in der Spalte "Datentyp" nach "Hw_Device".

Über den Parameter MODE wählen Sie die zu lesenden Informationen aus.

Am Parameter DATA weisen Sie den Datenbereich zu, in den die Anweisung die gelesenen Adressdaten schreibt. Zum Speichern der IP-Adresse verwenden Sie die Struktur "IF_CONF_v4". Zum Speichern der MAC-Adresse verwenden Sie die Struktur "IF_CONF_MAC".

Das Lesen der Adressdaten aktivieren Sie über den Steuerparameter REQ. Hierfür muss das IO-Device zugänglich sein.

Die Anweisung zeigt den Ausführungszustand des Leseauftrags über die Ausgangsparameter BUSY, DONE und ERROR und den Ausgangsparameter STATUS an.

Hinweis

Adressieren Sie das IO-Device nur über die Hardwarekennung der Station.

Die Station, das IO-Device und die PROFINET-Schnittstelle haben jeweils eine eigene Hardwarekennung. Verwenden Sie für die Anweisung GetStationInfo nur die Hardwarekennung der Station.

Wird beispielsweise eine PROFINET-Schnittstelle über den Parameter LADDR adressiert, werden die Adressdaten nicht gelesen und die CPU erzeugt den Fehlercode "8092".

Um die Adressdaten einer integrierten PROFINET-Schnittstelle oder eines CM/CP-Moduls in der zentralen Konfiguration zu lesen, verwenden Sie die Anweisung "RDREC".

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GetStationInfo:

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	Bool	Steuerparameter REQUEST Aktiviert das Lesen der Informationen mit REQ = 1.
LADDR	IN	HW_DEVICE	Hardwareerkennung der Station des IO-Device Die Nummer wird den Eigenschaften der Station in der Netzsicht oder dem Register "Systemkonstanten" in der Standardvariablen-tabelle entnommen.
DETAIL	IN	HW_SUBMODULE	Dieser Parameter DETAIL wird nicht verwendet. Verschalten Sie den Parameter nicht.
MODE	IN	UNIT	Auswahl von zu lesenden Adressdaten: <ul style="list-style-type: none"> MODE = 1: Adressparameter nach IPv4 (S7-1200 CPUs der Firmwareversion V4.2) MODE = 2: MAC-Adresse (S7-1200 CPUs der Firmwareversion V4.2)
DATA	IN_OUT	Variant	Pointer auf den Bereich, in den das Programm die Adressdaten des IO-Device schreibt. Verwenden Sie die Struktur "IF_CONF_v4" bei MODE = 1 und die Struktur "IF_CONF_MAC" bei MODE = 2.
DONE	OUT	Bool	Das Programm hat die Anweisung erfolgreich ausgeführt. Das Programm hat die Adressdaten an den Parameter DATA übertragen.
BUSY	OUT	Bool	Parameter STATUS: <ul style="list-style-type: none"> 0: Die Ausführung der Anweisung ist beendet. 1: Die Ausführung der Anweisung ist noch nicht beendet.
ERROR	OUT	Bool	Parameter STATUS: <ul style="list-style-type: none"> 0: Kein Fehler. 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen werden über den Parameter STATUS ausgegeben.
STATUS	OUT	Word	Parameter STATUS: Der Parameter wird nur für die Dauer eines Aufrufs gesetzt. Um den Status anzuzeigen, sollten Sie deshalb den Parameter STATUS in einen freien Datenbereich kopieren.

Parameter DATA

- Verwenden Sie die Struktur "IF_CONF_v4" am Parameter DATA, um den Adressparameter nach IPv4 zu speichern:

Byte	Parameter	Datentyp	Startwert	Beschreibung
0 ... 1	Id	UINT	30	ID der Struktur "IF_CONF_v4"
2 ... 3	Length	UNIT	18	Länge der gelesenen Daten in BYTE
4 ... 5	Mode	UNIT	0	Nicht relevant bei der Anweisung "GetStationInfo" (bleibt 0)

Byte	Parameter	Datentyp	Startwert	Beschreibung
6 ... 9	InterfaceAddress	ARRAY [1..4] of BYTE	-	IP-Adresse des IO-Device im Format IP_V4 (Beispiel: 192.168.3.10): <ul style="list-style-type: none"> • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 10
10 ... 13	SubnetMask	ARRAY [1..4] of BYTE	-	Subnetzmaske des IO-Device im Format IP_V4 (Beispiel: 255.255.255.0): <ul style="list-style-type: none"> • addr[1] = 255 • addr[2] = 255 • addr[3] = 255 • addr[4] = 0
14 ... 17	DefaultRouter	ARRAY [1..4] of BYTE	-	IP-Adresse des Routers im Format IP_V4 (Beispiel: 192.168.3.1): <ul style="list-style-type: none"> • addr[1] = 192 • addr[2] = 168 • addr[3] = 3 • addr[4] = 1

- Zum Speichern der MAC-Adresse verwenden Sie die Struktur "IF_CONF_MAC" am Parameter DATA.

Byte	Parameter	Datentyp	Startwert	Beschreibung
0 ... 1	Id	UINT	3	ID der Struktur "IF_CONF_MAC"
2 ... 3	Length	UNIT	12	Länge der gelesenen Daten in BYTE
4 ... 5	Mode	UNIT	0	Nicht relevant bei der Anweisung "GetStationInfo" (bleibt 0)
6 ... 11	MACAddress	ARRAY [1..6] of BYTE	-	MAC-Adresse des IO-Device (Beispiel: 08-00-06-12-34-56): <ul style="list-style-type: none"> • Mac[1] = 8 • Mac[2] = 0 • Mac[3] = 6 • Mac[4] = 12 • Mac[5] = 34 • Mac[6] = 56

Parameter STATUS

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler
7000	Kein Auftrag in Bearbeitung.
7001	Erster Aufruf der asynchronen Anweisung GetStationInfo. Die Ausführung der Anweisung ist noch nicht beendet (BUSY = 1, DONE = 0).

Fehlercode* (W#16#...)	Bedeutung
7002	Zusätzlicher Aufruf der asynchronen Anweisung GetStationInfo. Die Ausführung der Anweisung ist noch nicht beendet (BUSY = 1, DONE = 0).
8080	Der Wert am Parameter MODE wird nicht unterstützt.
8090	Die am Parameter LADDR angegebene Hardwarekennung ist nicht konfiguriert.
8092	Der Parameter LADDR adressiert kein PROFINET IO-Device.
8093	Ungültiger Datentyp am Parameter DATA.
80A0	Angeforderte Informationen werden nicht gelesen.
80C0	Adressiertes IO-Device ist nicht erreichbar.
80C3	Die maximale Anzahl gleichzeitiger Aufrufe der Anweisung GetStationInfo (10 Instanzen) ist erreicht.
* Die Fehlercodes werden im Programmmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt.	

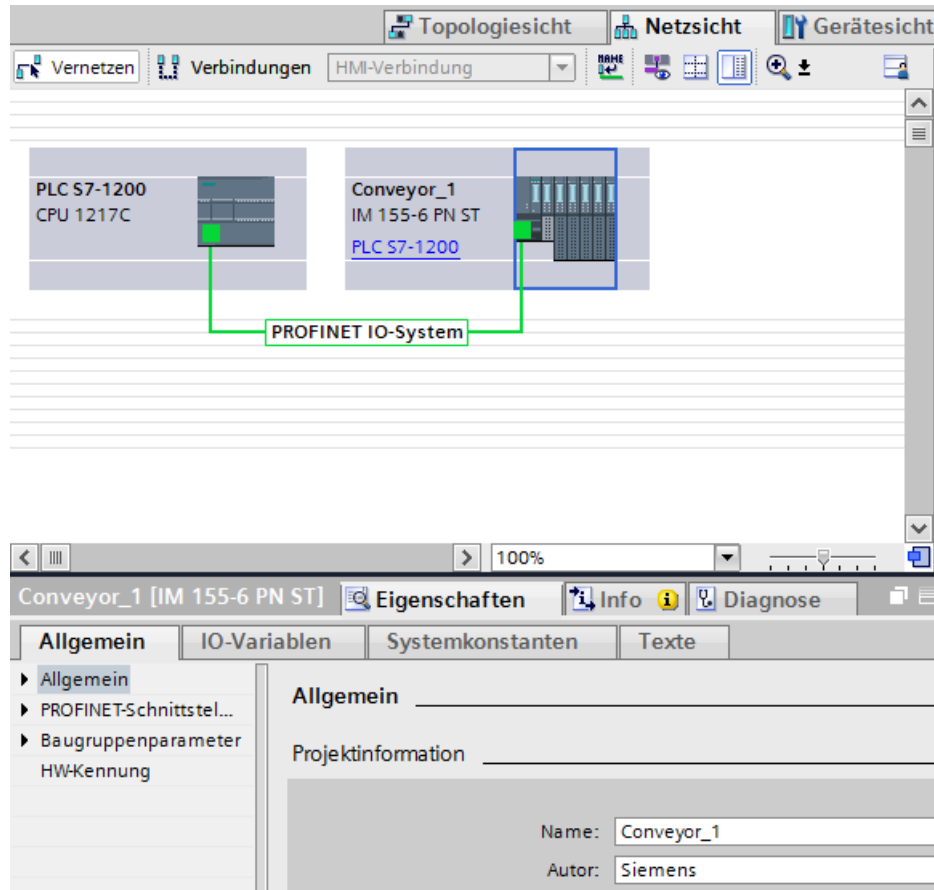
Beispiel

Im folgenden Beispiel lesen Sie mit der Anweisung GetStationInfo die IP-Adressdaten eines IO-Device und schreiben die Informationen in einen Datenbaustein. Die IP-Adressdaten umfassen die IP-Adresse, die Subnetzmaske und (sofern verwendet) die Adressdaten des Routers.

Der IO-Controller führt die Anweisung GetStationInfo aus, und die Anweisung liest die IP-Adressdaten eines untergeordneten IO-Device (in diesem Beispiel ein ET200SP):

1. ET200SP konfigurieren:

- Erstellen Sie das ET 200SP mit dem Stationsnamen "Conveyor_1" in der Netzsicht und weisen Sie es dem gleichen PROFINET IO-System zu wie die CPU.
- Weisen Sie die CPU als den IO-Controller für das ET 200SP zu.



2. Parameter für die Anweisung GetStationInfo zuweisen:

- Legen Sie fünf Variablen und eine Struktur mit dem Datentyp IF_CONF_v4 in einem globalen Datenbaustein zum Speichern der IP-Adressdaten an. Geben Sie der Struktur einen beliebigen Namen. (Im Beispiel lautet der Name der Struktur "IP_Address".)

GetStationInfo_Global_DB			
	Name	Datentyp	Startwert
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

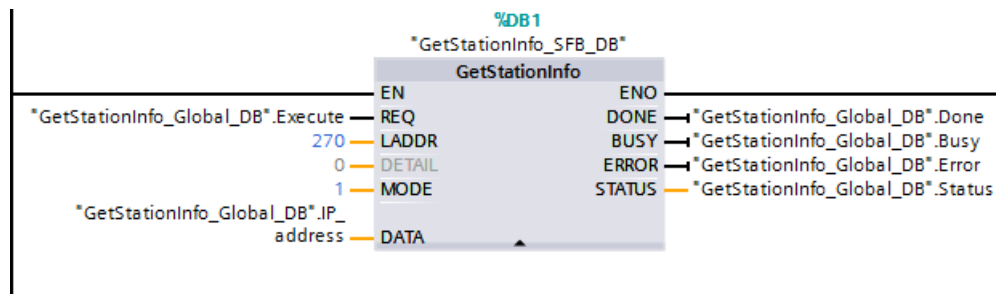
3. Parameter für die Anweisung GetStationInfo zuweisen:

- Geben Sie die Hardware-ID des IO-Device am Parameter LADDR ein. Die Hardwareerkennung identifiziert das Produkt eindeutig. In diesem Beispiel lautet die Hardware-ID "270". Sie finden die Hardware-ID an folgender Stelle: PLC-Variablen > Alle Variablen anzeigen > Register "Systemkonstanten". Suchen Sie in der Spalte "Name" nach dem IO-Device und in der Spalte "Datentyp" nach "Hw_Device". Der zugewiesene Wert ist die Hardware-ID, die Sie am Parameter LADDR eingeben.
- Wählen Sie für den Parameter MODE den Wert 1 (Adressparameter nach IPv4 lesen).
- Verbinden Sie die Struktur IF_CONF_v4 am Parameter DATA.

Hinweis

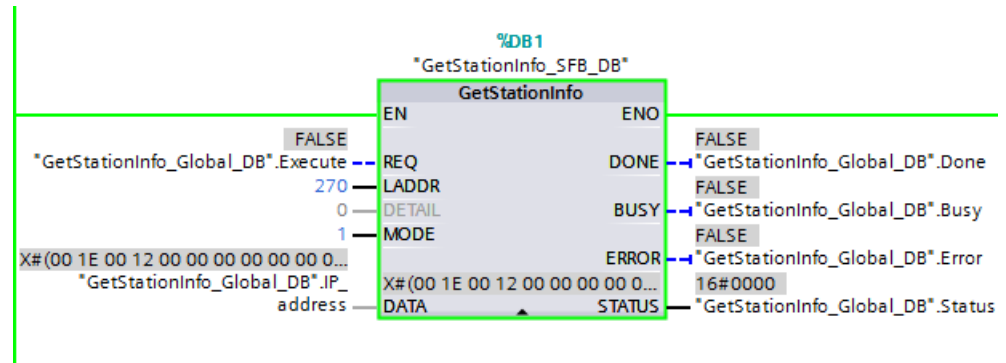
Wenn Sie bei der Konfiguration der Variablen am Parameter DATA für Ihre Auswahl die Klappliste verwenden, wählen Sie den DB (im Beispiel "GetStationInfo_Global_DB") und die Variable (im Beispiel "IP address") aus. Um den gesamten Datentyp IF_CONF_v4 zu lesen, müssen Sie den Punkt löschen, der auf "IP address" folgt, um folgendes Ergebnis zu erzielen: "GetStationInfo_Global_DB".IP address

- Definieren Sie die PLC-Variablen Ihres globalen DB für die Ausgangsparameter der Anweisung.



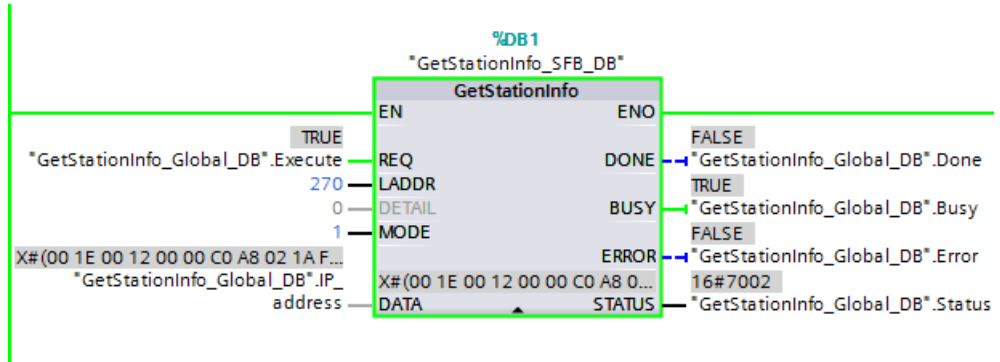
4. Anweisung GetStationInfo ausführen:

- Wenn der Eingang REQ = 1 ist (FALSE), zeigt die Anweisung am Eingangs-/Ausgangsparameter DATA keine IP-Adressdaten bzw. am Ausgangsparameter STATUS Informationen zum Fehlercode an.



5. Ausführung der Anweisung GetStationInfo beenden:

- Wenn der Eingang REQ = 1 ist (TRUE), führt das Programm die Anweisung aus und schreibt die IP-Adresse in den Datenbaustein. Das Programm schreibt die IP-Adresse, "C0 A8 02 1A" (Dezimalwert von "192.168.2.26"), in den Eingangs-/Ausgangsparameter DATA.



GetStationInfo_Global_DB				
	Name	Datentyp	Startwert	Beobachtungswert
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 Anweisung DeviceStates

Mit der Anweisung DeviceStates können Sie die Zustände aller dezentralen E/A-Slavegeräte, die an einen spezifischen dezentralen E/A-Master angeschlossen sind, ausgeben.

Tabelle 9-156 Anweisung DeviceStates

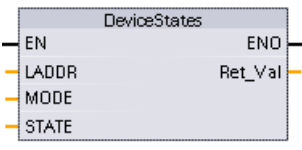
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := DeviceStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>DeviceStates ruft die Betriebszustände von E/A-Geräten in einem E/A-Untersystem ab. Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Geräts in einer Bitliste (für die zugewiesenen Parameter LADDR und MODE). Diese Informationen entsprechen dem Gerätestatus in der Diagnoseansicht von STEP 7.</p> <p>Der Eingang LADDR der Anweisung DeviceStates verwendet die Hardwarekennung einer dezentralen E/A-Schnittstelle. Im TIA Portal finden Sie die Hardwarekennungen für einen PLC über die Datentypen "HW_IOSYSTEM" im Register "Systemkonstanten" in der PLC-Variablentabelle.</p>

Tabelle 9-157 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
LADDR	IN	HW_IOSYSTEM Logische Adresse: (Kennung des E/A-Systems)
MODE	IN	UInt Unterstützt fünf Betriebsarten. Der Eingang MODE legt fest, welche Daten an der für STATE-Informationen angegebenen Stelle ausgegeben werden. Folgende Betriebsarten sind möglich: <ul style="list-style-type: none"> • 1: Gerätekonfiguration aktiv • 2: Gerät defekt • 3: Gerät deaktiviert • 4: Gerät vorhanden • 5: Problem im Gerät
RET_VAL	OUT	Int Ausführungsbedingung
STATE ¹	InOut	Variant Puffer, der den Fehlerzustand der einzelnen Geräte empfängt: Als Datentyp für den Parameter STATE können Sie einen beliebigen Bittyp (Bool, Byte, Word oder DWord) oder ein Array eines Bittyps auswählen. <ul style="list-style-type: none"> • Bit 0 des ersten Bytes der ausgegebenen STATE-Daten ist ein Zusammenfassungsbit. Ist es auf WAHR gesetzt, zeigt es an, dass andere Daten verfügbar sind. • Die vom Parameter STATE ausgegebenen Daten zeigen eine 1-zu-1-Übereinstimmung zwischen einer Bitposition und einer dezentralen E/A-Adresse. Diese Geräteadressierung ist WAHR für PROFIBUS und PROFINET. Beispiel: Bit 4 im ersten Byte stimmt mit PROFIBUS-Adresse 4 oder PROFINET-Gerätenummer 4 überein.

¹ Bei PROFIBUS DP beträgt die Länge der Zustandsinformationen 128 Bits. Bei PROFINET I/O beträgt die Länge 1024 Bit.

Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Geräts als Bitliste (für die zugewiesenen LADDR und MODE).

Tabelle 9-158 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0	Kein Fehler
8091	LADDR ist nicht vorhanden.
8092	LADDR spricht kein E/A-System an.
8093	Ungültiger Datentyp für den Parameter STATE: Gültige Datentypen sind (Bool, Byte, Word oder Dword) oder ein Array aus (Bool, Byte, Word oder Dword)
80Bx	Die Anweisung DeviceStates wird von der CPU für diesen Parameter LADDR nicht unterstützt.
8452	Die vollständigen Zustandsdaten sind für den zugewiesenen Parameter STATE zu groß. Der Puffer STATE enthält ein Teilergebnis.

9.7.7.1 Beispiele für die Konfiguration von DeviceStates

Beispiel in PROFIBUS

Das Beispiel in PROFIBUS besteht aus den folgenden Komponenten:

- 16 PROFIBUS-Geräte mit den Namen "DPSlave_10" bis "DPSlave_25"
- Die 16 PROFIBUS-Geräte verwenden jeweils eine der PROFIBUS-Adressen 10 bis 25.
- Für jedes Slavegerät sind mehrere E/A-Module konfiguriert.
- Es werden die ersten vier Bytes der ausgegebenen STATE-Parameterinformationen angezeigt.

MODE	Beispiel 1: Normalbetrieb ohne Fehler	Beispiel 2: PROFIBUS Slavegerät DPSlave_12 mit einem gezogenen Modul	Beispiel 3: PROFIBUS Slavege- rät DPSlave_12 ge- trennt
1: Gerätekonfiguration aktiv	0x01FC_FF03	0x01FC_FF03	0x01FC_FF03
2: Gerät defekt	0x0000_0000	0x0110_0000	0x0110_0000
3: Gerät deaktiviert	0x0000_0000	0x0000_0000	0x0000_0000
4: Gerät vorhanden	0x01FC_FF03	0x01FC_FF03	0x01EC_FF03
5: Problem im Gerät	0x0000_0000	0x0110_0000	0x0110_0000

Die folgenden vier Tabellen zeigen eine binäre Aufschlüsselung der vier analysierten Datenbytes:

Tabelle 9-159 Beispiel 1: Keine Fehler: Für MODE 1 (Gerätekonfiguration aktiv) wird der Wert 0x01FC_FF03 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0xFC	Bit 15 1111-1100 Bit 8	

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 3 0xFF	Bit 23 1111-1111 Bit 16	
Byte 4 0x03	Bit 31 0000-0011 Bit 24	

Die Geräte sind an den Adressen 10 (Bit 10) bis 25 (Bit 25) konfiguriert.

An den Adressen 1 bis 9 sind keine Geräte konfiguriert.

MODE Daten 4 (Gerät vorhanden) stimmen mit MODE 1 (Gerätekonfiguration aktiv) überein, also entsprechen die konfigurierten Geräte den vorhandenen Geräten.

Tabelle 9-160 Beispiel 2: Ein Modul wurde aus dem PROFIBUS-Slavegerät "DPSlave_12" gezogen. Für MODE 2 (Gerät defekt) wird der Wert 0x0110_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Gerät 12 (Bit 12) ist als defekt gekennzeichnet.

MODE 5 (Problem im Gerät) gibt die gleichen Informationen wie MODE 2 (Gerät defekt) aus.

Tabelle 9-161 Beispiel 2 (Fortsetzung): Ein Modul wurde aus dem PROFIBUS-Slavegerät "DPSlave_12" gezogen. Für MODE 4 (Gerät vorhanden) wird der Wert 0x01FC_FF03 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Obwohl Gerät 12 (Bit 12) wie in MODE 2 (Gerät defekt) oben gezeigt einen Fehler aufweist, funktioniert das Gerät trotzdem noch im Netzwerk, weshalb MODE 4 (Gerät vorhanden) das Gerät als "vorhandenes Gerät" anzeigt.

Tabelle 9-162 Beispiel 3: PROFIBUS-Slavegerät "DPSlave_12" ist vom PROFIBUS-Netzwerk getrennt (Kabel getrennt oder Spannungsverlust). "DPSlave_12" wird weiterhin als defektes Gerät sowie als Gerät mit Fehler erkannt. Der Unterschied ist, dass "DPSlave_12" nicht mehr als vorhandenes Gerät erkannt wird. Für MODE 4 (Gerät vorhanden) wird der Wert 0x01EC_FF03 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0xEC	Bit 15 1110-1100 Bit 8	
Byte 3 0xFF	Bit 23 1111-1111 Bit 16	
Byte 4 0x03	Bit 31 0000-0011 Bit 24	

Gerät 12 (Bit 12) ist als nicht vorhanden gekennzeichnet. Mit dieser Ausnahme werden die übrigen Geräte 10 bis 25 weiterhin als vorhanden gemeldet.

Beispiel in PROFINET

Das Beispiel in PROFINET besteht aus den folgenden Komponenten:

- 16 PROFINET-Slavegeräte mit den Namen "et200s_1" bis "et200s_16".
- Die 16 PROFINET-Geräte verwenden jeweils eine der PROFINET-Gerätenummern 1 bis 16.
- Für jedes Slavegerät sind mehrere E/A-Module konfiguriert.
- Es werden die ersten vier Bytes der ausgegebenen STATE-Parameterinformationen angezeigt.

MODE	Beispiel 1: Normalbetrieb ohne Fehler	Beispiel 2: PROFINET Gezogenes Modul an Slave et200s_1	Beispiel 3: PROFINET Slave et200s_1 getrennt
1: Gerätekonfiguration aktiv	0xFFFF_0100	0xFFFF_0100	0xFFFF_0100
2 - Gerät defekt	0x0000_0000	0x0300_0000	0x0300_0000
3 - Gerät deaktiviert	0x0000_0000	0x0000_0000	0x0000_0000
4 - Gerät vorhanden	0xFFFF_0100	0xFFFF_0100	0xFDFD_0100
5 - Problem im Gerät	0x0000_0000	0x0300_0000	0x0300_0000

Die folgenden vier Tabellen zeigen eine binäre Aufschlüsselung der vier analysierten Datenbytes:

Tabelle 9-163 Beispiel 1: Keine Fehler: Für MODE 1 (Gerätekonfiguration aktiv) wird der Wert 0xFFFF_0100 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Die Geräte sind an den Adressen 1 (Bit 1) bis 16 (Bit 16) konfiguriert. An den Adressen 1 bis 9 sind keine Geräte konfiguriert. MODE Daten 4 (Gerät vorhanden) stimmen mit MODE 1 (Gerätekonfiguration aktiv) überein, also entsprechen die konfigurierten Geräte den vorhandenen Geräten.

Tabelle 9-164 Beispiel 2: Ein Modul wurde aus dem PROFINET-Slavegerät "et200s_1" gezogen. Für MODE 2 (Gerät defekt) wird der Wert 0x0300_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x03	Bit 7 0000-0011 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Gerät 1 (Bit 1) ist als defekt gekennzeichnet. Da das Gerät weiterhin vorhanden ist, zeigt MODE 4 (Gerät vorhanden) die gleichen Daten wie im Normalbetrieb an. MODE 5 (Problem im Gerät) gibt die gleichen Informationen wie MODE 2 (Gerät defekt) aus.

Tabelle 9-165 Beispiel 2 (Fortsetzung): Ein Modul wurde aus dem PROFIBUS-Slavegerät "et200s_1" gezogen. Für MODE 4 (Gerät vorhanden) wird der Wert 0xFFFF_0100 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Obwohl Gerät 1 (Bit 1) wie in MODE 2 (Gerät defekt) oben gezeigt einen Fehler aufweist, funktioniert das Gerät trotzdem noch im Netzwerk, weshalb MODE 4 (Gerät vorhanden) das Gerät als "vorhandenes Gerät" anzeigt.

Tabelle 9-166 Beispiel 3: PROFINET-Slavegerät "et200s_1" ist vom PROFINET-Netzwerk getrennt (Kabel getrennt oder Spannungsverlust). Für MODE 4 (Gerät vorhanden) wird der Wert 0xFDFF_0100 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFD	Bit 7 1111-1101 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Gerät 1 (Bit 1) ist nicht vorhanden. Geräte 2 (Bit 2) bis 16 (Bit 16) sind vorhanden.

9.7.8 Anweisung ModuleStates

Mit der Anweisung ModuleStates können Sie den Zustand aller Module in einer PROFIBUS- oder PROFINET-Station ausgeben.

Tabelle 9-167 Anweisung ModuleStates

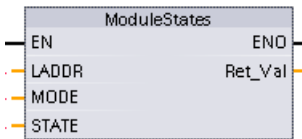
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := ModuleStates(laddr:=_word_in_, mode:=_uint_in_, state:=_variant_inout);</pre>	<p>ModuleStates ruft die Betriebszustände von E/A-Modulen ab. Nach der Ausführung enthält der Parameter STATE den Fehlerzustand jedes einzelnen E/A-Moduls in einer Bitliste (für die zugewiesenen Parameter LADDR und MODE). Diese Informationen entsprechen dem Modulstatus in der Diagnoseansicht von STEP 7.</p> <p>Der Eingang LADDR der Anweisung ModuleStates verwendet die Hardwarekennung einer dezentralen E/A-Station und nicht des Kopfmoduls selbst. Die Hardwarekennung können Sie ermitteln, indem Sie die gesamte Station in der Netzsicht selektieren und dann in den Eigenschaften im Bereich "Hardwarekennung" nachsehen. Sie finden Sie außerdem über die Datentypen "Hw_Device" und "Hw_DpSlave" im Register "Systemkonstanten" in der PLC-Variablen-tabelle.</p>

Tabelle 9-168 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
LADDR	IN	HW_DEVICE	Logische Adresse (Kennung der E/A-Module)
MODE	IN	UInt	<p>Unterstützt fünf Betriebsarten. Der Eingang MODE legt fest, welche Daten an der für STATE-Informationen angegebenen Stelle ausgegeben werden. Folgende Betriebsarten sind möglich:</p> <ul style="list-style-type: none"> • 1: Modulkonfiguration aktiv • 2: Modul defekt • 3: Modul deaktiviert • 4: Modul vorhanden • 5: Problem im Modul

Parameter und Datentyp		Datentyp	Beschreibung
RET_VAL	OUT	Int	Status (Bedingungscode)
STATE ¹	InOut	Variant	<p>Puffer, der den Fehlerzustand der einzelnen Module empfängt: Als Datentyp für den Parameter STATE können Sie einen beliebigen Bittyp (Bool, Byte, Word oder DWord) oder ein Array eines Bittyps auswählen.</p> <ul style="list-style-type: none"> • Bit 0 des ersten Bytes der ausgegebenen STATE-Daten ist ein Zusammenfassungsbit. Ist es auf WAHR gesetzt, zeigt es an, dass andere Daten verfügbar sind. • Die vom Parameter STATE ausgegebenen Daten zeigen eine 1-zu-1-Übereinstimmung zwischen einer Bitposition und einer Modulposition. Diese Steckplatzadressierung ist WAHR für PROFIBUS und PROFINET. Beispiel: Bei einem ET 200SP mit einem Kopfmodul, Powermodul und einem Paar E/A-Module stimmt Bit 1 im ersten Byte mit dem Kopfmodul überein, Bit 2 mit dem Powermodul und die Bits 3 und 4 mit den E/A-Modulen.

¹ Es können maximal 128 Bit zugewiesen werden. Die Anzahl der erforderlichen Bits ist von der Nutzung Ihres E/A-Moduls abhängig.

Tabelle 9-169 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0	Kein Fehler
8091	Von LADDR angegebene Modul ist nicht vorhanden.
8092	Das von LADDR angegebene Modul spricht kein E/A-Gerät an.
8093	Ungültiger Datentyp für den Parameter STATE: Gültige Datentypen sind (Bool, Byte, Word oder Dword) oder ein Array aus (Bool, Byte, Word oder Dword).
80Bx	Die Anweisung ModuleStates wird von dieser CPU für diesen Parameter LADDR nicht unterstützt.
8452	Die vollständigen Zustandsdaten sind für den zugewiesenen Parameter STATE zu groß. Der Puffer STATE enthält ein Teilergebnis.

9.7.8.1 Beispiele für die Konfiguration von ModuleStates

Beispiel in PROFIBUS

Das Beispiel in PROFIBUS besteht aus den folgenden Komponenten:

- 16 PROFIBUS-Geräte mit den Namen "DPSlave_10" bis "DPSlave_25"
- Die 16 PROFIBUS-Geräte verwenden jeweils eine der PROFIBUS-Adressen 10 bis 25.
- Für jedes Slavegerät sind mehrere E/A-Module konfiguriert.
- Im Beispiel wird der Parameter LADDR des PROFIBUS-Slave "DPSlave_12" verwendet, der ein Kopfmodul, ein Powermodul und zwei E/A-Module enthält.
- Es werden die ersten vier Bytes der ausgegebenen STATE-Parameterinformationen angezeigt.

MODE	Beispiel 1: Normalbetrieb ohne Fehler	Beispiel 2: PROFIBUS Slavegerät DPSlave_12 mit gezogenem Modul	Beispiel 3: PROFIBUS Slavegerät DPSlave_12 getrennt
1: Modulkonfiguration aktiv	0x1F00_0000	0x1F00_0000	0x1F00_0000
2: Modul defekt	0x0000_0000	0x0900_0000	0x1F00_0000
3: Modul deaktiviert	0x0000_0000	0x0000_0000	0x0000_0000
4: Modul vorhanden	0x1F00_0000	0x1700_0000	0x0000_0000
5: Problem im Modul	0x0000_0000	0x0900_0000	0x1F00_0000

Die folgenden vier Tabellen zeigen eine binäre Aufschlüsselung der vier analysierten Datenbytes:

Tabelle 9-170 Beispiel 1: Keine Fehler: Für MODE 1 (Modulkonfiguration aktiv) wird der Wert 0x1F00_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x1F	Bit 7 0001-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Die Steckplätze 1 (Bit 1) bis 4 (Bit 4) enthalten Module. Die Steckplätze 5 (Bit 5) und darüber hinaus enthalten keine Module. Die Daten von MODE 4 (Modul vorhanden) stimmen mit MODE 1 (Modulkonfiguration aktiv) überein, also entsprechen die konfigurierten Module den vorhandenen Modulen.

Tabelle 9-171 Beispiel 2: Ein Modul wurde aus dem PROFIBUS-Slavegerät "DPSlave_12" gezogen. Für MODE 2 (Modul defekt) wird der Wert 0x0900_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x09	Bit 7 0000-1001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Nur Modul 3 (Bit 3) ist als defekt gekennzeichnet. Alle anderen Module sind funktionsfähig.

Tabelle 9-172 Beispiel 2 (Fortsetzung): Ein Modul wurde aus dem PROFIBUS-Slavegerät "DPSlave_12" gezogen. Für MODE 4 (Modul defekt) wird der Wert 0x1700_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x17	Bit 7 0001-0111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0x00	Bit 15 0000-0000 Bit 8	

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Modul 3 (Bit 3) wird als fehlend gezeigt. Die Module 1, 2 und 4 (Bits 1, 2 und 4) werden als vorhanden gezeigt.

Tabelle 9-173 Beispiel 3: PROFIBUS-Slavegerät "DPSlave_12" ist vom PROFIBUS-Netzwerk getrennt (Kabel getrennt oder Spannungsverlust). Für MODE 2 (Modul defekt) wird der Wert 0x1F00_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x1F	Bit 7 0001-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Die Module in den Steckplätzen 1 bis 4 (Bits 1 bis 4) sind alle als defekt gekennzeichnet, weil das Gerät fehlt. MODE 5 (Problem im Modul) zeigt die gleichen Informationen wie MODE 2 (Modul defekt).

Beispiel in PROFINET

Das Beispiel in PROFINET besteht aus den folgenden Komponenten:

- 16 PROFINET-Slavegeräte mit den Namen "et200s_1" bis "et200s_16".
- Die 16 PROFINET-Geräte verwenden jeweils eine der PROFINET-Gerätenummern 1 bis 16.
- Für jedes Slavegerät sind mehrere E/A-Module konfiguriert.
- Im Beispiel wird der PROFINET-Slave "et200s_1" verwendet, der ein Kopfmodul, ein Powermodul und 18 E/A-Module enthält.
- Es werden die ersten vier Bytes der ausgegebenen STATE-Parameterinformationen angezeigt.

MODE	Beispiel 1: Normalbetrieb ohne Fehler	Beispiel 2: PROFINET Gezogenes Modul an Slave et200s_1	Beispiel 3: PROFINET Slave et200s_1 getrennt
1: Modulkonfiguration aktiv	0xFFFF_1F00	0xFFFF_1F00	0xFFFF_1F00
2: Modul defekt	0x0000_0000	0x0180_0000	0xFFFF_1F00
3: Modul deaktiviert	0x0000_0000	0x0000_0000	0x0000_0000
4: Modul vorhanden	0xFFFF_1F00	0xFF7F_1F00	0x0000_0000
5: Problem im Modul	0x0000_0000	0x0180_0000	0xFFFF_1F00

Die folgenden vier Tabellen zeigen eine binäre Aufschlüsselung der vier analysierten Datenbytes:

Tabelle 9-174 Beispiel 1: Keine Fehler: Für MODE 1 (Modulkonfiguration aktiv) wird der Wert 0xFFFF_1F00 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
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	

Die Steckplätze 1 (Bit 1) bis 20 (Bit 20) enthalten Module. Die Steckplätze 21 (Bit 21) und darüber hinaus enthalten keine Module. Die Daten von MODE 4 (Modul vorhanden) stimmen mit MODE 1 (Modulkonfiguration aktiv) überein, also entsprechen die konfigurierten Module den vorhandenen Modulen.

Tabelle 9-175 Beispiel 2: Ein Modul wurde aus dem PROFINET-Slavegerät "et200s_1" gezogen. Für MODE 2 (Modul defekt) wird der Wert 0x0180_0000 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0x01	Bit 7 0000-0001 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0x80	Bit 15 1000-0000 Bit 8	
Byte 3 0x00	Bit 23 0000-0000 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Nur Modul 15 (Bit 15) ist als defekt gekennzeichnet. Alle anderen Module sind funktionsfähig.

Tabelle 9-176 Beispiel 2 (Fortsetzung): Ein Modul wurde aus dem PROFIBUS-Slavegerät "et200s_1" gezogen. Für MODE 4 (Modul defekt) wird der Wert 0xFF7F_1F00 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0x7F	Bit 15 0111-1111 Bit 8	
Byte 3 0x1F	Bit 23 0001-1111 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Modul 15 (Bit 15) wird als fehlend gezeigt. Die Module 1 bis 14 (Bits 1 bis 14) und 16 bis 20 (Bits 16 bis 20) werden als fehlend gezeigt.

Tabelle 9-177 Beispiel 3: PROFINET-Slavegerät "et200s_1" ist vom PROFINET-Netzwerk getrennt (Kabel getrennt oder Spannungsverlust). Für MODE 2 (Modul defekt) wird der Wert 0xFFFF_1F00 ausgegeben.

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 1 0xFF	Bit 7 1111-1111 Bit 0	Bit 0 ist wahr; Daten sind verfügbar.
Byte 2 0xFF	Bit 15 1111-1111 Bit 8	

Byte mit Wert	Bitmuster mit Wert	Hinweise
Byte 3 0x1F	Bit 23 0001-1111 Bit 16	
Byte 4 0x00	Bit 31 0000-0000 Bit 24	

Die Module in den Steckplätzen 1 bis 20 (Bits 1 bis 20) sind alle als defekt gekennzeichnet, weil das Gerät fehlt.


MODE 5 (Problem im Modul) zeigt die gleichen Informationen wie MODE 2 (Modul defekt).

9.7.9 GET_DIAG (Diagnoseinformationen lesen)

Beschreibung

Mit der Anweisung GET_DIAG können Sie die Diagnoseinformationen eines Hardwaregeräts auslesen. Das Hardwaregerät wird über den Parameter LADDR ausgewählt. Mit dem Parameter MODE wählen Sie aus, welche Diagnoseinformationen ausgelesen werden sollen.

Tabelle 9-178 Anweisung GET_DIAG

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := GET_DIAG(mode:=_uint_in_, laddr:=_word_in_, cnt_diag=>_uint_out_, diag:=_variant_inout_, detail:=_variant_inout_);</pre>	Liest die Diagnoseinformationen aus einem zugewiesenen Hardwaregerät aus.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung GET_DIAG:

Tabelle 9-179 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
MODE	IN	UInt
LADDR	IN	HW_ANY (Word)
RET_VAL	OUT	Int
CNT_DIAG	OUT	UInt
DIAG	InOut	Variant
DETAILS	InOut	Variant

Parameter MODE

Abhängig vom Wert des Parameters MODE werden unterschiedliche Diagnosedaten an den Ausgabeparametern DIAG, CNT_DIAG und DETAILS ausgegeben:

Tabelle 9-180 Parameter MODE

MODE	Beschreibung	DIAG	CNT_DIAG	DETAILS
0	Ausgabe aller unterstützten Diagnoseinformationen für ein Modul als DWord, wobei Bit X=1 kenntlich macht, dass der Modus X unterstützt wird.	Bitzeichenfolge der unterstützten Modi als DWord, wobei Bit X=1 kenntlich macht, dass der Modus X unterstützt wird. Wenn Parameter MODE = 0 ist, ignoriert die S7-1200 CPU Parameter LADDR.	0	-
1	Ausgabe des zugehörigen Zustands des adressierten Hardwareobjekts.	Diagnosezustand: Ausgabe in Übereinstimmung mit der Struktur DIS. (Hinweis: Beachten Sie auch die Informationen zur "DIS-Struktur" unten und das Beispiel für die Anweisung GET_DIAG am Ende des Abschnitts.)	0	-
2	Ausgabe des Zustands aller untergeordneten Module des adressierten Hardwareobjekts.	Ausgabe der Diagnosedaten in Übereinstimmung mit der Struktur DNN. (Hinweis: Beachten Sie auch die Informationen zur "DNN-Struktur" unten und das Beispiel für die Anweisung GET_DIAG am Ende des Abschnitts.)	0	-

DIS-Struktur

Bei einem MODE-Parameter = 1 werden die Diagnoseinformationen in Übereinstimmung mit der Struktur DIS ausgegeben. Die folgende Tabelle zeigt die Bedeutung der einzelnen Parameterwerte:

Tabelle 9-181 Struktur der Diagnoseinformationsquelle (DIS, Diagnostic Information Source)

Parameter	Datentyp	Wert	Beschreibung
MaintenanceState	DWord	Enum	
		0	Keine Wartung erforderlich
		1	Das Modul oder Gerät ist deaktiviert.
		2	-
		3	-
		4	-
		5	Wartung notwendig
		6	Wartung angefordert
		7	Fehler
		8	Zustand unbekannt/Fehler im untergeordneten Modul
		9	-
10	Eingänge/Ausgänge sind nicht verfügbar.		

Parameter	Datentyp	Wert	Beschreibung
Componentstate Detail	DWord	Bit-Array	Zustand der Submodule des Moduls: <ul style="list-style-type: none"> • Bit 0 bis 15: Statusmeldung des Moduls • Bit 16 bis 31: Statusmeldung der CPU
		0 bis 2 (Enum)	Weitere Informationen: <ul style="list-style-type: none"> • Bit 0: Keine weiteren Informationen • Bit 1: Übertragung nicht zulässig
		3	Bit 3 = 1: Mindestens ein Kanal unterstützt Kennzeichner für die Diagnose.
		4	Bit 4 = 1: Mindestens ein Kanal oder eine Komponente muss gewartet werden
		5	Bit 5 = 1: Für mindestens einen Kanal oder eine Komponente wurde die Wartung angefordert
		6	Bit 6 = 1: Fehler bei mindestens einem Kanal oder einer Komponente
		7 bis 10	Reserviert (immer = 0)
		11 bis 14	Bit 11 = 1: PNIO - Submodul korrekt Bit 12 = 1: PNIO - Ersatzmodul Bit 13 = 1: PNIO - falsches Modul Bit 14 = 1: PNIO - Modul getrennt
		15	Reserviert (immer = 0)
		16 bis 31	Von der CPU erzeugte Zustandsinformationen für Module: Bit 16 = 1: Modul deaktiviert Bit 17 = 1: CiR-Operation aktiv Bit 18 = 1: Eingang nicht verfügbar Bit 19 = 1: Ausgang nicht verfügbar Bit 20 = 1: Überlauf Diagnosepuffer Bit 21 = 1: Diagnose nicht verfügbar Bit 22 - 31: Reserviert (immer 0)
OwnState	UInt16	Enum	Der Wert des Parameters OwnState beschreibt den Wartungszustand des Moduls.
		0	Keine Störung
		1	Das Modul oder Gerät ist deaktiviert.
		2	Wartung notwendig
		3	Wartung angefordert
		4	Fehler
		5	Das Modul oder Gerät kann von der CPU nicht erreicht werden (gilt für Module und Geräte unter einer CPU).
		6	Eingänge/Ausgänge sind nicht verfügbar.
7	-		

Parameter	Datentyp	Wert	Beschreibung
IO State	UInt16	Bit-Array	E/A-Zustand des Moduls
		0	Bit 0 = 1: Keine Wartung erforderlich
		1	Bit 1 = 1: Das Modul oder Gerät ist deaktiviert.
		2	Bit 2 = 1: Wartung notwendig
		3	Bit 3 = 1: Wartung angefordert
		4	Bit 4 = 1: Fehler
		5	Bit 5 = 1: Das Modul oder Gerät kann von der CPU nicht erreicht werden (gilt für Module und Geräte unter einer CPU).
		6	Kennzeichner; Bit 7 = 1, wenn Bit 0, 2 oder 3 gesetzt ist
		7	Eingänge/Ausgänge sind nicht verfügbar.
		8 bis 15	Reserviert (immer = 0)
OperatingState	UInt16	Enum	
		0	-
		1	In STOP/Firmware-Update
		2	In STOP/Urlöschen des Speichers
		3	In STOP/Selbstanlauf
		4	In STOP
		5	Urlöschen des Speichers
		6	In START
		7	In RUN
		8	-
		9	In HOLD
		10	-
		11	-
		12	Modul defekt
		13	-
		14	Keine Spannung
		15	CiR
		16	In STOP/ohne DIS
		17	Eingang
		18	
		19	
20			

DNN-Struktur

Bei einem MODE-Parameter = 2 werden die Diagnoseinformationsdetails in Übereinstimmung mit der Struktur DNN ausgegeben. Die folgende Tabelle zeigt die Bedeutung der einzelnen Parameterwerte:

Tabelle 9-182 Struktur des Diagnosenavigationsknotens (DNN, Diagnostic Navigation Node)

Parameter	Datentyp	Wert	Beschreibung
SubordinateState	UINT	Enum	Zustand des untergeordneten Moduls (siehe Parameter OwnState der Struktur DIS)
SubordinateIOState	WORD	Bitarray	Zustand der Eingänge und Ausgänge des untergeordneten Moduls (siehe Parameter IO State der Struktur DIS)
DNNmode	WORD	Bitarray	<ul style="list-style-type: none"> • Bit 0 = 0: Diagnose aktiviert • Bit 0 = 1: Diagnose deaktiviert • Bit 1 bis 15: Reserviert

Parameter RET_VAL

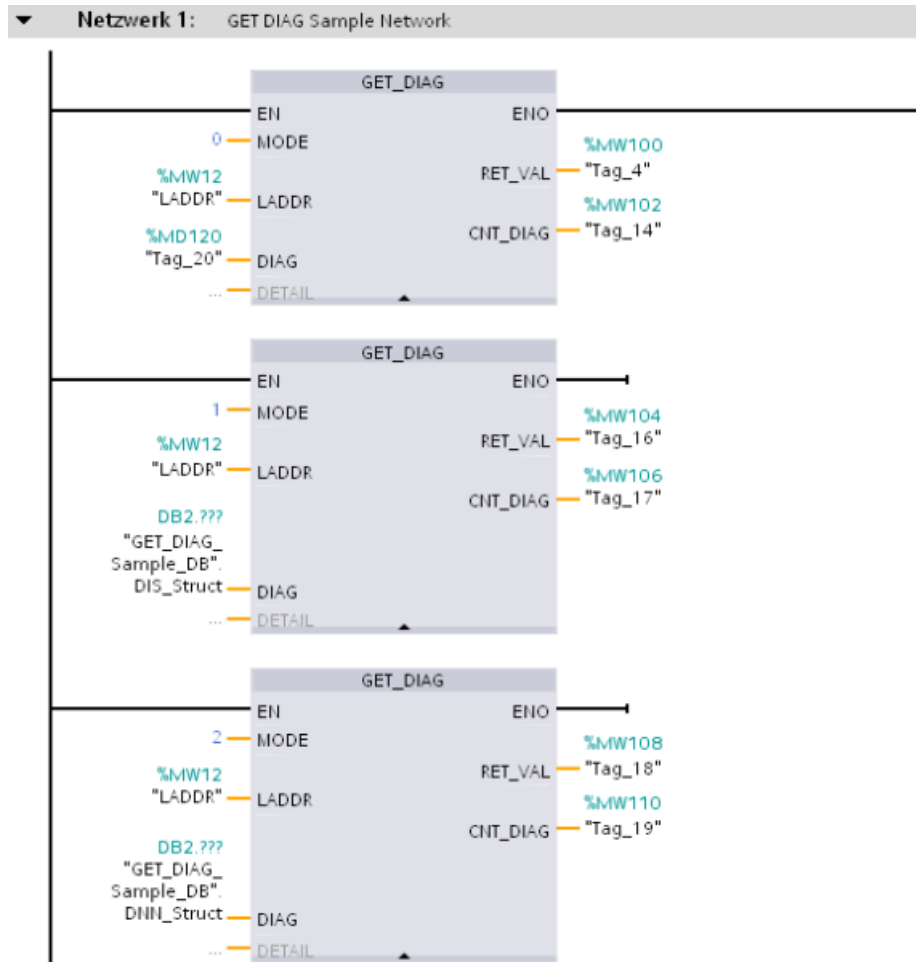
Tabelle 9-183 Fehlercodes des Parameters RET_VAL

Fehlercode (W#16#...)	Beschreibung
0	Kein Fehler
8080	Wert im Parameter MODE wird nicht unterstützt.
8081	Typ im Parameter DIAG wird nicht mit dem ausgewählten Modus unterstützt (Parameter MODE).
8082	Typ im Parameter DETAILS wird nicht mit dem ausgewählten Modus unterstützt (Parameter MODE).
8090	LADDR ist nicht vorhanden.
8091	Der ausgewählte Kanal im Parameter CHANNEL ist nicht vorhanden.
80C1	Nicht genügend Ressourcen für die parallele Ausführung

Beispiel

Das folgende KOP-Netzwerk und der folgende DB zeigen, wie Sie die drei Betriebsarten mit den drei Strukturen verwenden:

- DIS
- DNN



GET_DIAG_Sample_DB							
	Name	Datentyp	Offset	Startwert	Rem...	Sichtbar in ...	Kommentar
1	Static						
2	DNN_Struct	DNN	0.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
3	SubordinateState	UInt	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
4	SubordinateIOState	Word	2.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
5	DNNmode	Word	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
6	DIS_Struct	DIS	6.0		<input type="checkbox"/>	<input checked="" type="checkbox"/>	
7	MaintenanceState	DWord	0.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
8	ComponentStateDetail	DWord	4.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
9	OwnState	UInt	8.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
10	IOState	Word	10.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
11	OperatingState	UInt	12.0	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	

- ① DNN
② DIS

Hinweis

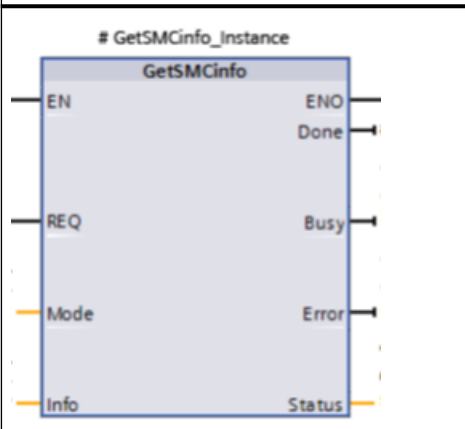
Im DB müssen Sie den Datentyp manuell eingeben, um auf jede der drei Strukturen zuzugreifen. Eine Auswahl über eine Klappliste ist nicht möglich. Geben Sie die Datentypen genau wie im Folgenden gezeigt ein:

- DNN
- DIS

9.7.10 GetSMCInfo (Informationen über die Memory Card auslesen)

Die Anweisung GetSMCInfo ruft Informationen über die gesteckte SIMATIC Memory Card ab. Die auszulesende Information wird mit dem Parameter "Mode" angegeben.

Tabelle 9-184 Anweisung GetSMCInfo

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := GetSMCInfo(Mode:=_uint_in_, Info:=_variant_inout_);</pre>	<p>Mit der Anweisung "GetSMCInfo" lesen Sie Informationen über die gesteckte Memory Card aus. Ist keine Memory Card gesteckt, gibt die Anweisung den Fehlercode W#16#8081 aus. Über den Parameter Mode wählen Sie die zu lesenden Informationen aus.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "GetSMCInfo":

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Steuerparameter Request Aktiviert das Lesen der Informationen mit REQ = "1".
Mode	IN	UInt	Die Informationen über die SIMATIC Memory Card, die Sie auslesen möchten, wählen Sie mit Hilfe des Parameters Mode aus: <ul style="list-style-type: none"> • 0: Speichergröße in KiB (1 KiB = 1024 Byte) • 1: Belegter Speicher in KiB • 2: Wartungsinformationen: Zuvor verwendeter Anteil der Lebensdauer in % Hinweis: <ul style="list-style-type: none"> – Die S7-1200 liefert jetzt die Wartungsinformationen. Bei S7-PLCSIM wird immer 0x00 in Info eingegeben (keine Wartungsinformationen verfügbar). – Wenn die Informationen von der SIMATIC Memory Card oder der aktuell installierten Firmware Ihrer CPU nicht unterstützt werden, wird 0xFF in Info eingegeben. • 3: Wenn die festgelegte Schwelle der Lebensdauer in % überschritten wird, erstellt die PLC einen Eintrag im Diagnosepuffer und aktiviert die Wartungs-LED. Hinweis: <ul style="list-style-type: none"> – Wenn Sie die Generierung von Diagnosealarmen deaktiviert haben, wird 0xFF in Info eingegeben. – Bei S7-PLCSIM wird immer 0xFF in Info eingegeben. • 10 oder 20: Entspricht Mode 0, ist aber für S7-1500-R/H CPUs reserviert. • 11 oder 21: Entspricht Mode 1, ist aber für S7-1500-R/H CPUs reserviert. • 12 oder 22: Entspricht Mode 2, ist aber für S7-1500-R/H CPUs reserviert. • 13 oder 23: Entspricht Mode 3, ist aber für S7-1500-R/H CPUs reserviert.
Done	OUT	Bool	1: Die Anweisung wurde erfolgreich ausgeführt. Die ausgelesenen Informationen werden an den Parameter Info übertragen.
Busy	OUT	Bool	Statusparameter <ul style="list-style-type: none"> • 0: Die Ausführung der Anweisung ist beendet • 1: Die Ausführung der Anweisung ist noch nicht beendet
Error	OUT	Bool	Statusparameter <ul style="list-style-type: none"> • 0: Kein Fehler. • 1: Bei der Ausführung der Anweisung ist ein Fehler aufgetreten Ausführliche Informationen werden am Parameter Status ausgegeben
Status	OUT	Word	Fehlercode
Info	INOUT	UDInt	Puffer für die ausgelesenen Informationen

Weitere Informationen zu gültigen Datentypen finden Sie unter "AUTOHOTSPOT".

Hinweis

Ermitteln der I&M 0-Daten

Die I&M 0-Daten der Memory Card können mit der Anweisung GetSMCinfo nicht ermittelt werden. Verwenden Sie hierfür die Anweisung "Get_IM_Data".

Parameterstatus

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler
7000	Kein Auftrag in Bearbeitung.
7001	Erster Aufruf: Auftrag anstoßen (Betrieb = 1, Fertig = 0).
7002	Zwischenzeitlicher Aufruf: Auftrag bereits aktiv (Betrieb = 1, Fertig = 0).
8080	Ungültiger Wert des Parameters "Mode"
8081	Keine SIMATIC Memory Card gesteckt
8092	Keine Daten verfügbar, beispielsweise weil "GetSMCinfo" von der CPU nicht unterstützt wird
80C3	Die maximale Anzahl gleichzeitiger Aufrufe der Anweisung "GetSMCinfo" ist erreicht.

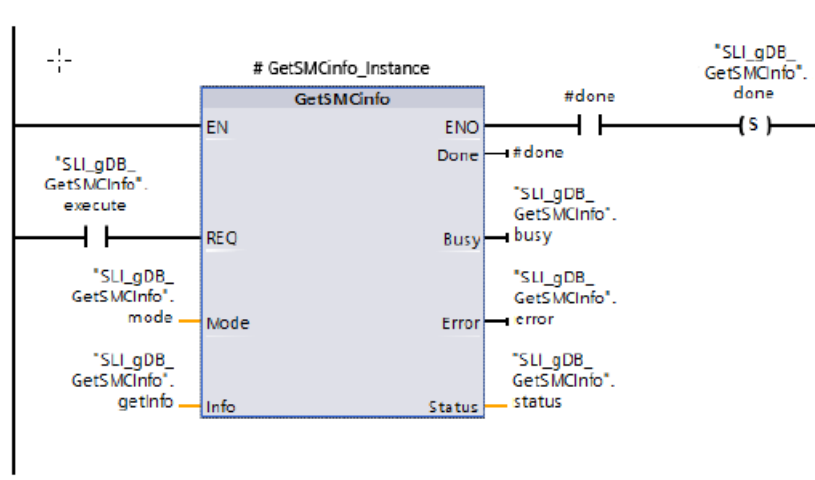
* Die Fehlercodes werden im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".

Beispiel: Speichergröße der verwendeten SIMATIC Memory Card ermitteln

Im folgenden Beispiel ermitteln Sie die Speichergröße der verwendeten SIMATIC Memory Card. Erstellen Sie die folgenden Variablen zum Speichern der Daten in einem globalen Datenbaustein:

SLL_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

Erstellen Sie einen FB. Erstellen Sie im Statusbereich des FB eine lokale Variable "#done" mit dem Datentyp Bool. Verschalten Sie die Parameter der Anweisung "GetSMCinfo" wie folgt:



Die Anweisung "GetSMCInfo" wird nur ausgeführt, wenn der Eingangsparameter REQ ("execute") den Signalzustand "WAHR" zurückgibt. Der zum Auslesen der SIMATIC Memory Card zu verwendende Modus ist im Eingangsparameter MODE ("mode") hinterlegt. Im folgenden Beispiel wird die Speichergröße der SIMATIC Memory Card entsprechend dem für "mode" festgelegten Wert "0" ausgelesen und in KB am Parameter INFO ("getInfo") angezeigt. Der Erfolgsstatus von GetSMCInfo wird am Ausgangsparameter DONE ("done") angezeigt und in der Variable "done" gespeichert.

Die Ausgangsparameter STATUS ("status") und ERROR ("error") weisen darauf hin, dass die Verarbeitung in dem Beispiel fehlerfrei abgeschlossen wurde.

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 Diagnoseereignisse für die dezentrale Peripherie

Hinweis

In einem PROFIBUS IO-System geht die CPU nach einem Ladevorgang oder dem Aus- und Wiedereinschalten in den Betriebszustand RUN, sofern die Hardwarekompatibilität nicht so eingestellt ist, dass akzeptable Ersatzmodule (Seite 153) zulässig sind und mindestens ein Modul fehlt oder es sich dabei nicht um ein akzeptables Ersatzmodul für das konfigurierte Modul handelt.

9.8 Impuls

Wie Sie in der folgenden Tabelle sehen, unterstützt die CPU diejenige Diagnose, die für die Komponenten des dezentralen E/A-Systems konfiguriert werden kann. Jeder dieser Fehler erzeugt einen Protokolleintrag im Diagnosepuffer.

Tabelle 9-185 Behandlung von Diagnoseereignissen bei PROFINET und PROFIBUS

Fehlertyp	Diagnoseinformation für die Station?	Eintrag im Diagnosepuffer?	CPU-Betriebszustand
Diagnosefehler	Ja	Ja	Bleibt im Betriebszustand RUN
Fehler bei Baugruppenträger oder Station	Ja	Ja	Bleibt im Betriebszustand RUN
Peripheriezugriffsfehler ¹	Nein	Ja	Bleibt im Betriebszustand RUN
Peripheriezugriffsfehler ²	Nein	Ja	Bleibt im Betriebszustand RUN
Ziehen/Stecken-Ereignis	Ja	Ja	Bleibt im Betriebszustand RUN

- ¹ Beispielursache für einen Peripheriezugriffsfehler: Ein Modul wurde entfernt.
- ² Beispielursache für einen Peripheriezugriffsfehler: Azyklische Kommunikation mit einem Submodul, das nicht kommuniziert.

Über die Anweisung GET_DIAG (Seite 459) können Sie für jede Station die Diagnoseinformationen abrufen. Auf diese Weise können Sie die im Gerät aufgetretenen Fehler programmatisch behandeln und, sofern gewünscht, die CPU in den Betriebszustand STOP versetzen. Für dieses Verfahren müssen Sie das Hardwaregerät angeben, aus dem die Statusinformationen ausgelesen werden sollen.

Die Anweisung GET_DIAG verwendet die "L-Adresse" (LADDR) der Station, um den Zustand der gesamten Station abzurufen. Diese L-Adresse finden Sie in der Netzsicht der Gerätekonfiguration, indem Sie den gesamten Baugruppenträger der Station auswählen (den gesamten grauen Bereich). Die L-Adresse wird in der Registerkarte "Eigenschaften" der Station angezeigt. Sie finden den Parameter LADDR für jedes einzelne Modul entweder in den Eigenschaften des Moduls (in der Gerätekonfiguration) oder in der Standardvariablen-tabelle der CPU.

9.8 Impuls

9.8.1 CTRL_PWM (Impulsdauermodulation)

Tabelle 9-186 Anweisung CTRL_PWM (Impulsdauermodulation)

KOP/FUP	SCL	Beschreibung
	<pre>"CTRL_PWM_DB" (PWM:=_uint_in_, ENABLE:=_bool_in_, BUSY=>_bool_out_, STATUS=>_word_out_);</pre>	<p>Bietet eine feste Zykluszeit mit variabler relativer Einschaltdauer. Der PWM-Ausgang läuft nach dem Start kontinuierlich mit der angegebenen Frequenz (Zykluszeit). Die Impulsdauer wird nach Bedarf verändert, um die gewünschte Steuerung zu erzielen.</p>

- ¹ Wenn Sie die Anweisung einfügen, zeigt STEP 7 den Dialog "Aufrufoptionen" zum Erstellen des zugehörigen DB an.
- ² Im SCL-Beispiel ist "CTRL_PWM_DB" der Name des Instanz-DBs.

Tabelle 9-187 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
PWM	IN	HW_PWM (Word)	PWM-Kennung: Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablen-Tabelle "Konstanten" und können als PWM-Parameter genutzt werden. (Standardwert: 0)
ENABLE	IN	Bool	1 = Impulsgenerator starten 0 = Impulsgenerator stoppen
BUSY	OUT	Bool	Funktion beschäftigt (Standardwert: 0)
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Die Anweisung CTRL_PWM speichert die Parameterinformationen im DB. Die Parameter des Datenbausteins werden nicht separat vom Anwender geändert, sondern von der Anweisung CTRL_PWM gesteuert.

Geben Sie den gewünschten Impulsgenerator über den Variablennamen für den Parameter PWM an.

Wenn der Eingang EN gleich WAHR ist, startet oder stoppt die Anweisung PWM_CTRL die angegebene PWM anhand des Werts am Eingang ENABLE. Die Impulsdauer wird von dem Wert der zugewiesenen Adresse des Ausgangsworts angegeben.

Weil die CPU die Anforderung verarbeitet, während die Anweisung CTRL_PWM ausgeführt wird, meldet der Parameter BUSY immer FALSCH. Wird ein Fehler erkannt, wird ENO auf FALSCH gesetzt und der Parameter STATUS enthält einen Fehlercode.

Die Impulsdauer wird beim ersten Wechsel der CPU in RUN auf den in der Gerätekonfiguration eingegebenen Anfangswert gesetzt. Zum Ändern der Impulsdauer schreiben Sie die gewünschten Werte in die in der Gerätekonfiguration angegebene Ausgangswortadresse ("Ausgangsadressen"/"Anfangsadresse:"). Um die gewünschte Impulsdauer in das entsprechende Ausgangswort zu schreiben, nutzen Sie eine Anweisung wie Übertragen, Umwandeln, Arithmetik oder PID. Sie müssen den gültigen Bereich für den Wert des Ausgangsworts einhalten (Prozent, Tausendstel, Zehntausendstel oder S7-Analogformat).

Hinweis

Zu PWM und PTO zugewiesene digitale E/A können nicht geforct werden.

Die von der Impulsdauermodulation (PWM) und der Impulsfolge (PTO) verwendeten digitalen E/A werden während der Gerätekonfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Funktion zum Forcen in der Beobachtungstabelle geändert werden.

Tabelle 9-188 Wert des Parameters STATUS

STATUS	Beschreibung
0	Kein Fehler
80A1	PWM-Kennung adressiert keine gültige PWM.

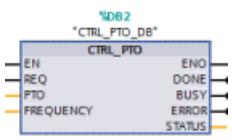
9.8.2 CTRL_PTO (Impulsfolge)

Die PTO-Anweisung liefert einen Rechteckausgang mit einer relativen Einschaltdauer von 50 % mit einer angegebenen Frequenz. Mit der Anweisung CTRL_PTO können Sie die Frequenz ohne ein Technologieobjekt (TO) Achsen-Datenbaustein (DB) zuweisen.

Diese Anweisung erfordert einen Impulsgenerator. Sie müssen den Impulsgenerator in der Hardwarekonfiguration aktivieren und einen Signaltyp auswählen. Weitere Informationen finden Sie unter "Konfigurieren eines Impulskanals für PWM oder PTO" (Seite 477).

Zugriff auf die Anweisung CTRL_PTO haben Sie in den Task Cards, Erweiterte Anweisungen.

Tabelle 9-189 Anweisung CTRL_PTO (Impulsfolge)

KOP/FUP ¹	SCL ²	Beschreibung
	<pre>"CTRL_PTO_DB" (REQ:=_bool_in_, PTO:=_uint_in_, FREQUENCY:=_udint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	Über die PTO-Anweisung können Sie die Frequenz eines Rechteckausgangs (mit einer relativen Einschaltdauer von 50 %) steuern.

¹ Wenn Sie die Anweisung einfügen, zeigt STEP 7 den Dialog "Aufrufoptionen" zum Erstellen des zugehörigen DB an.

² Im SCL-Beispiel ist "CTRL_PTO_DB" der Name des Instanz-DBs.

Tabelle 9-190 Datentypen für die Parameter

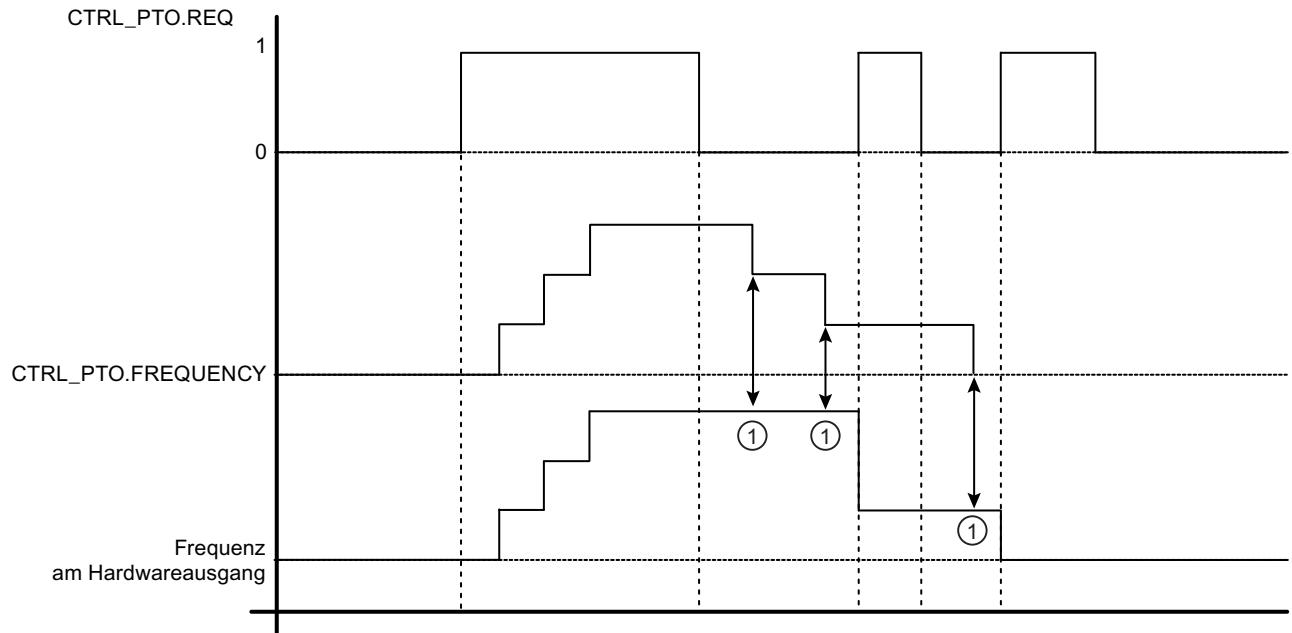
Parameter und Datentyp	Datentyp	Beschreibung
EN	IN	Bool 1 = Anweisung aktiviert 0 = Anweisung deaktiviert
REQ	IN	Bool 1 = PTO-Ausgangsfrequenz auf den Wert des Eingangs FREQUENCY festlegen 0 = Keine Änderung der PTO
PTO	IN	HW_PTO (Word) PTO-Kennung: Hardwarekennung des Impulsgenerators: <ul style="list-style-type: none"> Die Namen der aktivierten Impulsgeneratoren werden zu Variablen in der Variablen-tabelle "Konstanten" und können als PTO-Parameter genutzt werden. (Standardwert = 0.) Die Hardware-ID finden Sie in den Eigenschaften des Impulsgenerators in der Gerätesicht. Die Systemkonstanten führen die Hardware-IDs der Impulsgeneratoren ebenfalls auf. (Standardwert = 0.)
FREQUENCY	IN	UDInt Gewünschte Frequenz (in Hz) der PTO. Dieser Wert gilt nur, wenn REQ = 1 (der Standardwert ist 0 Hz).
DONE	OUT	Bool Funktion fehlerfrei beendet (Standardwert: 0)
BUSY	OUT	Bool Funktion beschäftigt (Standardwert: 0)
ERROR	OUT	Word Fehler erkannt (Standardwert: 0)
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Die Anweisung CTRL_PTO speichert die Parameterinformationen im DB. Die Parameter des Datenbausteins werden nicht separat vom Anwender geändert, sondern von der Anweisung CTRL_PTO gesteuert.

Geben Sie den gewünschten aktivierten Impulsgenerator über den Variablennamen oder die Hardwarekennung am Parameter PTO an.

Wenn der Eingang EN gleich WAHR ist, startet oder stoppt die Anweisung CTRL_PTO der angegebenen PTO. Wenn der Eingang EN gleich FALSCH ist, wird die Anweisung CTRL_PTO nicht ausgeführt, und die PTO bleibt in ihrem aktuellen Zustand.

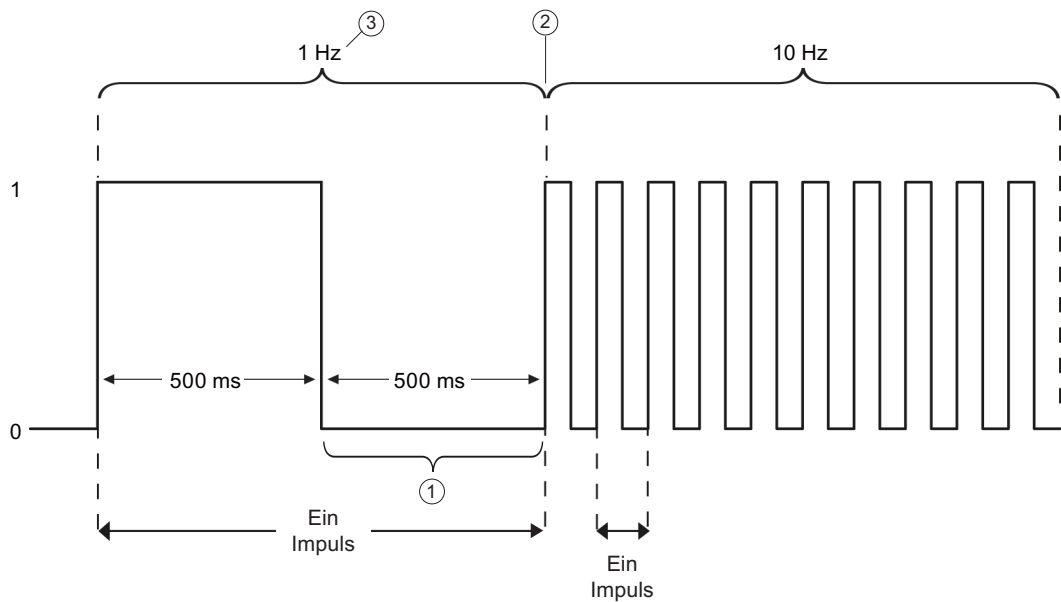
Wenn Sie den Eingang REQ auf WAHR setzen, wird der Wert FREQUENCY wirksam. Fall REQ gleich FALSCH ist, kann die Ausgangsfrequenz der PTO nicht geändert werden, und die PTO gibt weiterhin Impulse aus.



① Keine Änderung der Ausgangsfrequenz, solange REQ = 0

Da die Anweisung CTRL_PTO die PTO nur startet, wird die Anweisung CTRL_PTO sofort beendet. Deshalb wird der Ausgang BUSY nie eingeschaltet. Der Ausgang DONE wird eingeschaltet, solange kein Fehler auftritt. Wird ein Fehler erkannt, wird der Parameter ERROR auf WAHR gesetzt, und der Parameter STATUS enthält einen Fehlercode.

Wenn die Anweisung CTRL_PTO mit einer vorgegebenen Frequenz aktiviert wird, gibt die S7-1200 eine Impulsfolge mit dieser angegebenen Frequenz aus. Die gewünschte Frequenz kann jederzeit geändert werden. Bei einer Änderung der Frequenz beendet die S7-1200 den aktuellen Impuls zunächst, bevor sie die Frequenz in den neuen gewünschten Wert ändert. Beispiel: Ist die gewünschte Frequenz 1 Hz (wofür 1000 ms für die Durchführung benötigt werden) und die Frequenz wird nach 500 ms in 10 Hz geändert, wird die Änderung der Frequenz am Ende des Zeitraums von 1000 ms vorgenommen.



- ① Die Frequenz wird nach 500 ms in 10 Hz geändert.
- ② Der 1-Hz-Impuls muss beendet werden, bevor die Frequenz in den neuen Wert von 10 Hz geändert werden kann.
- ③ 1 Hz entspricht 1000 ms.

Das Hardwareobjekt des Impulsgenerators unterliegt den folgenden Einschränkungen: Nur eine Anweisung kann den Impulsgenerator als PTO nutzen, und die Verwendung des Impulsgenerators wird vom Hardware-Konfigurationseditor verwaltet. Andere Anweisungen, die versuchen, auf die PTO zuzugreifen, geben einen Fehler aus: "0x8090" (Impulsgenerator mit der angegebenen Hardware-ID wird verwendet).

Hinweis

Zu PWM und PTO zugewiesene digitale E/A können nicht geforct werden.

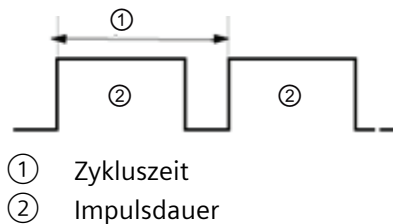
Die von der Impulsdauermodulation (PWM) und der Impulsfolge (PTO) verwendeten digitalen E/A werden während der Gerätekonfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Funktion zum Forcen in der Beobachtungstabelle geändert werden.

Tabelle 9-191 Fehlercodewerte am Parameter STATUS

Fehlercode (W#16#...)	Beschreibung
0	Keine Fehler
0x8090	Der Impulsgenerator mit der angegebenen Hardware-ID wird verwendet.
0x8091	Die Frequenz liegt außerhalb des Bereichs. Die gewünschte Frequenz überschreitet die maximale Frequenz des ausgewählten Impulsausgangs.
0x80A1	PTO-Kennung (Hardware-ID) adressiert keine gültige PTO.

Fehlercode (W#16#...)	Beschreibung
0x80D0	Der Impulsgenerator mit der angegebenen Hardware-ID ist nicht aktiviert. Aktivieren Sie den Impulsgenerator in den CPU-Eigenschaften unter "Impulsgeneratoren (PTO/PWM)".
0x80D1	Der Impulsgenerator mit der angegebenen Hardware-ID hat keine PTO-Auswahl. Wählen Sie die PTO in der Hardwarekonfiguration aus.

9.8.3 Funktionsweise der Impulsausgänge



Die Impulsdauer kann als Hundertstel der Zykluszeit (0 bis 100), als Tausendstel (0 bis 1000), als Zehntausendstel (0 bis 10000) oder als S7-Analogformat angegeben werden.

Die Impulsdauer kann zwischen 0 (kein Impuls, immer aus) und Vollausschlag (kein Impuls, immer ein) liegen.

Weil der PWM-Ausgang zwischen 0 und Vollausschlag liegen kann, bietet er einen digitalen Ausgang, der in vielerlei Hinsicht einem Analogausgang gleicht. Der PWM-Ausgang kann z. B. zur Steuerung der Drehzahl eines Motors vom Stillstand bis zur vollen Drehzahl dienen oder er kann dafür eingesetzt werden, die Position eines Ventils von geschlossen bis vollständig geöffnet zu steuern.

Die Frequenz konfigurieren Sie in der Hardwarekonfiguration (Seite 477). Die Impulsdauer steuern Sie über das Anwenderprogramm

Zur Steuerung schneller Impulsausgänge stehen vier Impulsgeneratoren zur Verfügung: PWM und Impulsfolge (PTO). PTO wird von den Bewegungssteuerungsanweisungen genutzt. Sie können jeden Impulsgenerator entweder PWM oder PTO zuordnen, jedoch nicht beiden gleichzeitig.

Sie können integrierte CPU-Ausgänge oder die Ausgänge eines optionalen Signalboards nutzen. In der folgenden Tabelle werden die Adressen der Ausgänge aufgeführt (wobei die Standardkonfiguration der Ausgänge vorausgesetzt wird). Wenn Sie die Adressen der Ausgänge geändert haben, entsprechen die Adressen den von Ihnen zugewiesenen. Beachten Sie, dass PWM nur einen Ausgang benötigt, während PTO optional zwei Ausgänge je Kanal nutzen kann. Wenn ein Ausgang für eine Impulsfunktion nicht erforderlich ist, steht er zu anderen Zwecken zur Verfügung. Die E/A-Zuweisung entnehmen Sie der folgenden Tabelle.

Die nachstehende Tabelle zeigt die Standard-E/A-Zuweisungen. Die vier Impulsgeneratoren können jedoch für jeden integrierten oder digitalen SB-Ausgang einer CPU konfiguriert werden.

Die verschiedenen Ausgänge unterstützen unterschiedliche Spannungen und Geschwindigkeiten. Berücksichtigen Sie dies bei der Zuweisung der PWM/PTO-Ausgänge.

Hinweis

Impulsfolgen können nicht von anderen Operationen im Anwenderprogramm verwendet werden.

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM- oder Bewegungssteuerungs- PTO-Anweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Anwenderprogramm verwendet werden. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Hinweis

PTO-Richtungsausgänge können für die Nutzung an anderer Stelle im Programm freigesetzt werden.

Für jeden PTO müssen zwei Ausgänge zugewiesen werden: ein Ausgang als Impulsausgang und ein Ausgang als Richtungsausgang. Sie können auch nur den Impulsausgang und nicht den Richtungsausgang verwenden. So können Sie den Richtungsausgang für andere Zwecke in Ihrem Anwenderprogramm freisetzen.

Tabelle 9-192 Standard-Ausgangszuweisungen der Impulsgeneratoren³

Beschreibung	Impuls	Richtung
PTO1		
Integrierte E/A	A0.0	A0.1
SB-E/A	A4.0	A4.1
PWM1		
Integrierte Ausgänge	A0.0	-
SB-Ausgänge	A4.0	-
PTO2		
Integrierte E/A	A0.2	A0.3
SB-E/A	A4.2	A4.3
PWM2		
Integrierte Ausgänge	A0.2	-
SB-Ausgänge	A4.2	-
PTO3		
Integrierte E/A	A0.4 ¹	A0.5 ¹
SB-E/A	A4.0	A4.1
PWM3		
Integrierte Ausgänge	A0.4 ¹	-
SB-Ausgänge	A4.1	-
PTO4		
Integrierte E/A	A0.6 ²	A0.7 ²

Beschreibung	Impuls	Richtung
SB-E/A	A4.2	A4.3
PWM4		
Integrierte Ausgänge	A0.6 ²	-
SB-Ausgänge	A4.3	-

¹ Die CPU 1211C hat keine Ausgänge A0.4, A0.5, A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1211C nicht verwendet werden.

² Die CPU 1212C hat keine Ausgänge A0.6 und A0.7. Deshalb können diese Ausgänge bei der CPU 1212C nicht verwendet werden.

³ Diese Tabelle gilt für die PTO/PWM-Funktionen der CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C und CPU 1217C.

9.8.4 Konfigurieren eines Impulskanals für PWM oder PTO

Um den PWM- oder PTO-Betrieb vorzubereiten, konfigurieren Sie zunächst einen Impulskanal in der Gerätekonfiguration durch Auswahl der CPU, des Impulsgenerators (PTO/PWM) und PWM1/PTO1 bis PWM4/PTO4. Aktivieren Sie den Impulsgenerator (Optionskästchen). Wenn ein Impulsgenerator aktiviert wird, wird diesem bestimmten Impulsgenerator ein eindeutiger Standardname zugewiesen. Diesen Namen können Sie im Bearbeitungsfeld "Name:" ändern, der Name muss jedoch eindeutig sein. Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablentabelle "Konstanten" und können genutzt werden als:

- PWM-Parameter der Anweisung CTRL_PWM
- PTO-Parameter der Anweisung CTRL_PTO

Im Bearbeitungsfeld "Kommentar:" können Sie auch einen Kommentar zu diesem spezifischen Impulsgenerator eingeben.

Tabelle 9-193 CPU-Ausgang: Höchstfrequenz (PTO) und Mindestzykluszeit (PWM)

CPU	CPU-Ausgangskanal:	PTO-Höchstfrequenz	PWM-Mindestzykluszeit
1211C	Aa.0 bis Aa.3	100 kHz	10 µs
1212C	Aa.0 bis Aa.3	100 kHz	10 µs
	Aa.4, Aa.5	20 kHz	50 µs
1214C und 1215C	Aa.0 bis Aa.3	100 kHz	10 µs
	Aa.4 bis Ab.1	20 kHz	50 µs
1217C	DAa.0 bis DAa.3 (.0+, .0- bis .3+, .3-)	1 MHz	1 µs
	DAa.4 bis DAb.1	100 kHz	10 µs

Tabelle 9-194 Signalboard-Ausgang: Höchsthfrequenz (PTO) und Mindestzykluszeit (PWM)

Signalboard (SB)	SB-Ausgangskanal	PTO-Höchsthfrequenz	PWM-Mindestzykluszeit
SB 1222, 200 kHz	DAe.0 bis DAe.3	200 kHz	5 μ s
SB 1223, 200 kHz	DAe.0, DAe.1	200 kHz	5 μ s
SB 1223	DAe.0, DAe.1	20 kHz	50 μ s

Hinweis

Die Mindestzykluszeit jedes Ausgangs von CPU und Signalboard ist in den Tabellen oben angegeben. Sie werden vom TIA Portal jedoch nicht gewarnt, wenn Sie einen PWM-Impulsgenerator mit einer Zykluszeit, die diese Mindestzykluszeit der Hardware unterschreitet, konfigurieren. Dies kann zu Problemen in Ihrer Anwendung führen. Stellen Sie also stets sicher, dass sich die Zykluszeit innerhalb der Hardwaregrenzwerte befindet.

Hinweis

Wenn Sie die Impulsdauer eines PWM-Signals festlegen, muss die tatsächliche Zeit der Impulsdauer (Zeit, die der Impuls hoch ist) größer oder gleich 1 Millisekunde sein, sofern die Zeitbasis "Millisekunden" ist. Wenn die Zeitbasis "Mikrosekunden" ist, muss die tatsächliche Zeit der Impulsdauer größer oder gleich 1 Mikrosekunde sein. Der Ausgang wird ausgeschaltet, wenn die Impulsdauer kleiner als 1 "Zeitbasis" ist.

Beispiel: Eine Zykluszeit von 10 Mikrosekunden und eine Impulsdauer von 5 Hundertstel erzeugt eine Impulsdauerzeit von 0,5 Mikrosekunden. Da dieser Wert unter 1 Mikrosekunde liegt, ist das PWM-Signal aus.

Parametrierung

Im Bereich für die Parametrierung können die Parameter des Ausgangsimpulses konfiguriert werden. Die folgenden Optionen sind abhängig davon, ob PWM oder PTO ausgewählt wird, verfügbar:

- Signaltyp: Impulsausgang als PWM oder PTO konfigurieren. Zur PTO-Auswahl finden Sie weitere Informationen im Abschnitt "Phasenlage".
 - PWM
 - PTO (Impuls A und Richtung B)
 - PTO (Impuls Vorwärtszählen A und Impuls Rückwärtszählen B)
 - PTO (A/B phasenverschoben)
 - PTO (A/B phasenverschoben - vierfach)
- Zeitbasis (gilt nur für PWM): Auswahl der zu verwendenden Zeiteinheiten:
 - Millisekunden
 - Mikrosekunden

- Impulsdauerformat (gilt nur für PWM): Zuweisung der Auflösung der Impulsdauer (Breite):
 - Hundertstel (0 bis 100)
 - Tausendstel (0 bis 1000)
 - Zehntausendstel (0 bis 10000)
 - S7-Analogformat (0 bis 27648)
- Zykluszeit (gilt nur für PWM): Zuweisung der Zeitdauer für die Durchführung eines Zyklus (Zeit hoher Impuls plus Zeit niedriger Impuls entspricht der Zykluszeit). Sie können die Zykluszeit während der Laufzeit ändern, indem Sie das Kontrollkästchen "Änderung der Zykluszeit zur Laufzeit gestatten" aktivieren. Weitere Informationen finden Sie im nachfolgenden Abschnitt "E/A-Adressen". Der Bereich ist von 1 bis 16.777.215 Zeiteinheiten.
- Anfangsimpulsdauer (gilt nur für PWM): Zuweisung der Impulsdauer des ersten Impulses. Sie können diesen Wert während der Laufzeit über die in den E/A-Adressen konfigurierte Ausgangswortadresse ändern. Der Bereich basiert auf dem Impulsdauerformat.
- Änderung der Zykluszeit zur Laufzeit gestatten (gilt nur für PWM): Wenn Sie diese Option auswählen, kann Ihr Programm die Zykluszeit des PWM-Signals ändern, während das Programm läuft. Weitere Informationen finden Sie im nachfolgenden Abschnitt "E/A-Adressen".

Hinweis

Berücksichtigen Sie beim Einstellen der Impulsdauer eines PWM-Signals die Schaltverzögerung des Ausgangskanals wie in Anhang A spezifiziert. Die am Ausgang gemessene tatsächliche Impulsdauer kann größer sein als die gewählte Impulsdauer. Der Anstieg der Impulsdauer ist bei kleiner Impulsdauer und höheren Frequenzen ausgeprägter. Stellen Sie sicher, dass die am Ausgang gemessene Impulsdauer Ihren Anforderungen entspricht.

Ermittlung des Impulsdauerwerts

Die "Impulsdauer" ergibt sich durch Multiplizieren der "Anfangsimpulsdauer" mit der "Zykluszeit". Wenn Sie eine "Zeitbasis", ein "Impulsdauerformat", eine "Zykluszeit" und die "Anfangsimpulsdauer" wählen, müssen Sie berücksichtigen, dass die Summe der "Impulsdauer" kein Bruchwert sein darf. Wenn Ihre resultierende "Impulsdauer" ein Bruchwert ist, müssen Sie Ihre "Anfangsimpulsdauer" anpassen oder Ihre "Zeitbasis" ändern, um eine Ganzzahl zu erzeugen.

Hier zwei Beispiele:

- Beispiel 1: Sie wählen die folgenden Werte:
 - Zeitbasis = Millisekunden (ms)
 - Impulsdauerformat = Hundertstel (0 bis 100)
 - Zykluszeit = 3 ms
 - Anfangsimpulsdauer = 75

Resultierende "Impulsdauer" = $0,75 \times 3 \text{ ms} = 2,25 \text{ ms}$

Dieser Wert von "Impulsdauer" ist ein Bruchwert und verursacht einen Fehler, wenn Sie die Anweisung CTRL_PWM ausführen. Der Wert von "Impulsdauer" muss ein Ganzzahlwert sein.

- Beispiel 2: Sie wählen die folgenden Werte:
 - Zeitbasis = Mikrosekunden (μs)
 - Impulsdauerformat = Hundertstel (0 bis 100)
 - Zykluszeit = 3000 μs
 - Anfangsimpulsdauer = 75

Resultierende "Impulsdauer" = $0,75 \times 3000 \mu\text{s} = 2250 \mu\text{s}$

Dieser Wert von "Impulsdauer" ist ein Ganzzahlwert, und mit diesem Wert funktioniert die Anweisung CTRL_PWM einwandfrei.

Hardwareausgänge

Wählen Sie im Bereich der Hardwareausgänge den Ausgangskanal im Dropdown-Menü aus. Je nach Konfiguration steht einer oder stehen zwei Ausgänge zur Auswahl. Wenn Sie einen Ausgangskanal einem Impulsgenerator zuweisen, kann der Ausgangskanal nicht von einem anderen Impulsgenerator, HSC oder Prozessabbild verwendet werden.

Hinweis

Impulsgeneratorausgänge können nicht von anderen Anweisungen im Anwenderprogramm verwendet werden.

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM, PTO oder Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Ausgangsadressen aus dem Speicher der Ausgänge entfernt und können nicht für andere Zwecke in Ihrem Programm verwendet werden. Wenn Ihr Programm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

E/A-Adressen

Die PWM hat zwei Bytes im Speicherbereich A für die "Impulsdauer". Während die PWM läuft, können Sie den Wert im zugeordneten Speicherbereich A ändern und damit die Impulsdauer modifizieren.

Geben Sie im Bereich der E/A-Adressen die Adresse des Ausgangsworts ein, wo Sie den Wert für die Impulsdauer speichern möchten.

Die Standardadressen für die PWM-Impulsdauerwerte sind:

- PWM1: AW1000
- PWM2: AW1002
- PWM3: AW1004
- PWM4: AW1006

Bei PWM steuert der Wert an dieser Adresse die Impulsdauer und wird jedes Mal, wenn die CPU von STOP in RUN wechselt, mit dem (oben zugewiesenen) Wert "Anfangsimpulsdauer:" initialisiert. Sie ändern diesen Ausgangswortwert während der Laufzeit, um die Impulsdauer zu ändern. Der Bereich dieses Werts ist vom in der Parametrierung konfigurierten Impulsdauerformat abhängig.

Sie können auch weitere vier Bytes im Speicherbereich A der "Zykluszeit" des PWM-Signals zuordnen. Siehe "Funktionsweise der Impulsausgänge" (Seite 475) mit einem Diagramm des PWM-Signals. Wenn das Kontrollkästchen "Änderung der Zykluszeit zur Laufzeit gestatten" aktiviert ist, speichern die ersten zwei Bytes den Impulsdauerwert und die letzten vier Bytes den Zykluszeitwert.

Während die PWM läuft, können Sie den Wert des Doppelworts am Ende des dieser PWM zugeordneten Speicherbereichs A ändern. Dadurch wird die Zykluszeit des PWM-Signals geändert. Sie aktivieren diese Option beispielsweise, damit die CPU sechs Bytes für PWM1 zuordnet, und Sie wählen dafür AB1008 bis AB1013 aus. Sobald Sie das Programm geladen haben und die PWM starten, können Sie die Impulsdauer über AW1008 und die Zykluszeit über AD1010 ändern.

Wenn die CPU von STOP nach RUN wechselt, initialisiert die CPU den Zykluszeitwert im Speicherbereich A mit dem in Abschnitt "Parametrierung" zugewiesenen Wert für die "Zykluszeit". Die Einheiten und Wertebereiche für den Zykluszeitwert im Speicherbereich A entsprechen der Konfiguration im Abschnitt "Parametrierung".

Wenn Sie das Kontrollkästchen "Änderung der Zykluszeit zur Laufzeit gestatten" aktiviert haben, wählt das TIA Portal automatisch eine neue Adresse für die Ausgangsadresse aus. Die neue Ausgangsadresse kann nicht die gleiche Adresse sein wie die Standardadresse für diesen Impulsgenerator. Das TIA Portal verwendet den nächsten verfügbaren Block mit sechs aufeinander folgenden Bytes. Wird bei der Suche im Speicherbereich A vor dem Ende des Speicherbereichs kein verfügbarer Block gefunden, beginnt die Suche erneut an Adresse "0" im Speicherbereich A, bis ein verfügbarer Block gefunden wird.

Ein für PTO konfigurierter Impulsgenerator verwendet die Ausgangswortadresse nicht.

9.9 Rezepte und Datenprotokolle

9.9.1 Rezepte

9.9.1.1 Übersicht über Rezepte

Speicherung von Rezeptdaten

- Ein Rezeptdatenbaustein, den Sie in Ihrem Projekt erstellen, muss im **Ladespeicher** der CPU abgelegt werden. Hierfür kann der interne CPU-Speicher oder eine externe, als Programmkarte verwendete Memory Card verwendet werden.
- Ein weiterer DB, den Sie erstellen müssen, ist der aktive Rezeptdatenbaustein. Dieser DB muss sich im **Arbeitspeicher** befinden, in dem jeweils ein aktiver Rezeptdatensatz von Ihrer Programmlogik gelesen oder geschrieben wird.

Verwaltung von Rezeptdaten

Der Rezept-DB verwendet ein Array aus Produktrezeptdatensätzen. Jedes Element des Rezept-Arrays stellt eine unterschiedliche Rezeptvariante dar, die auf einem gemeinsamen Satz Komponenten basiert.

- Sie erstellen einen PLC-Datentyp oder einen Datentyp Struct, der alle Komponenten in einem Rezeptdatensatz definiert. Diese Datentypvorlage wird für alle Rezeptdatensätze wiederverwendet. Produktrezepte variieren je nach den Startwerten, die den Rezeptkomponenten zugewiesen sind.
- Eines der Rezepte kann jederzeit mit der Anweisung READ_DBL aus dem Rezept-DB (alle Rezepte im Ladespeicher) in den aktiven Rezept-DB (ein Rezept im Arbeitsspeicher) übertragen werden. Nachdem ein Rezeptdatensatz in den Arbeitsspeicher verschoben wurde, kann Ihre Programmlogik die Komponentenwerte lesen und mit dem Produktionslauf beginnen. Durch die Übertragung wird die Menge des für die Rezeptdaten erforderlichen CPU-Arbeitsspeichers minimiert.
- Werden die Komponentenwerte des aktiven Rezepts während eines Produktionslaufs von einem HMI-Gerät angepasst, können Sie die geänderten Werte mit der Anweisung WRIT_DBL in den Rezept-DB zurückschreiben.

Export von Rezepten (vom Rezept-DB in eine CSV-Datei)

Mit der Anweisung RecipeExport kann der vollständige Satz Rezeptdatensätze in eine CSV-Datei exportiert werden. Auch nicht verwendete Rezeptdatensätze werden dabei exportiert.

Import von Rezepten (aus einer CSV-Datei in einen Rezept-DB)

Nach dem Export der Rezepte können Sie die generierte CSV-Datei als Datenstrukturvorlage verwenden.

1. Eine vorhandene CSV-Rezeptdatei laden Sie über die Dateibrowser-Seite des CPU-Webservers von der CPU in einen PC.
2. Ändern Sie die CSV-Rezeptdatei mit Hilfe eines ASCII-Texteditors. Sie können die Startwerte der Komponenten ändern, jedoch nicht die Datentypen oder Datenstruktur.
3. Laden Sie die geänderte CSV-Datei vom PC wieder zurück in die CPU. Sie müssen dafür jedoch die alte CSV-Datei (mit dem gleichen Namen) im Ladespeicher der CPU löschen oder umbenennen, damit der CPU-Webserver den Ladevorgang durchführen kann.
4. Nachdem Sie die geänderte CSV-Datei in die CPU geladen haben, können Sie die neuen Startwerte mit der Anweisung `RecipelImport` aus der geänderten CSV-Datei (im CPU-Ladespeicher) in den Rezept-DB (im CPU-Ladespeicher) übertragen.

9.9.1.2 Beispiel für ein Rezept

Beispiel für Rezepte

In dieser Tabelle wird gezeigt, wie Sie Rezeptinformationen für die Verwendung in einem Rezept-DB vorbereiten. In diesem Beispiel für einen Rezept-DB sind fünf Datensätze gespeichert, von denen drei verwendet werden. Der vierte und fünfte Datensatz werden zur späteren Erweiterung freigehalten. Jede Zeile in der Tabelle stellt einen Datensatz mit Rezeptname, Komponenten-Datentypen und Komponentenwerten dar.

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

Erstellen eines Rezeptdatenbausteins

Hinweis

Regeln für Rezeptdatenbausteine

- Der Rezept-DB muss ein eindimensionales Array vom PLC-Datentyp oder Datentyp Struct enthalten. Im Rezeptbeispiel wird gezeigt, wie Sie einen Rezept-DB mit PLC-Datentyp erstellen.
- Im Beispiel haben die Komponenteninhaltsstoffe alle den Datentyp UINT. Die Datentypen der Komponenten können auch eine Mischung verschiedener Datentypen außer Struct sein. In einem Array-Element im Rezept-DB ist ein Struct in einem PLC-Datentyp oder ein Struct im Datentyp Struct nicht zulässig.

Erstellen Sie zunächst einen neuen PLC-Datentyp

Fügen Sie einen neuen PLC-Datentyp hinzu, dessen Name den Rezepttyp angibt. In der folgenden Abbildung ist "Beer_Recipe" der neue komplexe PLC-Datentyp, der eine Folge einfacher Datentypen speichert. Der PLC-Datentyp "Beer_Recipe" ist eine Datenvorlage, die in jedem Datensatz des Rezept-DBs und auch im aktiven Rezept-DB wiederverwendet wird. Geben Sie die Komponentennamen und Datentypen ein, die in allen Beispielrezepten gemeinsam vorhanden sind. Die einzelnen Komponentenwerte werden später im Rezept-DB hinzugefügt.

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

Erstellen Sie dann einen Rezeptdatenbaustein

- Erstellen Sie Ihren Rezept-DB als globalen Datenbaustein mit aktivierter DB-Eigenschaft "Nur im Ladespeicher ablegen".
- Der Name des Rezeptdatenbausteins wird als Dateiname der entsprechenden CSV-Datei verwendet. Die im DB-Namen enthaltenen Zeichen müssen sich an die Namensbeschränkungen des Windows-Dateisystems halten. Die Zeichen \ / : * ? " < > | und das Leerzeichen sind nicht zulässig.
- Die Zuweisung des Rezept-Arrays lautet "Products" als Array [1.. 5] of "Beer_Recipe". Die Array-Größe 5 ist die maximal mögliche Anzahl Rezeptvarianten.
- Die Werte für die Rezeptkomponenten werden als DB-Startwerte hinzugefügt.

In der folgenden Abbildung ist das Rezept "BlackBeer" erweitert, um alle Komponenten eines Rezeptdatensatzes zu zeigen.

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"	...	

Export von Rezepten (von Rezept-DB in CSV-Datei)

Durch die Ausführung der Anweisung "RecipeExport (Seite 487)" werden die Daten aus dem Rezept-DB in eine CSV-Datei übertragen (siehe folgende Textdatei).

```
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",0,0,0,0,0,0,0,0,0,0
5 "Not_used",0,0,0,0,0,0,0,0,0,0
```

Import von Rezepten (aus CSV-Datei in Rezept-DB)

1. Eine vorhandene CSV-Rezeptdatei laden Sie über die Dateibrowser-Seite (Seite 893) des-Webservers aus dem CPU-Ladespeicher in einen PC.
2. Ändern Sie die CSV-Rezeptdatei mithilfe eines ASCII-Texteditors. Sie können die Startwerte der Komponenten ändern, jedoch nicht die Datentypen oder Datenstruktur.
3. Laden Sie die geänderte CSV-Datei vom PC wieder zurück in die CPU. Sie müssen dafür jedoch die alte CSV-Datei (mit dem gleichen Namen) im Ladespeicher der CPU löschen oder umbenennen, damit der CPU-Webserver den Ladevorgang durchführen kann.
4. Nachdem Sie die geänderte CSV-Datei in die CPU geladen haben, können Sie die neuen Startwerte mit der Anweisung RecipeImport aus der geänderten CSV-Datei (im CPU-Ladespeicher) in den Rezept-DB (im CPU-Ladespeicher) übertragen.

CSV-Dateien müssen exakt mit der Struktur des entsprechenden Rezept-DBs übereinstimmen

- Die Werte in der CSV-Datei können geändert werden, doch die Struktur darf nicht geändert werden. Für die Anweisung RecipelImport ist es erforderlich, dass die Anzahl der Datensätze und Komponenten exakt der Struktur des Ziel-Rezept-DBs entspricht. Ansonsten schlägt die Ausführung der Anweisung RecipelImport fehl. Beispiel: Sind im Rezept-DB 10 Rezepte definiert, doch werden nur 6 davon verwendet, werden auch die Zeilen 7 bis 10 der CSV-Datei in den DB übertragen. Sie müssen regeln, ob diese Daten gültig sind oder nicht. So können Sie beispielsweise in nicht verwendeten Rezeptdatensätzen dem Produktnamen die Variable "Not_used" zuweisen.
- Wenn Sie der Textdatei Datensätze hinzufügen und die geänderte Datei importieren, müssen Sie darauf achten, dass der von Ihnen zugewiesene Array-Grenzwert des Rezept-DBs genügend Elemente für alle Rezeptdatensätze umfasst.
- Beim Export in die CSV-Datei wird automatisch eine Indexnummer generiert. Wenn Sie zusätzliche Datensätze erstellen, fügen Sie dementsprechend nachfolgende Indexnummern hinzu.
- Bei der Ausführung der Anweisung RecipelImport wird die CSV-Datei auf korrekte Struktur und daraufhin geprüft, ob die Werte in die im zugehörigen Rezept-DB zugewiesenen Datentypen passen. So kann beispielsweise der Datentyp Bool keinen ganzzahligen Wert speichern, und die Ausführung von RecipelImport schlägt in einem solchen Fall fehl.

Anzeige von CSV-Rezeptdaten in Excel

Sie können die CSV-Datei zum leichteren Lesen und Bearbeiten in Excel öffnen. Werden die Kommas nicht als Dezimaltrennzeichen erkannt, verwenden Sie die Importfunktion von Excel, um die Daten strukturiert auszugeben.

	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

Hinweis

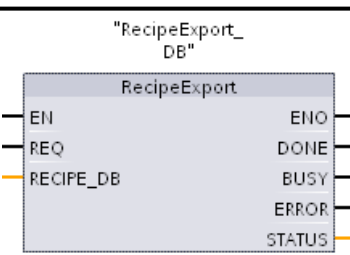
Kommas im Namensfeld des PLC-Datentypenlements

Geben Sie im Namensfeld von in einem Rezept verwendeten PLC-Datentypenlement keine Kommas ein. Wenn Sie Kommas im Namensfeld eingeben, fügt Excel zusätzliche Spalten in die angezeigte CSV-Datei ein. Diese Spalten können zu Fehlern führen, wenn Sie die Startwerte der Rezeptdatensatzdatei bearbeiten.

9.9.1.3 Programmanweisungen zum Übertragen von Rezeptdaten

RecipeExport (Rezeptexport)

Tabelle 9-195 Anweisung RecipeExport

KOP/FUP	SCL	Beschreibung
	<pre>"RecipeExport_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Recipe_DB:=_variant_inout_);</pre>	<p>Die Anweisung "RecipeExport" exportiert alle Rezeptdatensätze aus einem Rezeptdatenbaustein in das CSV-Dateiformat. Die CSV-Datei enthält Produktnamen, Komponentennamen und Startwerte. Die CSV-Datei wird im internen Ladespeicher abgelegt oder auch im externen Ladespeicher, sofern eine optionale als Programmkarte verwendete externe Memory Card installiert ist.</p> <p>Der Exportvorgang wird vom Parameter "REQ" angestoßen. Der Parameter BUSY wird während der Exportverarbeitung auf 1 gesetzt. Nach der Ausführung von RecipeExport wird BUSY auf 0 zurückgesetzt und das Ende des Vorgangs wird am Parameter DONE mit 1 angezeigt. Tritt während der Ausführung ein Fehler auf, zeigen die Parameter ERROR und STATUS das Ergebnis an.</p>

Damit der Export von Rezepten möglich ist, muss ein Rezept-DB erstellt werden. Der Name des Rezeptdatenbausteins wird als Dateiname der neuen CSV-Datei verwendet. Ist bereits eine CSV-Datei mit identischem Namen vorhanden, wird sie während des Exportvorgangs überschrieben.

Sie können über die Dateibrowser-Seite (Seite 893) des integrierten Webservers der CPU auf die CSV-Datei mit den Rezepten zugreifen. Die Datei wird im Rezeptordner im Stammverzeichnis des CPU-Ladespeichers abgelegt.

Tabelle 9-196 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Steuerparameter REQUEST: Aktiviert den Export bei einer steigenden Flanke.
RECIPE_DB	Durchgang	Variant	Pointer auf den Rezeptdatenbaustein. Weitere Informationen finden Sie im Beispiel für einen Rezept-DB (Seite 483). Die im DB-Namen enthaltenen Zeichen müssen sich an die Namenseinschränkungen des Windows-Dateisystems halten. Die Zeichen \ / : * ? " <> und das Leerzeichen sind nicht zulässig.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)
BUSY	OUT	Bool	Ausführung von RecipeExport <ul style="list-style-type: none"> 0: Keine Anweisung in Bearbeitung 1: Anweisung in Bearbeitung

Parameter und Datentyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist. <ul style="list-style-type: none"> • 0: Keine Warnung und kein Fehler • 1: Ein Fehler ist aufgetreten. Der Parameter STATUS enthält Informationen über die Fehlerart.
STATUS	OUT	Word	Ausführungsbedingung

Tabelle 9-197 Werte von ERROR und STATUS

ERROR	STATUS (W#16#....)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter Instanzspeicher ist belegt.
1	8090	Dateiname enthält ungültige Zeichen.
1	8091	Die in RECIPE_DB angegebene Datenstruktur kann nicht verarbeitet werden.
1	8092	Die in RECIPE_DB angegebene Datenstruktur überschreitet 5000 Byte.
1	80B3	Nicht genügend Platz auf MC oder im internen Ladespeicher.
1	80B4	MC ist schreibgeschützt.
1	80B6	Attribut "Nur im Ladespeicher ablegen" des Rezept-DBs ist nicht aktiviert.
1	80C0	CSV-Datei ist kurzzeitig gesperrt.
1	80C1	DB ist kurzzeitig gesperrt.

RecipeImport (Rezeptimport)

Tabelle 9-198 Anweisung RecipeImport

KOP/FUP	SCL	Beschreibung
<p>"RecipeImport_DB"</p> <p>RecipeImport</p> <p>EN ENO</p> <p>REQ DONE</p> <p>RECIPE_DB BUSY</p> <p>ERROR</p> <p>STATUS</p>	<pre>"RecipeImport_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, Recipe_DB:=_variant_inout_);</pre>	<p>Die Anweisung "RecipeImport" importiert Rezeptdaten aus einer CSV-Datei im CPU-Ladespeicher in einen vom Parameter RECIPE_DB angegebenen Rezeptdatenbaustein. Während des Importvorgangs werden die Startwerte im Rezeptdatenbaustein überschrieben. Der Importvorgang wird vom Parameter "REQ" angestoßen. Der Parameter BUSY wird während der Importverarbeitung auf 1 gesetzt. Nach der Ausführung von RecipeImport wird BUSY auf 0 zurückgesetzt und das Ende des Vorgangs wird am Parameter DONE mit 1 angezeigt. Tritt während der Ausführung ein Fehler auf, zeigen die Parameter ERROR und STATUS das Ergebnis an.</p>

Tabelle 9-199 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Steuerparameter REQUEST: Aktiviert den Import bei einer steigenden Flanke.
RECIPE_DB	Durchgang	Variant	Pointer auf den Rezeptdatenbaustein. Weitere Informationen finden Sie im "Beispiel für einen Rezept-DB (Seite 483)". Die im DB-Namen enthaltenen Zeichen müssen sich an die Namensbeschränkungen des Windows-Dateisystems halten. Die Zeichen \ / : * ? " < > und das Leerzeichen sind nicht zulässig.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 - Keine Anweisung in Bearbeitung 1 - Anweisung in Bearbeitung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Damit ein Importvorgang durchgeführt werden kann, muss ein Rezept-DB mit einer Struktur, die der Datenstruktur der CSV-Datei entspricht, vorhanden sein.

Regeln für CSV-Dateien:

- Die CSV-Datei muss sich im Stammverzeichnis "Recipes" im internen Ladespeicher oder auch im externen Ladespeicher, sofern eine optionale als Programmkarte verwendete externe Memory Card installiert ist, befinden.
- Der Name der CSV-Datei muss dem Namen des Datenbausteins im Parameter RECIPE_DB entsprechen.
- Die erste Zeile (Überschrift) der CSV-Datei enthält den Namen der Rezeptkomponenten. Die erste Zeile wird beim Import ignoriert. Die Namen der Rezeptkomponenten in der CSV-Datei und im Datenbaustein werden während des Importvorgangs nicht aufeinander abgestimmt.
- In jedem Fall ist der erste Wert in jeder Zeile der CSV-Datei die Indexnummer des Rezepts. Die einzelnen Rezepte werden in der Reihenfolge des Index importiert. Hierfür muss sich der Index in der CSV-Datei in aufsteigender Reihenfolge befinden und darf keine Lücken enthalten (ist dies nicht der Fall, wird im Parameter STATUS die Fehlermeldung 80B0 ausgegeben).
- Die CSV-Datei darf nicht mehr Rezeptdatensätze enthalten als im Rezeptdatenbaustein vorgesehen. Die maximale Anzahl Datensätze wird von den Array-Grenzwerten im Datenbaustein angegeben.

Tabelle 9-200 Werte von ERROR und STATUS

ERROR	STATUS (W#16#....)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter Instanzspeicher ist belegt.
1	8090	Der Dateiname enthält ungültige Zeichen.
1	8092	Keine passende CSV-Datei zum Importieren gefunden. Mögliche Ursache: Der Name der CSV-Datei entspricht nicht dem Namen des Rezept-DBs.
1	80C0	CSV-Datei ist kurzzeitig gesperrt.
1	80C1	Datenbaustein ist kurzzeitig gesperrt.
1	80B0	Die Nummerierung im Index der CSV-Datei ist nicht kontinuierlich, nicht in aufsteigender Reihenfolge oder überschreitet die maximale Anzahl (Array-Grenzwert) im Datenbaustein.
1	80B1	Struktur des Rezeptdatenbausteins und der CSV-Datei stimmen nicht überein: Die CSV-Datei enthält zu viele Felder.
1	80B2	Struktur des Rezeptdatenbausteins und der CSV-Datei stimmen nicht überein: Die CSV-Datei enthält zu wenige Felder.
1	80B6	Attribut "Nur im Ladespeicher ablegen" des Rezept-DBs ist nicht aktiviert.
1	80D0 +n	Struktur des Rezeptdatenbausteins und der CSV-Datei stimmen nicht überein: Datentyp im Feld n stimmt nicht überein (n <= 46).
1	80FF	Struktur des Rezeptdatenbausteins und der CSV-Datei stimmen nicht überein: Datentyp im Feld n stimmt nicht überein (n > 46).

9.9.1.4 Beispielprogramm für Rezepte

Voraussetzungen für das Beispielprogramm für Rezepte

Für das Beispielprogramm für Rezepte müssen die folgenden Voraussetzungen erfüllt sein:

- Ein Rezept-DB mit allen Rezeptdatensätzen. Der Rezept-DB ist im Ladespeicher abgelegt.
- Ein aktiver Rezept-DB mit einer Kopie eines Rezepts im Arbeitsspeicher.

Weitere Informationen zum Rezept-DB und der entsprechenden CSV-Datei finden Sie unter "Beispiel für einen Rezept-DB (Seite 483)".

Erstellen Sie den aktiven Rezept-DB

Im Fenster "Neuen Baustein hinzufügen":

- Wählen Sie im Fenster "Neuen Baustein hinzufügen" die Schaltfläche "Datenbaustein".
- Wählen Sie in der Klappliste "Typ" den zuvor erstellten PLC-Datentyp "Beer_recipe" aus.

Startwerte sind nicht erforderlich. Wenn ein Rezept aus dem Rezept-DB in den aktiven Rezept-DB übertragen wird, werden die DB-Datenwerte festgelegt. In dem Beispiel ist der aktive Rezept-DB das Ziel für die Daten der Anweisung READ_DBL und bietet der Anweisung WRITE_DBL Quelldaten. Die folgende Abbildung zeigt den DB "Active_Recipe".

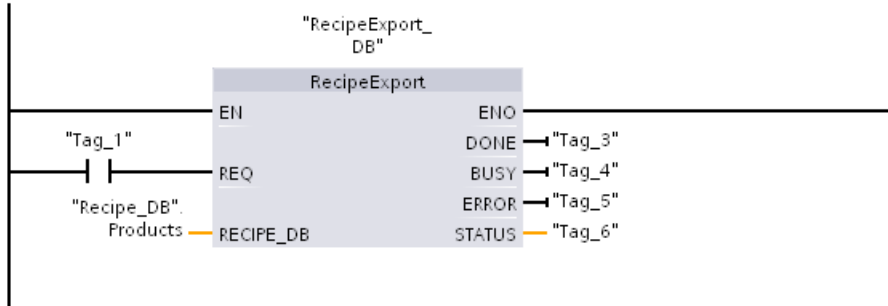
Active_Recipe			
	Name	Data type	Start value
1	▼ Static		
2	productname	String[20]	'Beer_Recipe'
3	water	UInt	0
4	barley	UInt	0
5	wheat	UInt	0
6	hops	UInt	0
7	yeast	UInt	0
8	waterTmp	UInt	0
9	mashTmp	UInt	0
10	mashTime	UInt	0
11	QTest	UInt	0

Instanz-DBs

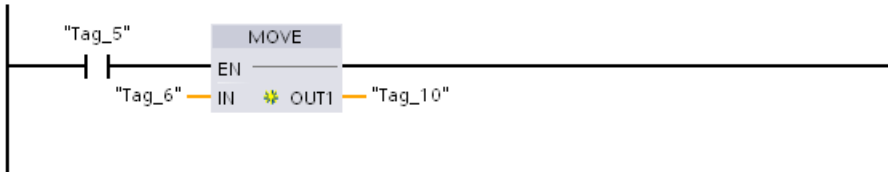
Die von den Anweisungen RecipeExport ("RecipeExport_DB") und RecipelImport ("RecipelImport_DB") verwendeten Instanz-DBs werden automatisch erstellt, wenn Sie die Anweisungen in Ihr Programm einfügen. Die Instanz-DBs dienen zum Steuern der Anweisungsausführung und werden in der Programmlogik nicht referenziert.

Beispielprogramm für Rezepte

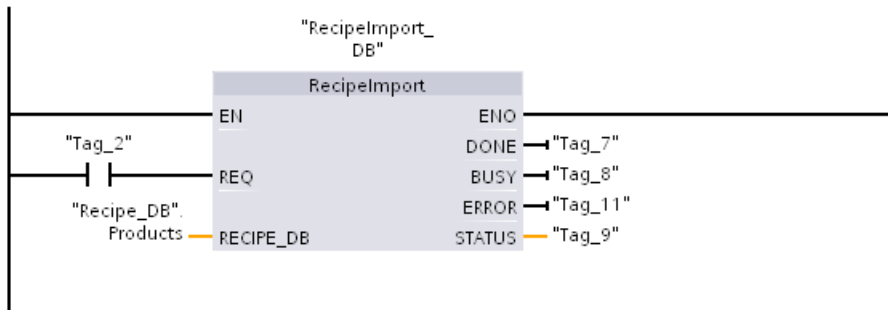
Netzwerk 1 Eine steigende Flanke an REQ startet den Exportvorgang. Aus den Daten des Rezept-DBs wird eine CSV-Datei generiert und im Rezeptordner im CPU-Speicher abgelegt.



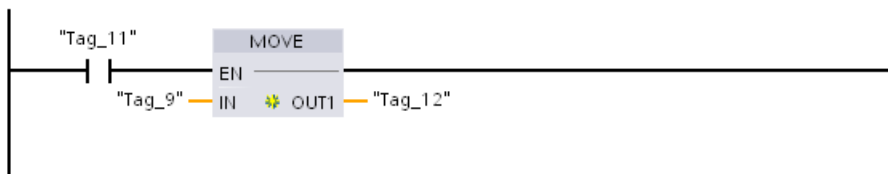
Netzwerk 2 Erfassen Sie den Ausgang STATUS der Ausführung von RecipeExport, weil er nur einen Zyklus lang gültig ist.



Netzwerk 3 Eine steigende Flanke an REQ startet den Importvorgang. In den vorhandenen Rezept-DB werden alle aus der entsprechenden CSV-Datei, die sich im Rezeptordner des CPU-Speichers befindet, gelesenen Rezeptdaten geladen.

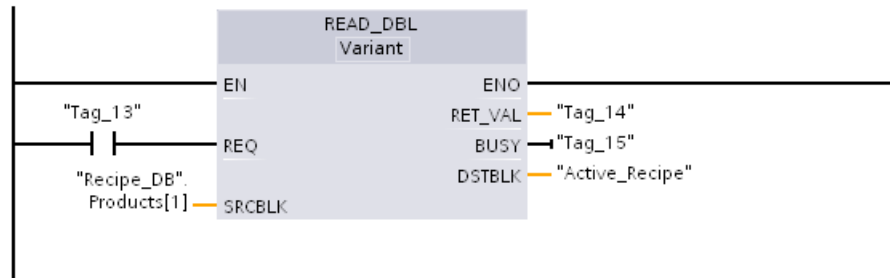


Netzwerk 4 Erfassen Sie den Ausgang STATUS der Ausführung von RecipeImport, weil er nur einen Zyklus lang gültig ist.

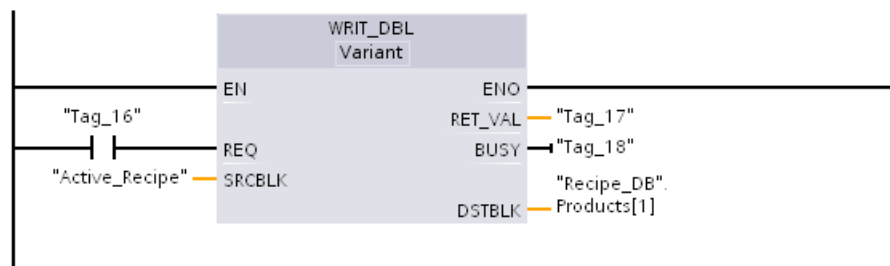


Netzwerk 5 READ_DBL kopiert die Startwerte von einem Rezept "Recipe_DB". Products[1] (im CPU-Ladespeicher) in die aktuellen Werte des DBs "Active_Recipe" (im CPU-Arbeitsspeicher). Nach der Ausführung von READ_DBL kann Ihre Programmlogik über die Adressen im Speicher des DBs "Active_Recipe" auf die Komponentenwerte des Rezepts zugreifen. So geben

beispielsweise die symbolischen Adressen ("Active_Recipe".productname) und ("Active_Recipe".water) Ihrer Programmlogik den aktuellen Rezeptnamen und die Menge des Wassers an.



Netzwerk 6 Während der Laufzeit könnte ein HMI-Gerät einen im DB "Active_Recipe" gespeicherten Komponentenwert ändern. Verbesserte Rezeptdaten können durch Ausführung von WRIT_DBL gespeichert werden. In dem Beispiel werden alle Startwerte des Rezept-DBs für das einzelne Rezept "Recipe_DB". Products[1] durch die aktuellen Werte aus dem DB "Active_Recipe" überschrieben.



9.9.2 Datenprotokolle

Ihr Steuerungsprogramm kann mit den Anweisungen Data log Laufzeitdatenwerte in beständigen Protokolldateien speichern. Die CPU speichert Datenprotokolldateien im Flash-Speicher (CPU oder Memory Card) im Standard-CSV-Format (durch Komma getrennte Werte). Die CPU organisiert die Datensätze in einer kreisförmigen Protokolldatei vordefinierter Größe.

Mit den Anweisungen Data log können Sie in Ihrem Programm Protokolldateien anlegen, öffnen, schließen und einen Datensatz in die Dateien schreiben. Sie entscheiden, welche Programmwerte protokolliert werden, indem Sie einen Datenpuffer anlegen, der einen einzigen Protokolldatensatz definiert. Die CPU nutzt Ihren Datenpuffer als temporären Speicher für einen neuen Protokolldatensatz. Ihr Steuerungsprogramm verschiebt während der Laufzeit neue aktuelle Werte in den Puffer. Wenn das Programm alle aktuellen Datenwerte aktualisiert hat, kann es die Anweisung DataLogWrite ausführen, um Daten aus dem Puffer in einen Protokolldatensatz zu übertragen.

Datenprotokolldateien können Sie auf der Dateibrowser-Seite des Webservers öffnen, bearbeiten, speichern, umbenennen und löschen. Um den Dateibrowser anzuzeigen, benötigen Sie Leserechte, und um Datenprotokolldateien zu bearbeiten, zu löschen oder umzubenennen, benötigen Sie Änderungsrechte.

9.9.2.1 Datensatzstruktur der Datenprotokolle

Die Parameter DATA und HEADER der Anweisung DataLogCreate weisen den Datentyp und die Spaltenkopfbeschreibung aller Datenelemente in einem Protokolldatensatz zu.

Parameter DATA für die Anweisung DataLogCreate

Der Parameter DATA verweist auf Speicher, der als temporärer Puffer für einen neuen Protokolldatensatz verwendet wird. Ihm muss eine M- oder DB-Adresse zugewiesen sein.

Zuweisen können Sie einen kompletten DB (abgeleitet von einem PLC-Datentyp, den Sie bei der Erstellung des DBs angeben) oder einen Teil eines DBs (bei dem angegebenen DB-Element kann es sich um einen beliebigen Datentyp, eine Datentypstruktur, einen PLC-Datentyp oder ein Daten-Array handeln).

Bei der Anweisung DataLogCreate sind die Strukturdatentypen auf eine einzige Schachtelungsebene begrenzt. Ein Array aus Zeichenketten gilt in diesem Kontext nicht als einzelne Schachtelungsebene. Derzeit gibt die Anweisung DataLogCreate keinen Fehler zurück. Die Anweisung verarbeitet nur die erste Zeichenkette des Arrays. Die Gesamtzahl der deklarierten Datenelemente muss der Anzahl der im Parameter HEADER angegebenen Spalten entsprechen. Sie können maximal 253 Datenelemente (mit Zeitstempel) bzw. 255 Datenelemente (ohne Zeitstempel) zuweisen. Durch diese Einschränkung bleibt Ihr Datensatz innerhalb des Grenzwerts von maximal 256 Spalten eines Excel-Arbeitsblatts.

Der Parameter DATA kann entweder remanente oder nicht remanente Datenelemente in einem DB vom Typ "Standard" (kompatibel mit S7-300/400) oder "Optimiert" zuweisen.

Um einen Protokolldatensatz zu schreiben, müssen Sie zunächst neue Prozesswerte in den temporären Datensatz DATA laden und dann die Anweisung DataLogWrite ausführen, wodurch neue Datensatzwerte in der Datenprotokolldatei gespeichert werden.

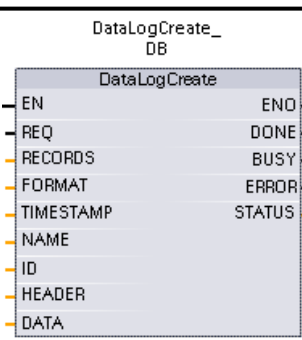
Parameter HEADER für die Anweisung DataLogCreate

Der Parameter HEADER zeigt auf die Spaltenköpfe in der obersten Zeile der Datenmatrix in der CSV-Datei. HEADER-Daten müssen sich im DB- oder M-Speicher befinden und die Zeichen müssen den üblichen Formatregeln für CSV-Dateien entsprechen, wobei die einzelnen Spaltennamen durch Komma zu trennen sind. Bei dem Datentyp kann es sich um Strings, Byte-Arrays oder Zeichen-Arrays handeln. Zeichen- bzw. Byte-Arrays ermöglichen eine größere Größe, während der Datentyp String auf maximal 255 Bytes begrenzt ist. Der Parameter HEADER ist optional. Wenn der HEADER nicht zugewiesen ist, wird keine Kopfzeile in der Datenprotokolldatei angelegt.

9.9.2.2 Programmanweisungen zum Steuern von Datenprotokollen

DataLogCreate (Datenprotokoll erstellen)

Tabelle 9-201 Anweisung DataLogCreate

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogCreate_DB" (req:=_bool_in_, records:=_uint_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_);</pre>	<p>Erstellt und initialisiert eine Datenprotokolldatei. Die CPU erstellt die Datei im Ordner \DataLogs, verwendet den Namen im Parameter NAME und öffnet die Datei implizit für Schreibvorgänge. Mit den Anweisungen Data log können Sie in Ihrem Programm Laufzeit-Prozessdaten im Flash-Speicher der CPU oder auf der Memory Card speichern.</p> <p>STEP 7 erstellt automatisch den zugehörigen Instanz-DB, wenn Sie die Anweisung einfügen.</p>

¹ Im SCL-Beispiel ist "DataLogCreate_DB" der Name des Instanz-DBs.

Tabelle 9-202 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
RECORDS	IN	UDint
FORMAT	IN	UInt
TIMESTAMP	IN	UInt

Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)

Die maximale Anzahl Datensätze, die das kreisförmige Datenprotokoll enthalten kann, bevor der älteste Eintrag überschrieben wird:

Der Datensatz mit der Kopfzeile ist in dieser Anzahl nicht enthalten. Es muss ausreichend PLC-Ladespeicher verfügbar sein, damit das Datenprotokoll erfolgreich angelegt wird. (Voreinstellung - 1)

Format des Datenprotokolls:

- 0 - Internes Format (nicht unterstützt)
- 1 - Durch Komma getrennte Werte, "csv-eng" (Standardwert)

Format des Zeitstempels der Daten: Spaltentitel für Datum- und Uhrzeitfelder sind optional. Der Zeitstempel kann entweder die Systemzeit (Coordinated Universal Time - UTC) oder die Ortszeit anzeigen.

- 0 - Kein Zeitstempel
- 1 - Systemzeit mm/tt/jjjj, hh:mm:ss
- 2 - Ortszeit mm/tt/jjjj, hh:mm:ss
- 3 - Systemzeit mm/tt/jjjj, hh:mm:ss
- 4 - Ortszeit jjjj-mm--tt, hh:mm:ss
- 5 - Systemzeit jjjj-mm-tt, hh:mm:ss

Parameter und Datentyp		Datentyp	Beschreibung
NAME	IN	Variant	<p>Name des Datenprotokolls: Sie geben einen Namen ein. Diese Variante unterstützt nur den Datentyp String und kann nur im DB- oder im lokalen Speicher abgelegt werden. (Standardwert: '')</p> <p>Diese Zeichenkette ist der Name der Datenprotokolldatei. Verwenden Sie Zeichen aus dem ASCII-Zeichensatz, die Zeichen \ / : * ? " < > und das Leerzeichen sind nicht zulässig.</p>
ID	Durchgang	DWord	<p>Numerische Kennung des Datenprotokolls: Sie speichern diesen generierten Wert zur Verwendung mit anderen Datenprotokollanweisungen. Der Parameter ID wird bei der Anweisung DataLogCreate nur als Ausgang verwendet. (Standardwert: 0)</p> <p>Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.</p>
HEADER	Durchgang	Variant	<p>Pointer auf die Spaltenköpfe des Datenprotokolls in der obersten Zeile der Datenmatrix in der CSV-Datei. (Standardwert: null).</p> <p>HEADER-Daten müssen sich im DB- oder M-Speicher befinden.</p> <p>Die Zeichen müssen den üblichen Formatregeln für CSV-Dateien entsprechen, und die einzelnen Spaltennamen sind durch Komma zu trennen. Bei dem Datentyp kann es sich um Strings, Byte-Arrays oder Zeichen-Arrays handeln. Zeichen- bzw. Byte-Arrays ermöglichen eine größere Größe, während der Datentyp String auf maximal 255 Bytes begrenzt ist.</p> <p>Der Parameter HEADER ist optional. Wenn der HEADER nicht parametrisiert ist, wird keine Kopfzeile in der Datenprotokolldatei angelegt.</p>
DATA	Durchgang	Variant	<p>Pointer auf die Datenstruktur des Datensatzes, anwenderdefinierter Datentyp (UDT) oder Array. Datensatzdaten müssen sich im DB- oder M-Speicher befinden.</p> <p>Der Parameter DATA gibt die einzelnen Datenelemente (Spalten) eines Protokolldatensatzes und deren Datentyp an. Bei der Anweisung DataLogCreate sind die Strukturdatentypen auf eine einzige Schachtelungsebene begrenzt. Ein Array aus Zeichenketten gilt in diesem Kontext nicht als einzelne Schachtelungsebene. Derzeit gibt die Anweisung DataLogCreate keinen Fehler zurück. Die Anweisung verarbeitet nur die erste Zeichenkette des Arrays. Die Anzahl der deklarierten Datenelemente muss der Anzahl der im Parameter HEADER angegebenen Spalten entsprechen. Sie können maximal 253 Datenelemente (mit Zeitstempel) bzw. 255 Datenelemente (ohne Zeitstempel) zuweisen. Durch diese Einschränkung bleibt Ihr Datensatz innerhalb des Grenzwerts von maximal 256 Spalten eines Excel-Arbeitsblatts.</p>
DONE	OUT	Bool	<p>Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)</p>

Parameter und Datentyp		Datentyp	Beschreibung
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 - Keine Anweisung in Bearbeitung • 1 - Anweisung in Bearbeitung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Die CPU erstellt basierend auf den Parametern RECORDS und DATA eine Datenprotokolldatei mit einer vordefinierten festen Größe und organisiert die Datensätze in einer kreisförmigen Protokolldatei. Die Anweisung DataLogCreate ordnet einen permanenten CPU-Speicher für das gesamte Datenprotokoll zu, wenn die Anweisung DONE = WAHR zurückgibt. Der erforderliche Speicherplatz in der CPU ist auf Grund des Dateisystemmanagements und verbundener Werte größer als die Datei. Der permanente Speicher für das Datenprotokoll bleibt zugewiesen, bis die CPU den Speicher auf eine der folgenden Weisen wieder freigibt:

- Das Anwenderprogramm ruft Anweisung DataLogDelete auf.
- Ein Webserver-Anwender löscht das Datenprotokoll im Webserver.
- Ein SIMATIC Automation Tool-Anwender löscht das Datenprotokoll im SIMATIC Automation Tool.

Wird die Datenprotokolldatei auf andere Weise gelöscht, zum Beispiel mit Hilfe eines Kartenlesers, wird der permanente Speicher für das Datenprotokoll in der CPU nicht freigegeben.

Die Anweisung DataLogWrite fügt neue Datensätze in die Datenprotokolldatei ein, bis die maximale Anzahl Datensätze, die im Parameter RECORDS vorgegeben ist, gespeichert ist. Danach überschreibt der nächste Datensatz, der geschrieben wird, den ältesten Datensatz. Eine weitere Anweisung DataLogWrite überschreibt den zweitältesten Datensatz usw.

Speicherauslastung:

- Die Datenprotokolle belegen nur Ladespeicher.
- Die Größe aller Datenprotokolle insgesamt ist durch die verfügbare Kapazität des Ladespeichers begrenzt. Maximal acht Protokolle können gleichzeitig geöffnet sein. Sie können Ihre Datenprotokolle über die Dateibrowser (Seite 893) Standard-Webseite verwalten. Richtlinien dazu, wie viele Datenprotokolle gleichzeitig gespeichert werden können, finden Sie in der Beschreibung dieser Standard-Webseite.
- Die maximal mögliche Anzahl für den Parameter RECORDS ist der Grenzwert einer UDint-Zahl (4.294.967.295). Der tatsächliche Grenzwert für den Parameter RECORD richtet sich nach der Größe eines einzelnen Datensatzes, der Größe anderer Datenprotokolle und der verfügbaren Kapazität des Ladespeichers. Zudem ist in Excel die Anzahl der Zeilen in einem Excel-Arbeitsblatt begrenzt.

Hinweis**Die Datenprotokollerstellung muss abgeschlossen sein, bevor eine Anweisung zum Schreiben des Datenprotokolls angestoßen werden kann**

- Die Operationen DataLogCreate und DataLogNewFile zum Erstellen von Datenprotokollen erstrecken sich über viele Programmzyklen. Die tatsächliche für die Erstellung der Protokolldatei benötigte Zeit hängt von der Datensatzstruktur und der Anzahl der Datensätze ab. Ihre Programmlogik muss den Wechsel des DONE-Bits nach WAHR, der die abgeschlossene Erstellung einer Protokolldatei meldet, überwachen und erfassen. Führt das Anwenderprogramm eine Anweisung DataLogWrite vor Abschluss der Datenprotokollerstellung aus, kann die Schreiboperation den neuen Protokolldatensatz nicht wie erwartet schreiben.
- In bestimmten Situationen, wenn ein sehr schneller Zyklus abläuft, kann die Erstellung des Datenprotokolls länger dauern. Wenn die langsame Erstellung zu lange dauert, müssen Sie sicherstellen, dass das Kontrollkästchen zum Aktivieren der Mindestzykluszeit für zyklische OBs aktiviert und die Mindestzykluszeit auf 1 ms oder mehr gesetzt ist. Weitere Informationen finden Sie unter Konfigurieren von Zykluszeit und Kommunikationslast (Seite 90).

Hinweis**Die Anweisung DataLogNewFile kopiert die Datensatzstruktur eines vorhandenen Datenprotokolls**

Wenn Sie das Überschreiben von Datensätzen verhindern möchten, können Sie mit der Anweisung DataLogNewFile ein neues Datenprotokoll basierend auf dem aktuellen Datenprotokoll anlegen, nachdem das aktuelle Datenprotokoll die maximale Anzahl von Datensätzen gespeichert hat. Neue Datensätze werden dann in der neuen Datenprotokolldatei gespeichert. Die alte Datenprotokolldatei und die Datensatzdaten bleiben im Flash-Speicher erhalten.

Tabelle 9-203 Werte von ERROR und STATUS

ERROR	STATUS (W#16#....)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	807F	Interner Fehler
1	8090	Ungültiger Dateiname
1	8091	Der Namensparameter ist keine Zeichenkette.
1	8093	Es ist bereits ein Datenprotokoll mit dem Namen vorhanden. Verwenden Sie einen anderen Namen, vergewissern Sie sich, dass die CSV-Datei des Datenprotokolls nicht geöffnet ist und löschen Sie dann das vorhandene Datenprotokoll über die Dateibrowser (Seite 893)-Seite des Webservers.
1	8097	Gewünschte Dateilänge überschreitet maximale Dateigröße für das Dateisystem.

ERROR	STATUS (W#16#....)	Beschreibung
1	80B2	Keine weiteren Ressourcen-IDs vorhanden Hinweis: Um diesen Fehler zu vermeiden, löschen Sie einige vorhandene Datenprotokolle oder verringern die Anzahl der Spalten in der Datensatzstruktur.
1	80B3	Nicht genügend Ladespeicher.
1	80B4	Die Memory Card (MC) ist schreibgeschützt.
1	80C0	Archivdatei ist gesperrt
1	80C1	Zu viele geöffnete Dateien: Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein.
1	8253	Ungültige Anzahl Datensätze
1	8353	Ungültige Formatauswahl
1	8453	Ungültige Zeitstempelauswahl
1	8B24	Ungültige HEADER-Bereichszuweisung: Beispiel: Verweist auf lokalen Speicher
1	8B51	Ungültiger Datentyp für Parameter HEADER
1	8B52	Zu viele Datenelemente im Parameter HEADER
1	8C24	Ungültige DATA-Bereichszuweisung: Beispiel: Zeigt auf lokalen Speicher
1	8C51	Ungültiger Datentyp für Parameter DATA
1	8C52	Zu viele Datenelemente im Parameter DATA

DataLogOpen (Datenprotokoll öffnen)

Tabelle 9-204 Anweisung DataLogOpen

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Öffnet eine bereits vorhandene Datenprotokolldatei. Sie müssen ein Datenprotokoll öffnen, damit Sie neue Datensätze in das Protokoll schreiben (Seite 501) können. Sie können Datenprotokolle einzeln öffnen und schließen. Maximal acht Datenprotokolle können gleichzeitig geöffnet sein.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

² Im SCL-Beispiel ist "DataLogOpen_DB" der Name des Instanz-DBs.

Tabelle 9-205 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
MODE	IN	UInt

Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)

Betriebsart:

- 0 - An vorhandene Daten anhängen (Standardwert)
- 1 - Alle vorhandenen Datensätze löschen

Parameter und Datentyp		Datentyp	Beschreibung
NAME	IN	Variant	Name eines vorhandenen Datenprotokolls: Diese Variante unterstützt nur den Datentyp String und kann nur im lokalen Speicher, im DB- oder M-Speicher abgelegt werden. (Standardwert: '')
ID	Durchgang	DWord	Numerische Kennung eines Datenprotokolls. (Standardwert: 0) Hinweis: Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Standardwert: Falsch)
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 - Keine Anweisung in Bearbeitung 1 - Vorgang in Bearbeitung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Sie können entweder den NAMEN oder eine ID (Parameter ID als Eingang) eines bereits vorhandenen Datenprotokolls eingeben. Wenn Sie beide Parameter angeben, und eine gültige ID entspricht dem Datenprotokoll NAME, wird die ID verwendet und der NAME wird ignoriert.

Der NAME muss dem Namen eines mit der Anweisung DataLogCreate angelegten Datenprotokolls entsprechen. Wird nur der NAME angegeben, und der NAME verweist auf ein gültiges Datenprotokoll, wird die entsprechende ID ausgegeben (Parameter ID als Ausgang).

Hinweis

Allgemeine Verwendung von Datenprotokolldateien

- Nach Ausführung der Anweisungen DataLogCreate und DataLogNewFile werden automatisch Datenprotokolldateien geöffnet.
- Nach einem Wechsel des PLC-Geräts von RUN in STOP oder nach einem Neustart des PLC-Geräts werden Datenprotokolldateien automatisch geschlossen.
- Damit eine neue Anweisung DataLogWrite durchgeführt werden kann, muss eine Datenprotokolldatei geöffnet sein.
- Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein. Mehr als acht Datenprotokolldateien dürfen vorhanden sein, von diesen müssen jedoch einige geschlossen werden, so dass maximal acht gleichzeitig geöffnet sind.

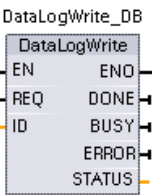
Tabelle 9-206 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0002	Warnung: Datenprotokolldatei wurde von diesem Anwendungsprogramm bereits geöffnet
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.

ERROR	STATUS (W#16#)	Beschreibung
1	8090	Datenprotokolldefinition stimmt nicht mit der vorhandenen Datenprotokoll-datei überein.
1	8091	Der Namensparameter ist keine Zeichenkette.
1	8092	Datenprotokoll ist nicht vorhanden.
1	80C0	Datenprotokolldatei ist gesperrt.
1	80C1	Zu viele geöffnete Dateien: Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein.

DataLogWrite (Datenprotokoll schreiben)

Tabelle 9-207 Anweisung DataLogWrite

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogWrite_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>Schreibt einen Datensatz in das angegebene Datenprotokoll. Das bereits bestehende Zieldatenprotokoll muss geöffnet (Seite 499) sein, damit Sie mit einer Anweisung DataLogWrite in das Protokoll schreiben können.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

² Im SCL-Beispiel ist "DataLogWrite_DB" der Name des Instanz-DBs.

Tabelle 9-208 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
ID	Durchgang	DWord Numerische Kennung des Datenprotokolls: Wird bei der Anweisung DataLogWrite nur als Eingang verwendet. (Standardwert: 0) Hinweis: Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> 0 - Keine Anweisung in Bearbeitung 1 - Anweisung in Bearbeitung
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung (Standardwert: 0)

Der Parameter DATA einer Anweisung DataLogCreate definiert die Speicheradresse und die Datenstruktur des Datensatzpuffers. Um neue Datensatzdaten aus dem Puffer in ein Datenprotokoll zu übertragen, muss das Steuerungsprogramm den Datensatzpuffer mit aktuellen Laufzeitprozesswerten laden und dann die Anweisung DataLogWrite ausführen.

Der Parameter ID gibt eine Datenprotokoll- und Datensatzkonfiguration an. Die Anweisung DataLogCreate erzeugt die Zahl ID.

Enthält die kreisförmige Datenprotokolldatei leere Datensätze, schreibt die Anweisung DataLogWrite in den nächsten freien Datensatz. Sind alle Datensätze belegt, überschreibt die Anweisung DataLogWrite den ältesten Datensatz.

ACHTUNG**Die Datenprotokollerstellung muss abgeschlossen sein, bevor eine Anweisung zum Schreiben des Datenprotokolls angestoßen werden kann**

Die Operationen DataLogCreate und DataLogNewFile zum Erstellen von Datenprotokollen erstrecken sich über viele Programmzyklen. Die tatsächliche für die Erstellung der Protokolldatei benötigte Zeit hängt von der Datensatzstruktur und der Anzahl der Datensätze ab. Ihre Programmlogik muss den Wechsel des DONE-Bits nach WAHR, der die abgeschlossene Erstellung einer Protokolldatei meldet, überwachen und erfassen. Wird eine Anweisung DataLogWrite vor Abschluss der Datenprotokollerstellung ausgeführt, schreibt die Schreiboperation keinen neuen Protokolldatensatz.

Hinweis**Auswirkung von Datenprotokollen auf den internen CPU-Speicher**

Jeder Schreibvorgang eines Datenprotokolls verbraucht mindestens 2 KB Speicher. Wenn Ihr Programm häufig kleinere Mengen von Daten schreibt, werden bei jedem Schreibvorgang mindestens 2 KB Speicher verbraucht. Eine bessere Umsetzung wäre die Ansammlung kleiner Datenelemente in einem Datenbaustein (DB), der dann weniger häufig ins Datenprotokoll geschrieben würde.

Wenn Ihr Programm sehr häufig viele Datenprotokolleinträge schreibt, können Sie auch eine austauschbare SD Memory Card verwenden.

ACHTUNG**Möglicher Protokolldatenverlust während eines CPU-Spannungsausfalls**

Tritt während einer noch nicht beendeten Anweisung DataLogWrite ein Spannungsausfall auf, geht der ins Datenprotokoll übertragene Datensatz möglicherweise verloren.

Tabelle 9-209 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0001	Zeigt an, dass das Datenprotokoll voll ist: Jedes Datenprotokoll wird mit einer vorgegebenen Maximalanzahl von Datensätzen angelegt. Der letzte Datensatz dieser Maximalanzahl wurde geschrieben. Die nächste Schreibenweisung überschreibt den ältesten Datensatz.
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	8092	Datenprotokoll ist nicht vorhanden.
1	80B0	Datenprotokolldatei ist nicht geöffnet (nur beim Modus mit explizitem Öffnen).

DataLogClear (Datenprotokoll leeren)

Beschreibung

Tabelle 9-210 Anweisung DataLogClear

KOP/FUP	SCL	Beschreibung
<pre> "DataLogClear_ DB" DataLogClear - EN ENO - - REQ DONE - - ID BUSY - ERROR - STATUS - </pre>	<pre> "DataLogClear_DB" (REQ:=_bool_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, ID:=_dword_inout_); </pre>	<p>Die Anweisung "DataLogClear" löscht alle Datensätze in einem vorhandenen Datenprotokoll. Die Anweisung löscht nicht den optionalen Header der CSV-Datei (siehe Beschreibung des Parameters HEADER der Anweisung "DataLogCreate (Seite 495)").</p> <p>Mit dem Parameter ID wählen Sie das Datenprotokoll aus, dessen Datensätze gelöscht werden sollen.</p>

"DataLogClear_DB" ist der Name des Instanz-DBs.

Voraussetzung

Damit Sie Datensätze löschen können, muss das Datenprotokoll offen sein. Verwenden Sie die Anweisung DataLogOpen (Seite 499), um ein Datenprotokoll zu öffnen.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "DataLogClear":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L, T, C oder Konstante (T und C sind nur bei der S7-1500 in KOP und FUP verfügbar)	Ausführung der Anweisung bei einer steigenden Flanke
ID	InOut	DWORD	E, A, M, D, L	Numerische Kennung des Datenprotokolls
DONE	Output	BOOL	E, A, M, D, L	Die Anweisung wurde erfolgreich ausgeführt.
BUSY	Output	BOOL	E, A, M, D, L	Die Ausführung der Anweisung ist noch nicht beendet.
ERROR	Output	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> 0: Kein Fehler. 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen werden am Parameter STATUS ausgegeben.
STATUS	Output	WORD	E, A, M, D, L	Statusparameter Der Parameter wird nur für die Dauer eines Aufrufs gesetzt. Um den Status anzuzeigen, sollten Sie deshalb den Parameter STATUS in einen freien Datenbereich kopieren.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

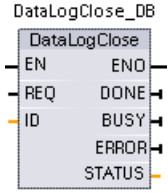
Parameter STATUS

Fehlercode* (W#16#...)	Bedeutung
0000	Kein Fehler.
7000	Keine laufende Auftragsbearbeitung.
7001	Start der Auftragsbearbeitung. Parameter BUSY = 1, DONE = 0
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; BUSY hat den Wert 1.
8080	Die mit dem Parameter ID ausgewählte Datenprotokolldatei kann mit der Anweisung "DataLogClear" nicht verarbeitet werden.
8092	Datenprotokoll ist nicht vorhanden.
80A2	Vom Dateisystem zurückgemeldeter Schreibfehler.
80B0	Datenprotokoll ist nicht geöffnet.
80B4	Die Memory Card ist schreibgeschützt.

* Die Fehlercodes können als ganzzahlige Werte oder als Hexadezimalwerte im Programmierer angezeigt werden. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".

DataLogClose (Datenprotokoll schließen)

Tabelle 9-211 Anweisung DataLogClose

KOP/FUP	SCL	Beschreibung
	<pre>"DataLogClose_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>Schließt eine geöffnete Datenprotokolldatei. Eine für ein geschlossenes Datenprotokoll ausgeführte Anweisung DataLogWrite führt zu einem Fehler. Schreibenweisungen für dieses Datenprotokoll sind erst zulässig, nachdem eine Anweisung DataLogOpen ausgeführt wurde.</p> <p>Beim Wechsel in den Betriebszustand STOP werden alle geöffneten Datenprotokolldateien geschlossen.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

² Im SCL-Beispiel ist "DataLogClose_DB" der Name des Instanz-DBs.

Tabelle 9-212 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Standardwert: Falsch)
ID	Durchgang	DWord Numerische Kennung eines Datenprotokolls. Wird bei der Anweisung DataLogClose nur als Eingang verwendet. (Standardwert: 0) Hinweis: Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> 0 - Keine Anweisung in Bearbeitung 1 - Vorgang in Bearbeitung

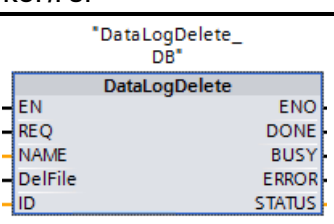
Parameter und Datentyp	Datentyp	Beschreibung
ERROR	OUT	Bool
		Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word
		Ausführungsbedingung (Standardwert: 0)

Tabelle 9-213 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	0001	Datenprotokoll nicht geöffnet
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8092	Datenprotokoll ist nicht vorhanden.

DataLogDelete (Datenprotokoll löschen)

Tabelle 9-214 Anweisung DataLogDelete

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung "DataLogDelete" löschen Sie eine Datenprotokolldatei. Das Datenprotokoll und die darin enthaltenen Datensätze können nur gelöscht werden, wenn sie mit der Anweisung "DataLogCreate" oder "DataLogNewFile" angelegt wurden.</p>

"DataLogDelete_DB" ist der Name des Instanz-DBs.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "DataLogDelete":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L, T, C oder Konstante (T und C sind nur bei der S7-1500 in KOP und FUP verfügbar)	Ausführung der Anweisung bei einer steigenden Flanke
NAME	Input	VARIANT	L, D	Dateiname des Datenprotokolls
DELFILE	Input	BOOL	E, A, M, D, L oder Konstante	<ul style="list-style-type: none"> 0: Datenprotokoll wird gespeichert. 1: Datenprotokoll wird gelöscht.
ID	InOut	DWORD	E, A, M, D, L	Numerische Kennung des Datenprotokolls

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DONE	Output	BOOL	E, A, M, D, L	Die Anweisung wurde erfolgreich ausgeführt.
BUSY	Output	BOOL	E, A, M, D, L	Das Löschen des Datenprotokolls ist noch nicht beendet.
ERROR	Output	BOOL	E, A, M, D, L	<ul style="list-style-type: none"> 0: Kein Fehler. 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen werden am Parameter STATUS ausgegeben.
STATUS	Output	WORD	E, A, M, D, L	Statusparameter Der Parameter wird nur für die Dauer eines Aufrufs gesetzt. Um den Status anzuzeigen, sollten Sie deshalb den Parameter STATUS in einen freien Datenbereich kopieren.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Parameter NAME und ID

Wählen Sie das zu löschende Datenprotokoll mit den Parametern NAME und ID aus. Der Parameter ID wird zuerst ausgewertet. Ist ein Datenprotokoll mit der relevanten ID vorhanden, wird der Parameter NAME nicht ausgewertet. Wenn am Parameter ID der Wert 0 verwendet wird, muss am ParameterNAME ein Wert mit dem Datentyp STRING verwendet werden.

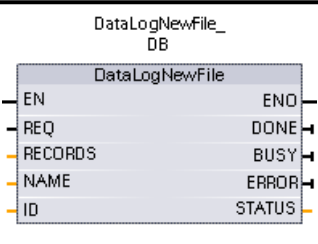
Parameter RET_VAL

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler.
7000	Keine laufende Auftragsbearbeitung.
7001	Start der Auftragsbearbeitung. Parameter BUSY = 1, DONE = 0
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; BUSY hat den Wert 1.
8091	Ein anderer Datentyp als STRING wird am Parameter NAME verwendet.
8092	Datenprotokoll ist nicht vorhanden.
80A2	Vom Dateisystem zurückgemeldeter Schreibfehler.
80B4	Die Memory Card ist schreibgeschützt.

* Die Fehlercodes können als ganzzahlige Werte oder als Hexadezimalwerte im Programmmeditor angezeigt werden. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".

DataLogNewFile (Datenprotokoll in neuer Datei)

Tabelle 9-215 Anweisung DataLogNewFile

KOP/FUP	SCL	Beschreibung
 <p>Das Diagramm zeigt die Anweisung DataLogNewFile mit den Parametern EN, REQ, RECORDS, NAME, ID, END, DONE, BUSY, ERROR, STATUS. Die Parameter sind in einem rechteckigen Kasten dargestellt, der in zwei Spalten angeordnet ist. Die Parameter EN, REQ, RECORDS, NAME, ID sind links aufgeführt, während END, DONE, BUSY, ERROR, STATUS rechts aufgeführt sind. Die Parameter EN, REQ, RECORDS, NAME, ID sind mit einem Minuszeichen (-) markiert, was auf Eingangsparameter hinweist, während END, DONE, BUSY, ERROR, STATUS mit einem Pluszeichen (+) markiert sind, was auf Ausgangsparameter hinweist.</p>	<pre>"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_);</pre>	<p>Ermöglicht Ihrem Programm, eine neue Datenprotokolldatei basierend auf einer bestehenden Datenprotokolldatei anzulegen.</p> <p>Wenn Sie die Anweisung einfügen, erstellt STEP 7 automatisch den zugehörigen Instanz-DB.</p>

² Im SCL-Beispiel ist "DataLogNewFile_DB" der Name des Instanz-DBs.

Tabelle 9-216 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
RECORDS	IN	UDInt
NAME	IN	Variant
ID	Durchgang	DWord
DONE	OUT	Bool
BUSY	OUT	Bool
ERROR	OUT	Bool
STATUS	OUT	Word

Die Beschreibung für den Parameter ID lautet:

Numerische Kennung des Datenprotokolls (Standardwert: 0):

- Während der Ausführung gibt der Eingang ID ein gültiges Datenprotokoll an. Die neue Datenprotokollkonfiguration wird aus diesem Datenprotokoll kopiert.
- Nach der Ausführung wird der Parameter ID zu einem Ausgang, der die ID der neu angelegten Datenprotokolldatei ausgibt.

Hinweis: Der Zugriff auf diesen Parameter über einen symbolischen Namen ist nicht zulässig.

Sie können die Anweisung DataLogNewFile ausführen, wenn ein Datenprotokoll voll ist oder nahezu voll ist und Sie keine Daten verlieren möchten, die im Datenprotokoll gespeichert sind. Basierend auf der Struktur der vollen Datenprotokolldatei kann eine neue, leere Datenprotokolldatei angelegt werden. Der Kopfdatensatz wird mit den Eigenschaften des ursprünglichen Datenprotokolls (DATA-Einstellungen für Datensatzpuffer, Datenformat und Zeitstempel) aus dem ursprünglichen Datenprotokoll dupliziert. Die ursprüngliche Datenprotokolldatei wird implizit geschlossen und die neue Datenprotokolldatei implizit geöffnet.

Auslösung von DataLogWrite durch Parameter: Ihr Programm muss die Parameter ERROR und STATUS jeder Anweisung DataLogWrite überwachen. Wenn der letzte Datensatz geschrieben wurde und ein Datenprotokoll voll ist, sind das ERROR-Bit der Anweisung DataLogWrite = 1 und das Datenwort STATUS der Anweisung DataLogWrite = 1. Diese ERROR- und STATUS-Werte sind nur einen Zyklus lang gültig, deshalb muss Ihre Überwachungslogik ERROR = 1 als Zeitfenster nutzen, um den STATUS-Wert zu erfassen und dann auf STATUS = 1 zu prüfen (Datenprotokoll ist voll).

Funktionsweise von DataLogNewFile: Wenn Ihre Programmlogik das Signal "Datenprotokoll voll" erhält, wird mit diesem Zustand eine Anweisung DataLogNewFile aktiviert. Sie müssen Anweisung DataLogNewFile mit der ID eines vorhandenen (üblicherweise vollen) und geöffneten Datenprotokolls ausführen, der Parameter NAME muss jedoch neu und eindeutig sein. Nach Ausführung der Anweisung DataLogNewFile wird ein neuer ID-Wert für das Datenprotokoll ausgegeben (als Ausgangsparameter), der dem Namen des neuen Datenprotokolls entspricht. Die neue Datenprotokolldatei wird implizit geöffnet und kann neue Datensätze speichern. Neue Anweisungen DataLogWrite, die sich an die neue Datenprotokolldatei richten, müssen den von der Anweisung DataLogNewFile ausgegebenen ID-Wert nutzen.

ACHTUNG**Die Datenprotokollerstellung muss abgeschlossen sein, bevor eine Anweisung zum Schreiben des Datenprotokolls angestoßen werden kann**

Die Operationen DataLogCreate und DataLogNewFile zum Erstellen von Datenprotokollen erstrecken sich über viele Programmzyklen. Die tatsächliche für die Erstellung der Protokolldatei benötigte Zeit hängt von der Datensatzstruktur und der Anzahl der Datensätze ab. Ihre Programmlogik muss den Wechsel des DONE-Bits nach WAHR, der die abgeschlossene Erstellung einer Protokolldatei meldet, überwachen und erfassen. Wird eine Anweisung DataLogWrite vor Abschluss der Datenprotokollerstellung ausgeführt, kann die Schreiboperation den neuen Protokolldatensatz nicht wie erwartet schreiben.

Tabelle 9-217 Werte von ERROR und STATUS

ERROR	STATUS (W#16#)	Beschreibung
0	0000	Kein Fehler
0	7000	Aufruf ohne REQ-Flanke: BUSY = 0, DONE = 0
0	7001	Erster Aufruf mit REQ-Flanke (in Bearbeitung): BUSY = 1, DONE = 0
0	7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1, DONE = 0
1	8070	Gesamter interner Instanzspeicher ist belegt.
1	8090	Ungültiger Dateiname
1	8091	Der Namensparameter ist keine Zeichenkette.

ERROR	STATUS (W#16#)	Beschreibung
1	8092	Datenprotokoll ist nicht vorhanden.
1	8093	Das Datenprotokoll ist bereits vorhanden.
1	8097	Gewünschte Dateilänge überschreitet maximale Dateigröße für das Dateisystem.
1	80B2	Keine weiteren Ressourcen-IDs vorhanden Hinweis: Löschen Sie einige vorhandene Datenprotokolle, um Platz für ein neues Datenprotokoll zu schaffen.
1	80B3	Nicht genügend Ladespeicher.
1	80B4	MC ist schreibgeschützt.
1	80C1	Zu viele geöffnete Dateien.

9.9.2.3 Arbeiten mit Datenprotokollen

Die Datenprotokolldateien werden im beständigen Flash-Speicher im CSV-Format (durch Komma getrennte Werte) gespeichert. Sie können die Datenprotokolle über den PLC-Webserver anzeigen oder indem Sie die Memory Card des PLC-Geräts entnehmen und in einen herkömmlichen PC-Kartenleser einlegen.

Datenprotokolle über den PLC-Webserver anzeigen

Wenn der PROFINET-Anschluss des PLC-Geräts und ein PC an ein Netzwerk angeschlossen sind, können Sie einen PC-Webbrowser wie Microsoft Internet Explorer oder Mozilla Firefox nutzen, um auf den integrierten PLC-Webserver zuzugreifen. Das PLC-Gerät kann sich, während Sie den PLC-Webserver bedienen, in der Betriebsart RUN oder STOP befinden. Befindet sich das PLC-Gerät im Betriebszustand RUN, wird Ihr Steuerungsprogramm weiterhin ausgeführt, während der PLC-Webserver über das Netzwerk Protokoll Daten überträgt.

Zugriff auf den Webserver:

1. Aktivieren Sie den Webserver in der Gerätekonfiguration der Ziel-CPU (Seite 850).
2. Verbinden Sie Ihren PC über das PROFINET-Netzwerk mit dem Zielsystem (Seite 855).
3. Greifen Sie über den integrierten Webserver auf die CPU zu (Seite 861).
4. Laden, bearbeiten und löschen Sie Datenprotokolldateien mit der Standard-Webseite "Dateibrowser" (Seite 893).
5. Öffnen Sie die .csv-Datei mit einem Tabellenkalkulationsprogramm wie Excel.

Hinweis

Verwaltung von Datenprotokollen

Speichern Sie maximal 1000 Datenprotokolle in einem Dateisystem. Bei Überschreiten dieser Anzahl kann es passieren, dass der Webserver nicht mehr genügend CPU-Ressourcen zur Verfügung hat, um die Datenprotokolle anzuzeigen.

Wenn Sie bemerken, dass die Webseite des Dateibrowsers die Datenprotokolle nicht anzeigen kann, müssen Sie die CPU in den Betriebszustand STOP versetzen, um Datenprotokolle anzeigen und löschen zu können.

Stellen Sie über die Verwaltung Ihrer Datenprotokolle sicher, dass Sie lediglich die erforderliche Anzahl Datenprotokolle speichern und die maximale Anzahl von 1000 Datenprotokollen nicht überschreiten.

Datenprotokolle auf einer Memory Card des PLC-Geräts anzeigen

Ist in der S7-1200 CPU eine Memory Card vom Typ "Programmkarte" gesteckt, können Sie die Memory Card entnehmen und in einen Standardkartensteckplatz wie SD (Secure Digital) oder MMC (MultiMediaCard) an einem PC oder PG einfügen. Das PLC-Gerät ist beim Entnehmen der Memory Card im Betriebszustand STOP und Ihr Steuerungsprogramm wird nicht ausgeführt.

Öffnen Sie dann den Windows Explorer für Dateien und navigieren Sie zum Verzeichnis \Datenprotokoll auf der Memory Card. In diesem Verzeichnis befinden sich alle Ihre CSV-Datenprotokolldateien.

Kopieren Sie die Datenprotokolldateien in ein lokales Laufwerk auf Ihrem PC. Dann können Sie eine lokale Kopie einer CSV-Datei mit Excel öffnen; dies ist nicht die ursprüngliche Datei, die auf der Memory Card gespeichert ist.

ACHTUNG

Sie können Datenprotokolldateien über einen PC-Kartenleser von einer S7-1200 Memory Card kopieren, jedoch noch ändern oder löschen.
--

Es ist empfehlenswert, den Standard-Dateibrowser des Webservers zu nutzen, um Datenprotokolldateien anzuzeigen, herunterzuladen (zu kopieren) und zu löschen.

Das direkte Aufrufen des Dateisystems der Memory Card im Windows Explorer birgt das Risiko, dass Sie versehentlich Datenprotokoll- oder andere Systemdateien löschen oder ändern können, so dass möglicherweise eine Datei beschädigt oder die Memory Card unbrauchbar wird.
--

ACHTUNG

Auswirkung von Datenprotokollen auf Memory Cards

Begrenzen Sie die Datenprotokollrate auf einen Wert von maximal alle 200 ms, um das Gesamtverhalten und die Robustheit Ihres Systems zu gewährleisten.
--

9.9.2.4 Grenzwert für die Größe von Datenprotokolldateien

Datenprotokolldateien nutzen den Ladespeicher des PLC-Geräts gemeinsam mit dem Programm, mit Programmdateien, Konfigurationsdateien, benutzerdefinierten Webseiten und PLC-Systemdaten. Ein großes Programm, das den internen Ladespeicher nutzt, belegt viel Platz im Ladespeicher. Möglicherweise steht dadurch nicht genügend freier Speicherplatz für Datenprotokolldateien zur Verfügung. In diesem Fall können Sie eine Memory Card als "Programmkarte" (Seite 124) nutzen, um die Kapazität des Ladespeichers zu erhöhen. S7-1200 CPUs können entweder internen oder externen Ladespeicher nutzen, jedoch nicht beide gleichzeitig.

Regel für die maximale Größe von Datenprotokolldateien

Die maximale Größe einer Datenprotokolldatei darf die Größe des freien Ladespeichers bzw. 500 Megabyte (je nachdem, was kleiner ist) nicht überschreiten. Die Größe von 500 Megabyte bezieht sich in diesem Fall auf die Dezimaldefinition von Megabyte, sodass die maximale Größe von Datenprotokolldateien 500.000.000 Byte bzw. 500×1000^2 Byte beträgt.

Tabelle 9-218 Größe des Ladespeichers

Datenbereich	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C, CPU 1217C	Datenspeicherung
Interner Ladespeicher Flash-Speicher	1 MB	2 MB	4 MB	4 MB	Anwenderprogramm und Programmdateien, Konfigurationsdateien, Datenprotokolle, benutzerdefinierte Webseiten und PLC-Systemdaten
Externer Ladespeicher Optionale Memory Card als "Programmkarte"	4 MB, 12 MB, 24 MB, 256 MB, 2 GB oder 32 GB, je nach Größe der SD-Karte				Anwenderprogramm und Programmdateien, Konfigurationsdateien, Datenprotokolle, benutzerdefinierte Webseiten und PLC-Systemdaten

Ermitteln des freien Ladespeichers

Die Kapazität des freien Ladespeichers variiert im Normalbetrieb, weil das Betriebssystem ständig Speicher benötigt und wieder freigibt. Um die Größe des Ladespeichers anzuzeigen, gehen Sie wie folgt vor.

1. Stellen Sie eine Online-Verbindung zwischen STEP 7 und dem S7-1200 Zielsystem her.
2. Laden Sie das Programm, das Ihre Datenprotokollanweisungen steuert.
3. Erstellen Sie optional nach Bedarf anwenderdefinierte Webseiten. Die Standard-Webseiten für den Zugriff auf Datenprotokolle sind in der Firmware des PLC-Geräts gespeichert und belegen keinen Platz im Ladespeicher.
4. Fragen Sie mit den Online- und Diagnosefunktionen (Seite 1205) oder der Diagnoseseite des Webservers (Seite 868) die Gesamtgröße des Ladespeichers und den freien Speicherplatz ab.

Berechnen der Größe einer Datenprotokolldatei (alle Datensätze)

Beim Anlegen der Datenprotokolldatei ordnet die CPU die maximale Speichergröße zu. Neben der erforderlichen Größe für alle Datensätze müssen Sie den Speicherplatz für einen Datenprotokoll-Header (sofern verwendet), Zeitstempel-Header (sofern verwendet), Datensatzindex-Header und die Mindestbausteingröße für die Speicherzuordnung berücksichtigen.

Ermitteln Sie die Größe Ihrer Datenprotokolldateien mit Hilfe der folgenden Formel und achten Sie darauf, dass Sie die Regel für die maximale Größe einhalten.

Datenbytes des Datenprotokolls = ((Datenbytes in einem Datensatz + Zeitstempel-Bytes + 12 Bytes) * Anzahl der Datensätze)

Header

Headerbytes des Datenprotokolls = Header-Zeichenbytes + 2 Byte

Header-Zeichenbytes

- Kein Daten-Header und keine Zeitstempel = 7 Byte
- Kein Daten-Header und Zeitstempel (mit Zeitstempel-Header) = 21 Byte
- Daten-Header und keine Zeitstempel = Anzahl der Zeichenbytes in allen Spaltenüberschriften einschließlich Trennzeichen-Kommas
- Daten-Header und Zeitstempel (mit Zeitstempel-Header) = Anzahl der Zeichenbytes in allen Spaltenüberschriften einschließlich Trennzeichen-Kommas + 21 Byte

Daten

Datenbytes des Datenprotokolls = ((Datenbytes in einem Datensatz + Zeitstempel-Bytes + 12 Bytes) * Anzahl der Datensätze)

Datenbytes in einem Datensatz

Der Parameter DATA der Anweisung DataLogCreate zeigt auf eine Struktur, die die Anzahl der Datenfelder und den Datentyp jedes Datenfelds für einen Protokolldatensatz zuweist.

Multiplizieren Sie die Anzahl des jeweiligen Datentyps mit der erforderlichen Anzahl Bytes für diesen Datentyp. Wiederholen Sie den Vorgang für jeden Datentyp in einem Datensatz und addieren Sie alle Datenbytes, um die Summe aller Datenelemente in einem Datensatz zu erhalten.

Größe der einzelnen Datenelemente

Die Daten in Datenprotokolldateien werden als Zeichenbytes im CSV-Format (durch Komma getrennte Werte) gespeichert. Die folgende Tabelle zeigt die Anzahl Bytes, die erforderlich sind, um jedes Datenelement zu speichern.

Datentyp	Anzahl der Bytes (einschließlich Daten plus ein Kommabyte)
Bool	2
Byte	5
Wort	7
DWord	12

Char	4
String	<p>Beispiel 1: MyString[10] Die maximale Zeichenkettengröße wird mit 10 Zeichen angegeben. Textzeichen + automatische Auffüllung mit Leerzeichen = 10 Byte Anführungszeichen am Anfang und am Ende + Kommazeichen = 3 Byte 10 + 3 = 13 Byte insgesamt</p> <p>Beispiel 2: Mystring2 Wird keine Größe in eckigen Klammern angegeben, werden standardmäßig 254 Byte zugewiesen. Textzeichen + automatische Auffüllung mit Leerzeichen = 254 Byte Anführungszeichen am Anfang und am Ende + Kommazeichen = 3 Byte 254 + 3 = 257 Byte insgesamt</p>
USInt	5
UInt	7
UDInt	12
SInt	5
Int	7
DInt	12
Real	16
LReal	25
Time	15
DTL	24

Anzahl der Datensätze in einer Datenprotokolldatei

Der Parameter RECORDS der Anweisung DataLogCreate legt die maximale Anzahl Datensätze in einer Datenprotokolldatei fest.

Zeitstempelbytes in einem Datensatz

- Kein Zeitstempel = 0 Byte
- Zeitstempel = 20 Byte

9.9.2.5 Beispielprogramm für Datenprotokolle

Dieses Beispielprogramm für Datenprotokolle zeigt nicht die gesamte Programmlogik, die erforderlich ist, um Abtastwerte eines dynamischen Prozesses abzurufen, es zeigt jedoch die wesentliche Funktionsweise der Anweisungen für Datenprotokolle. Die Struktur und Anzahl der verwendeten Protokolldateien richtet sich nach Ihren Prozesssteuerungsanforderungen.

Hinweis

Allgemeine Verwendung von Datenprotokolldateien

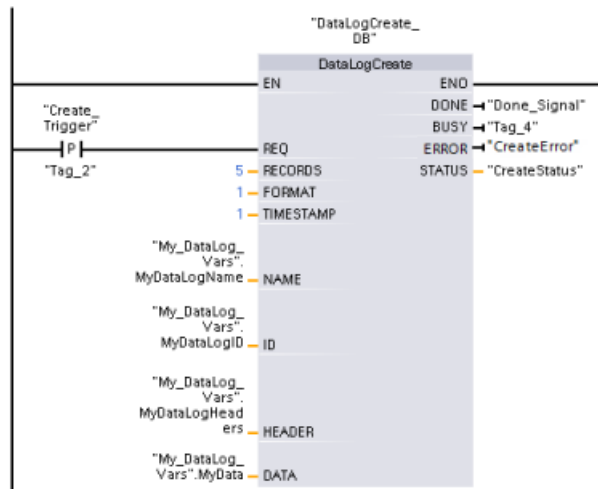
- Nach Ausführung der Anweisungen DataLogCreate und DataLogNew werden automatisch Datenprotokolldateien geöffnet.
 - Nach einem Wechsel des PLC-Geräts von RUN in STOP oder nach einem Neustart des PLC-Geräts werden Datenprotokolldateien automatisch geschlossen.
 - Damit eine Anweisung DataLogWrite durchgeführt werden kann, muss eine Datenprotokolldatei geöffnet sein.
 - Maximal acht Datenprotokolldateien dürfen gleichzeitig geöffnet sein. Mehr als acht Datenprotokolldateien dürfen vorhanden sein, von diesen müssen jedoch einige geschlossen werden, so dass maximal acht gleichzeitig geöffnet sind.
-

Beispielprogramm für Datenprotokolle

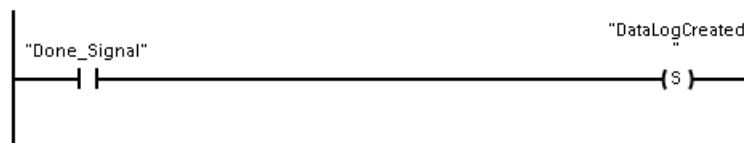
Beispielhafte Datenprotokollnamen, Kopfzeilentext und die Struktur MyData werden in einem Datenbaustein erstellt. Die drei MyData-Variablen speichern neue Abtastwerte temporär. Die Prozessabtastwerte an diesen DB-Adressen werden durch Ausführung der Anweisung DataLogWrite in eine Datenprotokolldatei übertragen.

My_Datalog_Vars			
	Name	Datentyp	Startwert
1	Static		
2	MyNewDataLogName	String	'MyNEWDatLog'
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

Netzwerk 1 Eine steigende Flanke an REQ startet die Datenprotokollerstellung.



Netzwerk 2 Erfassen Sie den Ausgang DONE von DataLogCreate, weil er nur einen Zyklus lang gültig ist.



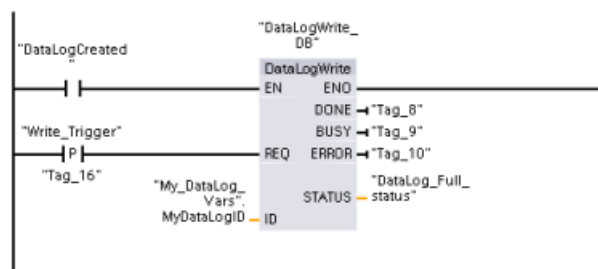
Netzwerk 3 Liegt ein Fehler vor, speichern Sie den Statusausgang.



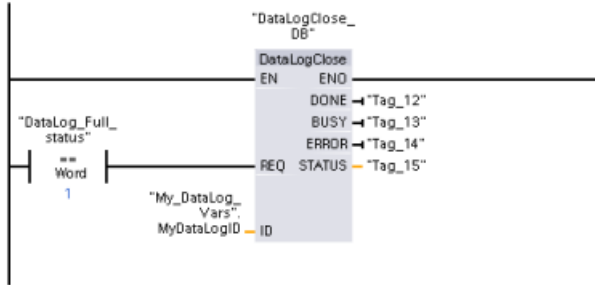
Netzwerk 4 Eine positive Flanke löst den Zeitpunkt aus, an dem neue Prozesswerte in der Struktur MyData gespeichert werden.



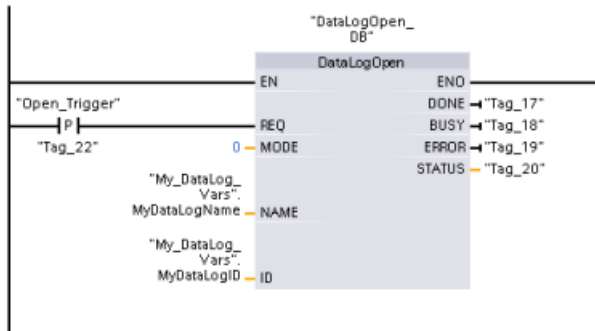
Netzwerk 5 Der Zustand des Eingangs EN basiert auf dem Zeitpunkt, zu dem die Ausführung von DataLogCreate beendet ist. Ein Erstellungsvorgang erstreckt sich über viele Zyklen und muss beendet sein, damit ein Schreibvorgang durchgeführt werden kann. Die positive Flanke am Eingang REQ ist das Ereignis, das eine aktivierte Schreiboperation auslöst.



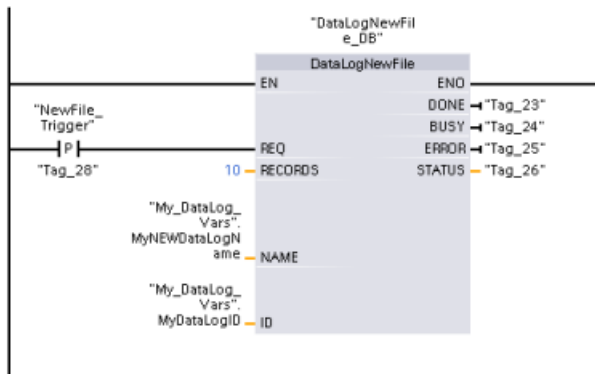
Netzwerk 6 Schließen Sie das Datenprotokoll, nachdem der letzte Datensatz geschrieben wurde. Nach Ausführung der Anweisung DataLogWrite, die den letzten Datensatz schreibt, wird der Zustand "Protokolldatei voll" gemeldet, wenn der Ausgang STATUS von DataLogWrite = 1 ist.



Netzwerk 7 Eine positive Flanke am Eingang REQ der Anweisung DataLogOpen simuliert, dass der Anwender an einem HMI-Gerät eine Taste drückt, die eine Datenprotokolldatei öffnet. Wenn Sie eine Datenprotokolldatei öffnen, in der alle Datensätze mit Prozessdaten belegt sind, überschreibt die nächste Ausführung der Anweisung DataLogWrite den ältesten Datensatz. Sie können jedoch auch das alte Datenprotokoll aufbewahren und stattdessen ein neues Datenprotokoll anlegen. Dies wird in Netzwerk 7 gezeigt.



Netzwerk 8 Der ID-Parameter ist ein IN/OUT-Typ. Zunächst geben Sie den ID-Wert des vorhandenen Datenprotokolls an, dessen Struktur Sie kopieren möchten. Nachdem die Anweisung DataLogNewFile durchgeführt wurde, wird ein neuer und eindeutiger ID-Wert für das neue Datenprotokoll in die Adresse des ID-Verweises zurückgeschrieben. Die erforderliche Erfassung ONE-Bit = WAHR wird nicht gezeigt. Ein Beispiel für die Logik des DONE-Bits finden Sie in den Netzwerken 1, 2 und 4.



Vom Beispielprogramm angelegte und mit dem Webserver der S7-1200 CPU angezeigte Datenprotokolldateien



- ① Die Option "Delete" ist nur verfügbar, wenn Sie mit Änderungsrechten angemeldet sind.
- ② Die Option "Rename" ist nur verfügbar, wenn Sie mit Änderungsrechten angemeldet sind.

Tabelle 9-219 In Excel angezeigte, heruntergeladene Beispiele für CSV-Dateien

Zwei geschriebene Datensätze in einer Datei, die maximal fünf Datensätze enthalten kann		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/29/2010	21:01:46	5	5.00E+00	5.00E+00
	3	2	9/29/2010	21:01:47	5	5.00E+00	5.00E+00
	4						
Fünf Datensätze in einer Datenprotokolldatei, die maximal fünf Datensätze enthalten kann		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	1	9/30/2010	20:28:56	1	9.86E+01	3.52E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
Nachdem ein weiterer Datensatz in die oben abgebildete Datei, die bereits voll ist, geschrieben wurde, überschreibt der sechste Schreibvorgang den ältesten Datensatz 1 mit Datensatz 6. Ein weiterer Schreibvorgang überschreibt Datensatz 2 mit Datensatz 7 usw.		A	B	C	D	E	F
	1	Record	Date	UTC Time	Count	Temperature	Pressure
	2	6	9/30/2010	20:32:03	6	9.86E+01	3.58E+01
	3	2	9/30/2010	20:28:43	2	1.00E+02	3.73E+01
	4	3	9/30/2010	20:29:03	3	9.99E+01	3.68E+01
	5	4	9/30/2010	20:29:21	4	9.95E+01	3.64E+01
	6	5	9/30/2010	20:30:19	5	9.92E+01	3.74E+01
7							

Hinweis

Es werden keine //END-Merker mehr am Ende eines Datenprotokolls gesetzt, das nicht voll ist. Vor der S7-1200 CPU V4.1 enthielten Datenprotokolle, die nicht voll waren, einen //END-Merker.

9.10 Datenbausteinststeuerung

9.10.1 CREATE_DB (Datenbaustein erstellen)

Tabelle 9-220 Anweisung CREATE_DB

KOP/FUP	SCL	Beschreibung
<p>CREATE_DB</p> <ul style="list-style-type: none"> - EN ENO - - REQ RET_VAL - - LOW_LIMIT BUSY - - UP_LIMIT DB_NUM - - COUNT - ATTRIB - SRCBLK 	<pre>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_);</pre>	<p>Mit der Anweisung "CREATE_DB" erstellen Sie einen neuen Datenbaustein im Lade- und/oder Arbeitsspeicher.</p> <p>Die Anweisung "CREATE_DB" ändert die Prüfsumme des Anwenderprogramms nicht.</p> <p>Ein Datenbaustein, den Sie nur im Arbeitsspeicher erstellen, hat die folgenden Eigenschaften:</p> <ul style="list-style-type: none"> Nach einem Umröschern oder AUSSCHALTEN / EINSCHALTEN ist dieser Baustein nicht mehr vorhanden. Beim Laden oder bei einem Wechsel von STOP in RUN wird der Inhalt nicht verändert.

Nummer des Datenbausteins

Dem erstellten Datenbaustein wird eine Nummer aus dem von den Parametern LOW_LIMIT (unterer Grenzwert) und UP_LIMIT (oberer Grenzwert) definieren Bereich zugewiesen. "CREATE_DB" weist dem DB die kleinstmögliche Nummer aus dem angegebenen Bereich zu. Die Nummern der bereits im Anwenderprogramm enthaltenen DBs können Sie nicht zuweisen.

Um einen DB mit einer bestimmten Nummer zu erstellen, geben Sie die gleiche Nummer für den oberen und den unteren Grenzwert des anzugebenden Bereichs ein. Wenn bereits ein DB mit dieser Nummer im Arbeitsspeicher und/oder im Ladespeicher vorhanden ist oder wenn der DB als kopierte Version vorhanden ist, wird die Anweisung beendet und am Parameter RET_VAL eine Fehlermeldung erzeugt.

Startwerte des Datenbausteins

Mit dem Parameter SRCBLK definieren Sie Startwerte für den zu erstellenden DB. Der Parameter SRCBLK ist ein Pointer auf einen DB oder einen DB-Bereich, aus dem Sie die Startwerte übernehmen. Der am Parameter SRCBLK adressierte DB muss mit Standardzugriff erstellt worden sein (Attribut "Optimierter Bausteinzugriff" deaktiviert).

- Wenn der am Parameter SRCBLK angegebene Bereich größer ist als der erstellte DB, werden die Werte bis zur Länge des erstellten DBs als Startwerte übernommen.
- Wenn der am Parameter SRCBLK angegebene Bereich kleiner ist als der erstellte DB, werden die übrigen Werte mit 0 aufgefüllt.

Um Datenkonsistenz zu gewährleisten, dürfen Sie diesen Datenbereich nicht ändern, während "CREATE_DB" ausgeführt wird (d.h. solange der Parameter BUSY den Wert TRUE hat).

Funktionsbeschreibung

Die Anweisung "CREATE_DB" arbeitet asynchron. Die Verarbeitung erfolgt über mehrere Aufrufe. Sie starten den Auftrag durch Aufruf von "CREATE_DB" mit REQ = 1.

Die Ausgangsparameter RET_VAL und BUSY melden den Status des Auftrags.

Siehe auch: DELETE_DB (Datenbaustein löschen) (Seite 527)

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "CREATE_DB":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L oder Konstante	Pegelausgelöster Steuerparameter "request to activate" REQ = 1: Anforderung zum Erstellen des Datenbausteins
LOW_LIMIT	Input	UINT	E, A, M, D, L oder Konstante	Unterer Grenzwert des Bereichs für die Zuweisung einer DB-Nummer. Die kleinstmögliche Nummer ist 60000.
UP_LIMIT	Input	UINT	E, A, M, D, L oder Konstante	Oberer Grenzwert des von "CREATE_DB" verwendeten Bereichs für die Zuweisung einer Nummer zu Ihrem DB (größtmögliche DB-Nummer: 60999)
COUNT	Input	UDINT	E, A, M, D, L oder Konstante	Der Zählwert gibt die Anzahl von Bytes an, die Sie für den erstellten DB generieren möchten. Die Anzahl von Bytes muss eine gerade Nummer sein. Die maximale Länge beträgt 65534 Bytes.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung												
ATTRIB	Input	BYTE	E, A, M, D, L oder Konstante	Sie verwenden die ersten 4 Bits des Byte am Parameter ATTRIB zum Definieren der Eigenschaften des Datenbausteins*:												
				<ul style="list-style-type: none"> • Bit 0 = 0: Attribut "Nur im Ladespeicher ablegen" ist nicht festgelegt. • Bit 0 = 1: Attribut "Nur im Ladespeicher ablegen" ist festgelegt. Bei dieser Einstellung belegt der DB keinen Speicherplatz im Arbeitsspeicher und wird nicht ins Programm aufgenommen. Auf den DB kann nicht über Bitbefehle zugegriffen werden. Wenn Bit 0 = 1, ist die Auswahl für Bit 2 irrelevant. <p>Um Kompatibilität mit STEP 7 V5.x zu gewährleisten, müssen die Bits 0 und 3 gemeinsam berücksichtigt werden (siehe unten).</p>												
				<ul style="list-style-type: none"> • Bit 1 = 0: Das Attribut "Datenbaustein im Gerät schreibgeschützt" ist nicht festgelegt. • Bit 1 = 1: Das Attribut "Datenbaustein im Gerät schreibgeschützt" ist festgelegt. 												
				<ul style="list-style-type: none"> • Bit 2 = 0: Der DB ist remanent (nur bei DBs, die im Lade- und Arbeitsspeicher erstellt wurden). Der DB wird als remanent betrachtet, wenn mindestens ein Wert als remanent definiert wurde. • Bit 2 = 1: Der DB ist nicht remanent. <p>Remanenz wird bei DBs, die nur im Ladespeicher oder nur im Arbeitsspeicher abgelegt sind, nicht unterstützt. Wenn Sie die Anweisung "CREATE_DB" mit einer der beiden Kombinationen "remanent und nur Ladespeicher" oder "remanent und nur Arbeitsspeicher" aufrufen, wird der zu erstellende DB nicht als remanent markiert.</p>												
				<ul style="list-style-type: none"> • Bit 3 = 0: Erstellung des DB entweder im Ladespeicher oder im Arbeitsspeicher (Auswahl über Bit 0, siehe oben) • Bit 3 = 1: Erstellung des DB sowohl im Ladespeicher als auch im Arbeitsspeicher (Bit 0 irrelevant) <p>Um Kompatibilität mit STEP 7 V5.x zu gewährleisten, müssen die Bits 0 und 3 gemeinsam verwendet werden. Wenn Bit 3 = 1, ist Bit 0 irrelevant.</p>												
				<table border="1"> <thead> <tr> <th>Bit 0</th> <th>Bit 3</th> <th>DB-Erstellung</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Nur im Arbeitsspeicher</td> </tr> <tr> <td>1</td> <td>0</td> <td>Nur im Ladespeicher</td> </tr> <tr> <td>Irrelevant</td> <td>1</td> <td>Arbeits- und Ladespeicher</td> </tr> </tbody> </table>	Bit 0	Bit 3	DB-Erstellung	0	0	Nur im Arbeitsspeicher	1	0	Nur im Ladespeicher	Irrelevant	1	Arbeits- und Ladespeicher
				Bit 0	Bit 3	DB-Erstellung										
0	0	Nur im Arbeitsspeicher														
1	0	Nur im Ladespeicher														
Irrelevant	1	Arbeits- und Ladespeicher														
<ul style="list-style-type: none"> • Bit 4 = 0 - Keine Startwerte angegeben (Eingangswerte am Parameter SRCBLK werden ignoriert). • Bit 4 = 1 - Startwerte angeben (Werte entsprechen dem vom Parameter SRCBLK adressierten DB). 																
SRCBLK	Input	VARIANT	D	Pointer auf den Datenbaustein, dessen Werte zum Initialisieren des zu erstellenden Datenbausteins verwendet werden.												
RET_VAL	Return	INT	E, A, M, D, L	Fehlerinformationen												
BUSY	Output	BOOL	E, A, M, D, L	BUSY = 1: Der Vorgang ist noch nicht beendet.												

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DB_NUM	Output	DB_DYN (UINT)	E, A, M, D, L	Nummer des erstellten DBs.
* Die hier ausgewählten Eigenschaften entsprechen den Attributen in den Eigenschaften eines Datenbausteins.				

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Parameter RET_VAL

Fehlercode* (W#16#...)	Beschreibung
0000	Kein Fehler
0081	Der Zielbereich ist größer als der Quellbereich. Der Quellbereich wird vollständig in den Zielbereich geschrieben. Die übrigen Bytes des Zielbereichs werden nicht geändert.
7000	Erster Aufruf mit REQ = 0: Keine Datenübertragung aktiv; BUSY hat den Wert 0.
7001	Erster Aufruf mit REQ = 1: Datenübertragung angestoßen; BUSY hat den Wert 1.
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Datenübertragung bereits aktiv; BUSY hat den Wert 1.
8081	Der Quellbereich ist größer als der Zielbereich. Der Zielbereich wird vollständig beschrieben. Die übrigen Bytes des Quellbereichs werden ignoriert.
8092	Die Funktion "Datenbaustein erstellen" ist gegenwärtig nicht verfügbar, weil <ul style="list-style-type: none"> • Die Funktion "Anwenderspeicher komprimieren" gerade aktiv ist. • Die maximale Anzahl von Bausteinen in Ihrer CPU bereits erreicht ist.
8093	Kein Datenbaustein oder ein Datenbaustein, der sich nicht im Arbeitsspeicher befindet, wurde für den Parameter SRCBLK angegeben.
8094	Ein ungültiger Wert wurde am Parameter ATTRIB angegeben.
80A1	DB-Fehlernummer: <ul style="list-style-type: none"> • Die Nummer ist 0. • Die Nummer überschreitet den CPU-spezifischen oberen Grenzwert für DB-Nummern. • Unterer Grenzwert > oberer Grenzwert.
80A2	DB-Längenfehler: <ul style="list-style-type: none"> • Die Länge ist 0. • Die Länge ist eine ungerade Zahl. • Die Länge ist größer als für die CPU zulässig.
80A3	Der Datenbaustein am Parameter SRCBLK wurde nicht mit Standardzugriff erstellt.
80B1	Es ist keine DB-Nummer frei.
80B2	Nicht genügend Arbeitsspeicher.
80B4	Die Memory Card ist schreibgeschützt.
80BB	Nicht genügend Ladespeicher.
80C3	Die maximale Anzahl gleichzeitig aktiver Anweisungen "CREATE_DB" ist bereits erreicht.
Allgemeine Fehlerinformationen	Siehe auch: Gemeinsame Fehlercodes für die erweiterten Anweisungen (Seite 535)
* Fehlercodes können im Programmreditor als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden.	

9.10.2 Anweisungen READ_DBL und WRIT_DBL (Datenbaustein im Ladespeicher lesen/schreiben)

Tabelle 9-221 Anweisungen READ_DBL und WRIT_DBL

KOP/FUP	SCL	Beschreibung
	<pre>READ_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_);</pre>	<p>Kopiert DB-Startwerte oder einen Teil der Werte aus dem Ladespeicher in einen Ziel-DB im Arbeitsspeicher.</p> <p>Der Inhalt des Ladespeichers wird während des Kopiervorgangs nicht verändert.</p>
	<pre>WRIT_DBL(req:=_bool_in_, srcblk:=_variant_in_, busy=>_bool_out_, dstblk=>_variant_out_);</pre>	<p>Kopiert aktuelle DB-Werte oder einen Teil der Werte aus dem Arbeitsspeicher in einen Ziel-DB im Ladespeicher.</p> <p>Der Inhalt des Arbeitsspeichers wird während des Kopiervorgangs nicht verändert.</p>

Tabelle 9-222 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	BOOL
SRCBLK	IN	VARIANT
RET_VAL	OUT	INT
BUSY	OUT	BOOL
DSTBLK	OUT	VARIANT

Typischerweise wird ein DB sowohl im Ladespeicher (Flash) als auch im Arbeitsspeicher (RAM) gespeichert. Die Startwerte (Anfangswerte) werden immer im Ladespeicher gespeichert und die aktuellen Werte werden immer im Arbeitsspeicher gespeichert. Mit READ_DBL können Sie einen Satz Startwerte aus dem Ladespeicher in die aktuellen Werte eines DBs im Arbeitsspeicher kopieren, der von Ihrem Programm referenziert wird. Mit WRIT_DBL können Sie die im internen

Ladespeicher oder auf einer Memory Card gespeicherten aktuellen Werte im Arbeitsspeicher aktualisieren.

Hinweis**Auswirkung der Anweisungen WRIT_DBL und READ_DBL auf den Flash-Speicher**

Die Anweisung WRIT_DBL führt Schreibvorgänge im Flash-Speicher durch (im internen Ladespeicher oder auf einer Memory Card). Um eine Reduzierung der Lebensdauer des Flash-Speichers zu verhindern, verwenden Sie die Anweisung WRIT_DBL für weniger häufige Aktualisierungen wie das Aufzeichnen von Änderungen an einem Produktionsprozess. Vermeiden Sie aus ähnlichen Gründen häufige Aufrufe von READ_DBL für Lesevorgänge.

Sie müssen die Datenbausteine für READ_DBL und WRIT_DBL vor dem Aufrufen dieser Anweisungen im STEP 7-Programm erstellen. Wenn Sie den Quell-DB als Baustein des Typs "Standard" erstellt haben, muss der Ziel-DB ebenfalls vom Typ "Standard" sein. Wenn der Quelldatenbaustein als Baustein des Typs "Optimiert" erstellt wurde, muss der Zieldatenbaustein ebenfalls vom Typ "Optimiert" sein.

Wenn es sich um Standard-DBs handelt, können Sie entweder einen Variablennamen oder einen P#-Wert angeben. Mit dem P#-Wert können Sie eine beliebige Anzahl von Elementen der angegebenen Größe (Byte, Word oder DWord) angeben und kopieren. So können Sie einen Teil eines DBs oder den gesamten DB kopieren. Handelt es sich um optimierte DBs, können Sie lediglich einen Variablennamen angeben. Den Operator P# können Sie nicht verwenden. Wenn Sie für einen Standard-DB oder einen optimierten DB (oder für andere Arten von Arbeitsspeichern) einen Variablennamen angeben, kopiert die Anweisung die Daten, auf die dieser Variablenname verweist. Hierbei kann es sich um einen benutzerdefinierten Typ, ein Array oder ein grundlegendes Element handeln. Wenn es sich um einen Standard-DB und nicht um einen optimierten DB handelt, können Sie bei diesen Anweisungen nur den Datentyp Struct verwenden. Sie müssen einen benutzerdefinierten Datentyp (UDT) verwenden, wenn es sich um eine Struktur im optimierten Speicher handelt. Nur mit einem benutzerdefinierten Typ wird sichergestellt, dass die "Datentypen" sowohl für die Quellstruktur als auch für die Zielstruktur identisch sind.

Hinweis**Struktur (Datentyp Struct) in einem "optimierten" DB verwenden**

Wenn Sie einen Datentyp Struct mit "optimierten" DBs verwenden, müssen Sie zunächst einen benutzerdefinierten Datentyp (UDT) für den Datentyp Struct erstellen. Dann konfigurieren Sie den Quell- und Ziel-DB mit dem Datentyp UDT. Der Datentyp UDT gewährleistet, dass die Datentypen im Datentyp Struct für beide DBs konsistent bleiben.

Bei "Standard"-DBs verwenden Sie den Datentyp Struct, ohne einen Datentyp UDT zu erstellen.

READ_DBL und WRIT_DBL werden asynchron zum Programmzyklus ausgeführt. Die Verarbeitung erstreckt sich über mehrere Aufrufe von READ_DBL und WRIT_DBL. Sie starten den DB-Übertragungsauftrag durch Aufruf von REQ = 1 und überwachen anschließend die

Ausgänge BUSY und RET_VAL, um zu ermitteln, wann die Datenübertragung beendet und korrekt ist.

Hinweis

Auswirkung der Anweisungen WRIT_DBL und READ_DBL auf die Kommunikationslast

Wenn die Anweisung WRIT_DBL oder READ_DBL kontinuierlich aktiv ist, kann sie dabei so viele Kommunikationsressourcen verbrauchen, dass STEP 7 die Kommunikation mit der CPU verliert. Verwenden Sie aus diesem Grund für den REQ-Parameter eine positive Flanke am Eingang (Seite 214) und keinen Schließer- oder Öffnereingang (Seite 209), der mehrere Zyklen lang eingeschaltet bleiben würde (Signalpegel High).

Um Datenkonsistenz sicherzustellen, ändern Sie den Zielbereich während der Bearbeitung von READ_DBL bzw. den Quellbereich während der Bearbeitung von WRIT_DBL nicht (d.h. solange der Parameter BUSY WAHR ist).

Einschränkungen bei den Parametern SRCBLK und DSTBLK:

- Ein Datenbaustein muss, damit er referenziert werden kann, zuvor erstellt worden sein.
- Die Länge eines VARIANT-Pointer des Typs BOOL muss durch 8 teilbar sein.
- Die Länge eines VARIANT-Pointer des Typs STRING muss in Quell- und Zielpointer identisch sein.

Informationen zu Rezepten und zur Maschineneinrichtung

Mit den Anweisungen READ_DBL und WRIT_DBL können Sie Rezepte oder Informationen für die Maschineneinrichtung verwalten. Dies ist prinzipiell eine weitere Vorgehensweise, um für Werte, die sich nicht sehr häufig ändern, remanente Daten zu speichern. Sie sollten jedoch die Anzahl der Schreibvorgänge begrenzen, um den Flash-Speicher nicht übermäßig zu beanspruchen. Sie können auf diese Weise den Umfang des remanenten Speichers über den normalen Umfang der remanenten Daten hinaus erweitern, zumindest für Werte, die sich nicht häufig ändern. Sie können mit der Anweisung WRIT_DBL Rezeptinformationen oder Informationen für die Maschineneinrichtung aus dem Arbeitsspeicher im Ladespeicher speichern, und Sie können diese Informationen mit der Anweisung READ_DBL wieder aus dem Ladespeicher in den Arbeitsspeicher abrufen.

Tabelle 9-223 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
0081	Warnung: Der Quellbereich ist kleiner als der Zielbereich. Die Quelldaten werden vollständig kopiert, wobei die zusätzlichen Bytes im Zielbereich nicht verändert werden.
7000	Aufruf mit REQ = 0: BUSY = 0
7001	Erster Aufruf mit REQ = 1 (in Bearbeitung): BUSY = 1
7002	N ^{ter} Aufruf (in Bearbeitung): BUSY = 1
8051	Typfehler Datenbaustein
8081	Der Quellbereich ist größer als der Zielbereich. Der Zielbereich wird vollständig belegt und die verbleibenden Bytes der Quelle werden ignoriert.

RET_VAL (W#16#...)	Beschreibung
8251	Typfehler Quelldatenbaustein
82B1	Fehlender Quelldatenbaustein
82C0	Der Quell-DB wird von einer anderen Anweisung oder Kommunikationsfunktion bearbeitet.
8551	Typfehler Zieldatenbaustein
85B1	Fehlender Zieldatenbaustein
85C0	Der Ziel-DB wird von einer anderen Anweisung oder Kommunikationsfunktion bearbeitet.
80C3	Mehr als 50 READ_DBL- oder 50 WRIT_DBL-Anweisungen befinden sich derzeit in der Warteschlange für die Ausführung.

Siehe auch Rezepte (Seite 482)

9.10.3 ATTR_DB (Attribute eines Datenbausteins lesen)

Tabelle 9-224 Anweisung ATTR_DB

KOP/FUP	SCL	Beschreibung
<p style="text-align: center;">ATTR_DB</p> <ul style="list-style-type: none"> - EN ENO - REQ RET_VAL - DB_NUMBER DB_LENGTH <li style="padding-left: 150px;">ATTRIB 	<pre>ret_val := ATTR_DB(REQ:=_bool_in_, DB_NUMBER:=_uint_in_, DB_LENGTH=>_udint_out_, ATTRIB=>_byte_out_);</pre>	<p>Mit der Anweisung "ATTR_DB" rufen Sie Informationen über einen Datenbaustein (DB) im Arbeitsspeicher der CPU ab. Die Anweisung ermittelt die am Parameter ATTRIB für den ausgewählten DB festgelegten Attribute.</p> <p>Die Länge kann für Datenbausteine mit optimierten Zugriff und für Datenbausteine, die sich nur im Ladespeicher befinden, nicht ausgelesen werden. In diesen Fällen hat der Parameter DB_LENGTH den Wert 0.</p> <p>Wenden Sie ATTR_DB nicht auf Datenbausteine mit optimierten Zugriff und aktivierter Speicherreserve an.</p> <p>Lesen Sie die Datenbausteine für Bewegungssteuerung nicht mit der Anweisung "ATTR_DB" aus. Dabei wird der Fehlercode 80B2 ausgegeben.</p>

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "ATTR_DB":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L oder Konstante	REQ = 1: Anforderung zum Lesen von Bausteinattributen
DB_NUMBER	Input	DB_ANY	E, A, M, D, L oder Konstante	Nummer des zu testenden DBs
RET_VAL	Output	INT	E, A, M, D, L	Fehlerinformationen

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
DB_LENGTH	Output	UDINT	E, A, M, D, L	<ul style="list-style-type: none"> Anzahl von Datenbytes, die der ausgewählte DB enthält 0 bei Datenbausteinen mit optimierten Zugriff und Datenbausteinen, die sich nur im Ladespeicher befinden
ATTRIB	Output	BYTE	E, A, M, D, L	<p>DB-Eigenschaften:</p> <ul style="list-style-type: none"> Bit 0* = 0: Attribut "Nur im Ladespeicher ablegen" ist nicht festgelegt. Bit 0* = 1: Attribut "Nur im Ladespeicher ablegen" ist festgelegt. Bit 1 = 0: Das Attribut "Datenbaustein im Gerät schreibgeschützt" ist nicht festgelegt. Bit 1 = 1: Das Attribut "Datenbaustein im Gerät schreibgeschützt" ist festgelegt. <p>Wenn Bit 0 = 1, ist Bit 2 irrelevant und erhält den Wert 1.</p> <ul style="list-style-type: none"> Bit 2 = 0: Remanent - Der DB wird als remanent betrachtet, wenn mindestens ein Wert als remanent definiert wurde. Bit 2 = 1: Nicht remanent - Der gesamte DB ist nicht remanent. Bit 3* = 0: Der DB ist entweder im Ladespeicher (Bit 0 = 1) oder im Arbeitsspeicher (Bit 0 = 0). Bit 3* = 1: Der DB wird sowohl im Ladespeicher als auch im Arbeitsspeicher erstellt.
<p>* Die Beziehung zwischen Bit 0 und Bit 3 wird in den Parametern der Anweisung "CREATE_DB (Datenbaustein erstellen) (Seite 518)" erläutert.</p>				

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Parameter RET_VAL

Fehlercode* (W#16#...)	Bedeutung
0000	Kein Fehler.
80A1	<p>Fehler im Eingangsparameter DB_NUMBER: der tatsächlich ausgewählte Parameter</p> <ul style="list-style-type: none"> Ist 0 Ist größer als die maximal zulässige DB-Nummer für die verwendete CPU
80B1	Der DB mit der angegebenen Nummer ist in der CPU nicht vorhanden.
80B2	Datenbausteine von Technologieobjekten für die Bewegungssteuerung können mit der Anweisung "ATTR_DB" nicht gelesen werden.
Allgemeine Fehlerinformationen	Siehe auch: Gemeinsame Fehlercodes für die erweiterten Anweisungen (Seite 535)
*Fehlercodes können im Programmierer als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden.	

9.10.4 DELETE_DB (Datenbaustein löschen)

Tabelle 9-225 Anweisung DELETE_DB

KOP/FUP	SCL	Beschreibung						
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center; margin: 0;">DELETE_DB</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: none;">- EN</td> <td style="border: none; text-align: right;">ENO</td> </tr> <tr> <td style="border: none;">- REQ</td> <td style="border: none; text-align: right;">Ret_Val</td> </tr> <tr> <td style="border: none;">- DB_NUMBER</td> <td style="border: none; text-align: right;">BUSY</td> </tr> </table> </div>	- EN	ENO	- REQ	Ret_Val	- DB_NUMBER	BUSY	<pre>ret_val := DELETE_DB(REQ := _bool_in_, DB_NUMBER := _uint_in_, BUSY => _bool_out_);</pre>	<p>Mit der Anweisung "DELETE_DB" löschen Sie einen Datenbaustein (DB), den das Anwenderprogramm durch Aufrufen der Anweisung "CREATE_DB (Seite 518)" erstellt hat.</p> <p>Wenn der Datenbaustein nicht mit "CREATE_DB" erstellt wurde, gibt DELETE_DB am Parameter RET_VAL den Fehlercode W#16#80B5 aus.</p> <p>Der Aufruf von DELETE_DB löscht den ausgewählten Datenbaustein nicht sofort, sondern nach der Ausführung des Zyklus-OB am Zykluskontrollpunkt.</p>
- EN	ENO							
- REQ	Ret_Val							
- DB_NUMBER	BUSY							

Funktionsbeschreibung

Die Anweisung "DELETE_DB" funktioniert asynchron, das heißt, ihre Ausführung erstreckt sich über mehrere Aufrufe. Sie starten die Alarmübertragung durch Aufrufen der Anweisung mit REQ = 1.

Der Ausgangsparameter BUSY und die Bytes 2 und 3 von Ausgangsparameter RET_VAL zeigen den Status des Auftrags.

Der Datenbaustein ist vollständig gelöscht, wenn der Ausgangsparameter BUSY den Wert FALSE hat.

Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung "DELETE_DB":

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L oder Konstante	REQ = 1: Anforderung zum Löschen des DBs mit der Nummer im Parameter DB_NUMBER
DB_NUMBER	Input	UINT	E, A, M, D, L oder Konstante	Nummer des DB, der gelöscht werden soll
RET_VAL	Output	INT	E, A, M, D, L	Fehlerinformationen (siehe Parameter "RET_VAL")
BUSY	Output	BOOL	E, A, M, D, L	BUSY= 1: Der Vorgang ist noch nicht beendet.

Weitere Informationen zu gültigen Datentypen finden Sie unter "Datentypen (Seite 105)".

Parameter RET_VAL

Fehlercode* (W#16#...)	Bedeutung
0000	Kein Fehler.
7000	Erster Aufruf mit REQ = 0: Keine Datenübertragung aktiv; BUSY hat den Wert 0.
7001	Erster Aufruf mit REQ = 1: Datenübertragung angestoßen; BUSY hat den Wert 1.


Fehlercode* (W#16#...)	Bedeutung
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Datenübertragung bereits aktiv; BUSY hat den Wert 1.
80A1	Fehler im Eingangsparameter DB_NUMBER: <ul style="list-style-type: none"> • Der Wert am Parameter ist 0. • Der Wert am Parameter ist größer als die maximal zulässige DB-Nummer für die verwendete CPU.
80B1	Der DB mit der angegebenen Nummer ist in der CPU nicht vorhanden.
80B4	Der DB kann nicht gelöscht werden, weil die Memory Card der CPU schreibgeschützt ist.
80B5	Der DB wurde nicht mit "CREATE_DB" erstellt.
80BB	Nicht genügend Ladespeicher.
80C3	Die Funktion "DB löschen" kann gegenwärtig wegen eines kurzzeitigen Ressourcenengpasses nicht ausgeführt werden.
Allgemeine Fehlerinformationen	Siehe auch: Gemeinsame Fehlercodes für die erweiterten Anweisungen (Seite 535)
*Fehlercodes können im Programmeditor als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden.	

9.11 Adressverarbeitung

9.11.1 GEO2LOG (Aus dem Steckplatz die Hardwarekennung ermitteln)

Mit Anweisung GEO2LOG wird die Hardwarekennung anhand der Steckplatzinformationen ermittelt.

Tabelle 9-226 Anweisung GEO2LOG

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := GEO2LOG(GEOADDR:=_variant_in_out_, laddr:=_word_out_);</pre>	<p>Mit Anweisung GEO2LOG wird die Hardwarekennung anhand der Steckplatzinformationen ermittelt.</p>

Mit Anweisung GEO2LOG wird die Hardwarekennung anhand der Steckplatzinformationen ermittelt, die mit dem Systemdatentyp GEOADDR festgelegt werden:

Je nach dem Hardwaretyp, der in Parameter HWTYPE festgelegt ist, werden die folgenden Informationen der anderen GEOADDR-Parameter ausgewertet:

- Bei HWTYPE = 1 (PROFINET IO-System):
 - Nur IOSYSTEM wird ausgewertet. Die anderen Parameter von GEOADDR werden nicht berücksichtigt.
 - Die Hardwarekennung des PROFINET IO-Systems wird ausgegeben.
- Bei HWTYPE = 2 (PROFINET IO-Device):
 - IOSYSTEM und STATION werden ausgewertet. Die anderen Parameter von GEOADDR werden nicht berücksichtigt.
 - Die Hardwarekennung des PROFINET IO-Device wird ausgegeben.
- Bei HWTYPE = 3 (Baugruppenträger):
 - Nur IOSYSTEM und STATION werden ausgewertet. Die anderen Parameter von GEOADDR werden nicht berücksichtigt.
 - Die Hardwarekennung des Baugruppenträgers wird ausgegeben.
- Bei HWTYPE = 4 (Modul):
 - IOSYSTEM, STATION, und SLOT werden ausgewertet. Parameter SUBSLOT von GEOADDR wird nicht berücksichtigt.
 - Die Hardwarekennung des Moduls wird ausgegeben.
- Bei HWTYPE = 5 (Submodul):
 - Alle Parameter von GEOADDR werden ausgewertet.
 - Die Hardwarekennung des Submoduls wird ausgegeben.

Parameter AREA des Systemdatentyps GEOADDR wird nicht ausgewertet.

Tabelle 9-227 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
GEOADDR	IN/OUT or IN ?	Variant	Pointer auf die Struktur des Systemdatentyps GEOADDR. Der Systemdatentyp GEOADDR enthält die Steckplatzinformationen, aus denen die Hardware-ID ermittelt wird. Weitere Informationen finden Sie unter "Systemdatentyp GEOADDR (Seite 534)".
RET_VAL	OUT or RETURN ?	Int	Ausgabe der Fehlerinformationen.
LADDR	OUT	HW_ANY	Hardwarekennung des Bauteils oder des Moduls. Die Nummer wird automatisch zugewiesen und ist in den Eigenschaften der Hardwarekonfiguration gespeichert.

Weitere Informationen über gültige Datentypen finden Sie unter "Übersicht der gültigen Datentypen" in der STEP 7 Online-Hilfe.

Tabelle 9-228 Bedingungscode

RET_VAL* (W#16#...)	Erklärung
0	Kein Fehler.
8091	Ungültiger Wert in GEOADDR für HWTYPE.
8094	Ungültiger Wert in GEOADDR für IOSYSTEM.
8095	Ungültiger Wert in GEOADDR für STATION.
8096	Ungültiger Wert in GEOADDR für SLOT.
8097	Ungültiger Wert in GEOADDR für SUBSLOT.

* Die Fehlercodes können als ganzzahlige oder als Hexadezimalwerte im Programmierer angezeigt werden.

9.11.2 LOG2GEO (Steckplatz über Hardwareerkennung ermitteln)

Mit der Anweisung LOG2GEO ermitteln Sie die geografische Adresse (Modulsteckplatz) aus der logischen Adresse einer Hardwareerkennung.

Tabelle 9-229 Anweisung LOG2GEO

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := LOG2GEO(laddr:=_word_in_, GEOADDR:=_variant_in_out_) ;</pre>	<p>Mit der Anweisung LOG2GEO ermitteln Sie den zu einer Hardwareerkennung gehörigen Modulsteckplatz.</p>

Die Anweisung LOG2GEO bestimmt die geografische Adresse einer logischen Adresse anhand der Hardwareerkennung:

- Mit dem Parameter LADDR wählen Sie die logische Adresse anhand der Hardwareerkennung aus.
- GEOADDR enthält die geografische Adresse der am Eingang LADDR angegebenen logischen Adresse.

Hinweis

Wenn der Typ HW eine Komponente nicht unterstützt, wird eine Untersteckplatznummer für ein Modul 0 ausgegeben.

Ein Fehler wird gemeldet, wenn der Eingang LADDR kein HW-Objekt adressiert.

Tabelle 9-230 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
LADDR	IN	HW_ANY	Hardwarekennung des IO-Systems oder des Moduls. Die Nummer wird automatisch zugewiesen und in den Eigenschaften der CPU oder der Schnittstelle der Hardwarekonfiguration gespeichert.
RET_VAL	OUT	Int	Fehlercode der Anweisung
GEOADDR	IN_OUT	Variante	Pointer auf den Systemdatentyp GEOADDR. Der Systemdatentyp GEOADDR enthält die Steckplatzinformationen. Weitere Informationen finden Sie unter "Systemdatentyp GEOADDR (Seite 534)".

Für weitere Informationen über gültige Datentypen, siehe "Übersicht der gültigen Datentypen" in der STEP 7 Online-Hilfe.

Tabelle 9-231 Bedingungscode

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Die im Parameter LADDR angegebene Adresse ist ungültig.
* Die Fehlercodes können als ganzzahlige oder als Hexadezimalwerte im Programmierer angezeigt werden.	

9.11.3 IO2MOD (Aus einer E/A-Adresse die Hardwarekennung ermitteln)

Mit Anweisung IO2MOD wird die Hardwarekennung eines Moduls anhand der E/A-Adresse eines Submoduls ermittelt.

Tabelle 9-232 Anweisung IO2MOD

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := IO2MOD(ADDR:=_word_in_, LADDR:=_word_out_);</pre>	<p>Mit der Anweisung IO2MOD ermitteln Sie den zu einer Hardwarekennung gehörigen Modulsteckplatz.</p>

Mit IO2MOD wird die Hardwarekennung des Moduls anhand einer E/A-Adresse (E, A, PE, PA) eines Submoduls ermittelt.

Geben Sie die E/A-Adresse in Parameter ADDR ein. Werden für diesen Parameter mehrere E/A-Adressen verwendet, so wird nur die erste Adresse für die Ermittlung der Hardwarekennung ausgewertet. Wird die erste Adresse richtig angegeben, so ist die Länge für die Adressangabe in ADDR nicht von Belang. Wird ein Adressbereich angegeben, der mehrere Module oder unbenutzte Adressen umfasst, kann auch die Hardwarekennung des ersten Moduls ermittelt werden.

Wird in Parameter ADDR keine E/A-Adresse eines (Sub-)Moduls angegeben, wird am Parameter RET_VAL der Fehlercode "8090" ausgegeben.

Hinweis

Eingabe der E/A-Adresse in SCL

Mit der E/A-Zugriffs-ID "%QWx:P" ist keine Programmierung in SCL möglich. In diesem Fall ist der symbolische Variablenname oder die absolute Adresse im Prozessbild zu verwenden.

Tabelle 9-233 Datentypen für die Parameter

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ADDR	IN or IN/OUT ?	Variant	E, A, M, D, L	E/A-Adresse (E, A, PE, PA) in einem Submodul. Achten Sie darauf, dass für Parameter ADDR kein Slice-Zugriff verwendet wird. In diesem Fall würden an Parameter LADDR fehlerhafte Werte ausgegeben.
RET_VAL	OUT or RETURN ?	Int	E, A, M, D, L	Fehlercode der Anweisung.
LADDR	OUT	HW_IO	E, A, M, D, L	Ermittelte Hardwarekennung (logische Adresse) des E/A-Submoduls.

Weitere Informationen über gültige Datentypen finden Sie unter "Übersicht der gültigen Datentypen" in der STEP 7 Online-Hilfe.

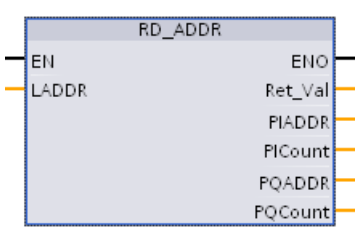
Tabelle 9-234 Bedingungscode

RET_VAL* (W#16#...)	Erklärung
0	Kein Fehler.
8090	Die in Parameter ADDR angegebene E/A-Adresse wird von keiner Hardwarekomponente benutzt.
* Die Fehlercodes können als ganzzahlige oder als Hexadezimalwerte im Programmeditor angezeigt werden.	

9.11.4 RD_ADDR (E/A-Adressen über Hardwarekennung ermitteln)

Mit der Anweisung RD_ADDR rufen Sie die E/A-Adressen eines Submoduls ab.

Tabelle 9-235 Anweisung RD_ADDR

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := RD_ADDR(laddr:=_word_in_, PIADDR=>_uint_out_, PICount=>_uint_out_, PQADDR=>_uint_out_, PQCount=>_uint_out_,);</pre>	<p>Mit der Anweisung RD_ADDR rufen Sie die E/A-Adressen eines Submoduls ab.</p>

Die Anweisung RD_ADDR ermittelt die Länge und die Anfangsadresse der Eingänge oder Ausgänge anhand der Hardwarekennung eines Submoduls:

- Mit dem Parameter LADDR wählen Sie das Eingangs- oder Ausgangsmodul anhand der Hardwarekennung aus.
- Abhängig davon, ob es sich um ein Eingangs- oder ein Ausgangsmodul handelt, werden die folgenden Ausgangsparameter verwendet:
 - Bei einem Eingangsmodul werden die ermittelten Werte an den Parametern PIADDR und PICOUNT ausgegeben.
 - Bei einem Ausgangsmodul werden die ermittelten Werte an den Parametern PQADDR und PQCOUNT ausgegeben.
- Die Parameter PIADDR und PQADDR enthalten jeweils die Anfangsadresse der E/A-Adressen des Moduls.
- Die Parameter PICOUNT und PQCOUNT enthalten jeweils die Anzahl der Bytes der Eingänge oder Ausgänge (1 Byte für 8 Eingänge/Ausgänge, 2 Bytes für 16 Eingänge/Ausgänge).

Tabelle 9-236 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
LADDR	IN	HW_IO
RET_VAL	OUT	Int
PIADDR	OUT	UDInt
PICOUNT	OUT	UInt
PQADDR	OUT	UDInt
PQCOUNT	OUT	UInt

Weitere Informationen über gültige Datentypen finden Sie unter "Übersicht der gültigen Datentypen" in der STEP 7 Online-Hilfe.

Tabelle 9-237 BedingungsCodes

RET_VAL (W#16#...)	Beschreibung
0000	Kein Fehler
8090	Die Hardwarekennung des Moduls am Parameter LADDR ist ungültig.
* Die Fehlercodes können als ganzzahlige oder als Hexadezimalwerte im Programmierer angezeigt werden.	

9.11.5 Systemdatentyp GEOADDR

Geografische Adresse

Der Systemdatentyp GEOADDR enthält die geografische Adresse eines Moduls (oder die Steckplatzinformationen).

- Geografische Adresse für PROFINET IO:
Für PROFINET IO besteht die geografische Adresse aus der ID des PROFINET IO-Systems, der Gerätenummer, der Steckplatznummer und dem Submodul (wenn ein Submodul verwendet wird).
- Geografische Adresse für PROFIBUS DP:
Für PROFIBUS DP besteht die geografische Adresse aus der ID des DP-Mastersystems, der Stationsnummer und der Steckplatznummer.

Die Steckplatzinformation der Module ist in der Hardwarekonfiguration jedes Moduls zu finden.

Struktur des Systemdatentyps GEOADDR

Die Struktur GEOADDR wird automatisch erzeugt, wenn als Datentyp "GEOADDR" im Datenbaustein eingegeben wird.

Parametername	Datentyp	Beschreibung
GEOADDR	STRUCT	
HWTYPE	UINT	Hardwaretyp: <ul style="list-style-type: none"> • 1: IO-System (PROFINET/PROFIBUS) • 2: IO-Device/DP-Slave • 3: Baugruppenträger • 4: Modul • 5: Submodul Wird ein Hardwaretyp von der Anweisung nicht unterstützt, wird der HWTYPE "0" ausgegeben.
AREA	UINT	Bereichskennung: <ul style="list-style-type: none"> • 0 = CPU • 1 = PROFINET IO • 2 = PROFIBUS DP • 3 = AS-i
IOSYSTEM	UINT	PROFINET IO-System (0 = zentrale Einheit im Baugruppenträger)
STATION	UINT	<ul style="list-style-type: none"> • Nummer des Baugruppenträgers, wenn die Bereichskennung AREA = 0 (Zentralmodul). • Stationsnummer, wenn die Bereichskennung AREA > 0.
SLOT	UINT	Steckplatznummer
SUBSLOT	UINT	Nummer des Submoduls. Dieser Parameter hat den Wert "0", wenn kein Submodul verfügbar oder steckbar ist.

9.12 Gemeinsame Fehlercodes für die erweiterten Anweisungen

Tabelle 9-238 Gemeinsame Bedingungscode für die erweiterten Anweisungen

Bedingungscode (W#16#....) ¹	Beschreibung
8x22 ²	Bereich für Eingang zu klein
8x23	Bereich für Ausgang zu klein
8x24	Unzulässiger Eingangsbereich
8x25	Unzulässiger Ausgangsbereich
8x28	Unzulässige Eingangsbitzuweisung
8x29	Unzulässige Ausgangsbitzuweisung
8x30	Ausgangsbereich ist ein schreibgeschützter DB.
8x3A	DB ist nicht vorhanden.

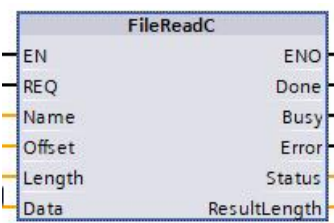
- ¹ Tritt einer dieser Fehler während der Ausführung eines Codebausteins auf, bleibt die CPU im Betriebszustand RUN (Voreinstellung) oder geht in STOP, wenn dies entsprechend konfiguriert wurde. Optional kann mit den Anweisungen GetError oder GetErrorID in diesem Codebaustein die lokale Fehlerbearbeitung (CPU bleibt in RUN) veranlasst und eine programmierte Reaktion auf den Fehler erzeugt werden.
- ² Das "x" steht für die Nummer des Parameters mit dem Fehler. Parameternummern beginnen mit 1.

9.13 Handhabung von Dateien

9.13.1 FileReadC: Datei von Memory Card lesen

Mit der Anweisung "FileReadC" lesen Sie Daten aus einer Datei auf der Memory Card und schreiben sie in einen Zielbereich auf der CPU. Sie geben die Datei anhand ihres Namens und des vollständigen Pfads an.

Tabelle 9-239 Anweisung FileReadC

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung "FileReadC" lesen Sie Daten aus einer Datei auf der Memory Card und schreiben sie in einen Zielbereich auf der CPU.</p>

Das Lesen aus der Datei aktivieren Sie mit dem Steuerparameter REQ. Die Ausgangsparameter Done, Busy, Error und Status geben den Auftragsstatus an.

Die folgenden Regeln gelten für die Angabe von Dateiname und Pfad:

- Der Dateiname darf maximal 55 Zeichen lang sein.
- Folgende Zeichen sind für den Ordner und den Dateinamen zulässig: "0" bis "9", "a" bis "z", "A" bis "Z", "-", "_"
- Der Dateiname darf maximal einen Punkt (".") enthalten. Der Punkt trennt den Namen von der Dateierweiterung. Der Dateiname muss mindestens ein Zeichen enthalten. Eine Dateierweiterung ist nicht notwendig.
- Der Pfadname darf nicht mit "/", "\" oder "." beginnen.
- Der Pfadname darf keine ".." enthalten.
- Der Pfadname darf unter dem Ordner "UserFiles" keinen Unterordner enthalten.

Beispiele für gültige Pfade und Dateinamen: "UserFiles/Lift16_DataBase.txt", "UserFiles/2017-04-13_ErrorLog.bin"

Mit FileReadC können Sie bis zu 16 MB (16.777.216 Bytes) lesen. Die Daten werden Segment für Segment gelesen. Die Segmentgröße ist gerätespezifisch. Eine S7-1500 CPU beispielsweise verwendet Blöcke von 32 KB (32.768 Byte). Wenn die Anzahl der zu lesenden Datenelemente größer ist als die Segmentgröße, müssen Sie die Anweisung in Ihrem Programm mehrere Male aufrufen. Aus Konsistenzgründen sollten Sie deshalb erst dann auf die gelesenen Daten zugreifen, nachdem die Anweisung zum letzten Mal aufgerufen wurde.

Die Parameter "Offset" und "Length" legen die Länge der zu lesenden Datenelemente fest. Die beschäftigte Ressource wird wieder freigegeben, sobald der Lesevorgang abgeschlossen ist.

Die Anweisung "FileReadC" arbeitet asynchron. Die Verarbeitung erfolgt über mehrere Aufrufe. Die Verarbeitung beginnt mit einer steigenden Flanke am Parameter "REQ".

Die Parameter "Busy" und "Done" geben den Auftragsstatus an.

Tritt während der Ausführung ein Fehler auf, wird dies von den Parametern "Error" und "Status" angezeigt.

Tabelle 9-240 Datentypen für die Parameter

Parameter und Typ		Datentyp	Beschreibung
REQ	Input	BOOL	Steuerparameter REQUEST Ermöglicht das Lesen einer Datei von der Memory Card bei einer steigenden Flanke
Name	Input	STRING	Name der zu lesenden Datei mit vollständigem Pfad
Offset	Input	UDINT	Byte-Offset, nach dem die Datei gelesen werden soll
Length	Input	UDINT	Länge des zu lesenden Bereichs in Byte Length = 0 bedeutet, dass die maximal mögliche Anzahl von Datenelementen pro Aufruf gelesen wird (bei einer S7-1200 CPU sind das 8 KB oder die Größe des Parameters "Data").
Done	Output	BOOL	1: Die Anweisung wurde erfolgreich ausgeführt. Die gelesenen Informationen wurden zum Parameter "Data" übertragen.

Parameter und Typ		Datentyp	Beschreibung
Busy	Output	BOOL	Statusparameter <ul style="list-style-type: none"> 0: Die Ausführung der Anweisung ist beendet oder noch nicht gestartet. 1: Die Ausführung der Anweisung ist noch nicht beendet.
Error	Output	BOOL	Statusparameter <ul style="list-style-type: none"> 0: Kein Fehler. 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen werden am Parameter "Status" ausgegeben.
Status	Output	WORD	Fehlercode
ResultLength	Output	UDINT	Länge der gelesenen Daten in Byte
Data	InOut	VARIANT	Zielbereich für die gelesenen Daten Zulässige Datentypen: BYTE und Array aus BYTE

Tabelle 9-241 Bedingungscode

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler
7000	Keine laufende Auftragsbearbeitung
7001	Start der Auftragsbearbeitung. Parameter Busy = 1, Done = 0.
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; Busy hat den Wert 1.
0081	Warnung: Offset + Length ist größer als die Länge der Datei. Daten werden ab "Offset" bis zum Ende der Datei gelesen. "Data" enthält weniger Datenelemente als gefordert. "ResultLength" ist kürzer als "Length". Daten nach "ResultLength" in "Data" werden nicht verändert.
8091	Pfad ist nicht vorhanden oder ungültig.
8092	Der Parameter "Name" hat nicht den Datentyp "STRING", ist zu lang oder enthält ungültige Zeichen.
8093	Der Parameter "Offset" zeigt auf einen Punkt über das Ende der zu lesenden Datei hinaus.
8094	Der Parameter "Length" ist größer als der maximal zulässige Wert. Bei einer S7-1500 oder S7-1200 CPU beispielsweise ist der maximal zulässige Wert 16 MB, also 16.777.216 Bytes.
80A1	Lesefehler. Der vom Parameter "Data" angegebene Zielbereich wird möglicherweise teilweise überschrieben.
80B1	Der vom Parameter "Data" angegebene Zielbereich ist kürzer als die vom Parameter "Length" geforderte Länge.
80C0	Auf die Datei kann nicht zugegriffen werden (schreibgeschützt oder von einem anderen Prozess gesperrt).
80C3	Die maximale Anzahl gleichzeitig aktiver Anweisungen "FileReadC" ist bereits erreicht.
8A30	Der Zielbereich ist schreibgeschützt, beispielsweise ein schreibgeschützter DB.
8A3A	Der Parameter "Data" zeigt auf einen unzulässigen Bereich, beispielsweise auf den Ladespeicher oder die Lokaldaten.
8A51	Ungültiger Datentyp des Parameters "Data".
8A52	Die Variable am Parameter Data ist nicht ausreichend. Ein Teil der Quellbereichsdaten wurde möglicherweise geschrieben.
* Die Fehlercodes werden im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".	

Siehe auch

Neue Funktionen (Seite 35)

9.13.2 FileWriteC: Datei auf die Memory Card schreiben

Mit der Anweisung "FileWriteC" schreiben Sie Daten aus einem Quellbereich auf der CPU in eine Datei im Ordner "UserFiles" auf der Memory Card.

Tabelle 9-242 Anweisung FileWriteC

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung "FileWriteC" schreiben Sie Daten aus einem Quellbereich auf der CPU in eine Datei im Ordner "UserFiles" auf der Memory Card. Sie geben die Datei anhand ihres Namens und des vollständigen Pfads an. Falls die Datei nicht vorhanden ist, wird sie von der CPU im Ordner "UserFiles" erstellt. Ist der Ordner "UserFiles" ebenfalls nicht vorhanden, wird auch er von der CPU erstellt. Unterordner werden nicht erstellt. In diesem Fall wird am Parameter "Status" W#16#8091 ausgegeben.</p>

Das Schreiben in die Datei aktivieren Sie mit dem Steuerparameter "REQ". Die Ausgangsparameter "Done", "Busy", "Error" und "Status" geben den Auftragsstatus an.

Die folgenden Regeln gelten für die Angabe von Dateiname und Pfad:

- Der Dateiname darf maximal 55 Zeichen lang sein.
- Folgende Zeichen sind für den Ordner und den Dateinamen zulässig: "0" bis "9", "a" bis "z", "A" bis "Z", "-", "_"
- Der Dateiname darf maximal einen Punkt (".") enthalten. Der Punkt trennt den Namen von der Dateierweiterung. Der Dateiname muss mindestens ein Zeichen enthalten. Eine Dateierweiterung ist nicht notwendig.
- Der Pfadname darf nicht mit "/", "\" oder "." beginnen.
- Der Pfadname darf keine ".." enthalten.
- Der Pfadname darf unter dem Ordner "UserFiles" keinen Unterordner enthalten.

Beispiele für gültige Pfade und Dateinamen: "UserFiles/Lift16_DataBase.txt", "UserFiles/2017-04-13_ErrorLog.bin"

Mit "FileWriteC" können Sie bis zu 16 MB (16.777.216 Bytes) schreiben. Die Daten werden Segment für Segment geschrieben. Die Segmentgröße ist gerätespezifisch. Eine S7-1200 CPU beispielsweise verwendet Blöcke von 8 KB (8192 Byte). Wenn die Anzahl der zu schreibenden Datenelemente größer ist als die Segmentgröße, müssen Sie die Anweisung in Ihrem Programm mehrere Male aufrufen. Aus Konsistenzgründen sollten Sie deshalb erst dann auf die geschriebenen Daten zugreifen, nachdem die Anweisung zum letzten Mal aufgerufen wurde. Wenn die verfügbare Datei zu kurz ist, wird sie auf die erforderliche Größe erweitert.

Die Parameter "Offset" und "Length" geben den Speicherort in der Datei an, in den die Daten geschrieben werden sollen. Die beschäftigte Ressource wird wieder freigegeben, sobald der Schreibvorgang abgeschlossen ist.

Die Anweisung "FileWriteC" beginnt den Schreibvorgang erst, wenn die folgende Bedingung erfüllt ist: "Offset" + "Length" ≤ 16 MB.

Die Anweisung "FileWriteC" arbeitet asynchron. Die Verarbeitung erfolgt über mehrere Aufrufe. Die Verarbeitung beginnt mit einer steigenden Flanke am Parameter "REQ".

Die Parameter "Busy" und "Done" geben den Auftragsstatus an.

Tritt während der Ausführung ein Fehler auf, wird dies von den Parametern "Error" und "Status" angezeigt.

Tabelle 9-243 Datentypen für die Parameter

Parameter und Typ		Datentyp	Beschreibung
REQ	Input	BOOL	Steuerparameter REQUEST Ermöglicht das Schreiben einer Datei auf die Memory Card bei einer steigenden Flanke
Name	Input	STRING	Name der zu schreibenden Datei mit vollständigem Pfad
Offset	Input	UDINT	Byte-Offset, ab dem die Datei geschrieben werden soll
Length	Input	UDINT	Länge des zu schreibenden Bereichs in Byte "Length" = 0 bedeutet, dass der gesamte vom Parameter "Data" angegebene Quellbereich geschrieben wird.
Done	Output	BOOL	1: Die Anweisung wurde erfolgreich ausgeführt.
Busy	Output	BOOL	Statusparameter <ul style="list-style-type: none"> 0: Die Ausführung der Anweisung ist beendet oder noch nicht gestartet. 1: Die Ausführung der Anweisung ist noch nicht beendet.
Error	Output	BOOL	Statusparameter <ul style="list-style-type: none"> 0: Kein Fehler. 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen werden am Parameter "Status" ausgegeben.
Status	Output	WORD	Fehlercode
ResultLength	Output	UDINT	Länge der geschriebenen Daten in Byte
Data	InOut	VARIANT	Quellbereich Zulässige Datentypen: BYTE und Array aus BYTE

Tabelle 9-244 Bedingungscode

Fehlercode* (W#16#...)	Bedeutung
0	Kein Fehler
7000	Keine laufende Auftragsbearbeitung
7001	Start der Auftragsbearbeitung. Parameter Busy = 1, Done = 0.

Fehlercode* (W#16#...)	Bedeutung
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; Busy hat den Wert 1.
8091	Pfad ist nicht vorhanden oder ungültig.
8092	Der Parameter "Name" hat nicht den Datentyp "STRING", ist zu lang oder enthält ungültige Zeichen.
8093	<ul style="list-style-type: none"> Der Parameter "Offset" zeigt auf einen Punkt über das Ende der zu schreibenden Datei hinaus. Die Erstellung der Datei wird abgelehnt, weil "Offset" größer als null ist.
8094	<ul style="list-style-type: none"> "Length" ist größer als der maximal zulässige Wert. Bei einer S7-1500 CPU beispielsweise ist der maximal zulässige Wert 16 MB, also 16.777.216 Bytes. "Length" + "Offset" ist größer als der maximal zulässige Wert.
80A1	Schreibfehler. Die Daten in der Datei auf der Memory Card werden möglicherweise teilweise überschrieben.
80B1	Der vom Parameter "Data" angegebene Quellbereich ist kürzer als die vom Parameter "Length" geforderte Länge.
80B3	Nicht genügend Speicherplatz auf der Memory Card oder im internen Ladespeicher.
80B4	Die Memory Card oder Datei ist schreibgeschützt.
80C0	Auf die Datei kann nicht zugegriffen werden.
80C3	Die maximale Anzahl gleichzeitig aktiver Anweisungen "FileWriteC" ist bereits erreicht.
8A24	Der Parameter "Data" zeigt auf einen unzulässigen Bereich, beispielsweise auf den Ladespeicher oder die Lokaldaten.
8A51	Ungültiger Datentyp des Parameters "Data".
8A52	Die Variable am Parameter Data ist nicht ausreichend. Ein Teil der Quellbereichsdaten wurde möglicherweise geschrieben.
* Die Fehlercodes werden im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".	

Siehe auch

Neue Funktionen (Seite 35)

9.13.3 FileDelete: Datei auf der Memory Card löschen

Mit der Anweisung "FileDelete" löschen Sie eine vorhandene Datei von der Memory Card.

Tabelle 9-245 Anweisung FileDelete

KOP/FUP	SCL	Beschreibung
<pre> FileDelete - EN ENO - - REQ Done - - Name Busy - Error - Status - </pre>	<pre> "FileDelete_DB_1" (REQ:=_bool_in_, Name:=_string_in_, Done=>_bool_out_, Busy=>_bool_out_, Error=>_bool_out_, Status=>_word_out_) </pre>	<p>Mit der Anweisung "FileDelete" löschen Sie eine vorhandene Datei von der Memory Card.</p>

Die Datei darf nicht geöffnet sein. Platzhalter im Parameter "Name" werden nicht unterstützt, was bedeutet, dass Namen wie "UserFiles/*.txt" und "UserFiles/? .txt" nicht zulässig sind.

Die Anweisung "FileDelete" ist nur in den Ordnern "Recipes" und "UserFiles" zulässig. Ordner innerhalb dieser Ordner sind möglich, z.B. "UserFiles/Test/file1.txt".

Hinweis

Löschen von Datenprotokollen

Das Löschen einer Datei im Ordner "DataLog" mit der Anweisung "FileDelete" ist nicht zulässig. Zum Löschen von Datenprotokollen müssen Sie die Anweisung "DataLogDelete" verwenden.

"FileDelete" ist eine asynchrone Anweisung. Die Verarbeitung kann sich über mehrere Aufrufe erstrecken. Die Verarbeitung beginnt mit einer steigenden Flanke am Parameter "REQ".

Die Parameter "Busy" und "Done" geben den Auftragsstatus an.

Tritt während der Ausführung ein Fehler auf, wird dies von den Parametern "Error" und "Status" angezeigt.

Tabelle 9-246 Datentypen für die Parameter

Parameter und Typ		Datentyp	Beschreibung
REQ	Input	BOOL	Steuerparameter REQUEST Die Verarbeitung beginnt bei einer steigenden Flanke an REQ.
Name	Input	STRING	Pfad und Name der zu löschenden Datei
Done	Output	BOOL	Statusparameter <ul style="list-style-type: none"> • 1: Die Anweisung wurde erfolgreich ausgeführt.
Busy	Output	BOOL	Statusparameter <ul style="list-style-type: none"> • 0: Die Anweisung wird derzeit nicht ausgeführt. • 1: Die Anweisung wird derzeit ausgeführt.
Error	Output	BOOL	Statusparameter <ul style="list-style-type: none"> • 0: Kein Fehler • 1: Während der Ausführung der Anweisung ist ein Fehler aufgetreten. Ausführliche Informationen finden Sie im Parameter "Status".
Status	Output	WORD	Fehlercode

Tabelle 9-247 Bedingungscode

Fehlercode* (W#16#...)	Bedeutung
0000	Die Anweisung wurde erfolgreich fertiggestellt.
7000	Keine laufende Auftragsbearbeitung
7001	Start der Auftragsverarbeitung: Busy = 1, Done = 0
7002	Zwischenzeitlicher Aufruf (REQ irrelevant): Anweisung bereits aktiv; Busy hat den Wert 1.
8090	Datei gesperrt, z.B. wenn die Datei geöffnet ist
8091	Pfad ist nicht vorhanden oder ungültig.
8092	Datei ist in dem Pfad nicht vorhanden

Fehlercode* (W#16#...)	Bedeutung
80A2	Schreibfehler
80A3	Die Datei ist zu groß (≥ 2147483648 Bytes) und kann nicht mit "FileDelete" gelöscht werden.
80B4	Die Memory Card ist schreibgeschützt.
80C3	Die maximale Anzahl gleichzeitig aktiver Anweisungen "FileDelete" ist bereits erreicht.
* Die Fehlercodes werden im Programmeditor als ganzzahlige Werte oder als Hexadezimalwerte angezeigt.	

Siehe auch

Neue Funktionen (Seite 35)

10.1 Zählen (schnelle Zähler)

Die grundlegenden Zähleranweisungen, die unter "Zähler" (Seite 225) beschrieben werden, zählen nur solche Ereignisse, die langsamer auftreten als der Zyklus der S7-1200 CPU. Der schnelle Zähler (High-Speed Counter, HSC) bietet die Möglichkeit, Impulse zu zählen, die schneller auftreten als der PLC-Zyklus. Darüber hinaus können Sie den HSC so konfigurieren, dass er die Frequenz und Periode der aufgetretenen Impulse zählt, oder ihn so einrichten, dass die Bewegungssteuerung mit dem HSC ein Motorgebersignal lesen kann.

Um die HSC-Funktion zu nutzen, muss der HSC zunächst in der Gerätekonfiguration im Register "Eigenschaften" der CPU aktiviert und konfiguriert werden. Wie Sie den HSC konfigurieren, erfahren Sie unter "Konfigurieren eines schnellen Zählers" (Seite 560).

Nachdem Sie die Hardwarekonfiguration geladen haben, kann der HSC Impulse zählen oder die Frequenz messen, ohne dass dafür Anweisungen aufgerufen werden müssen. Wenn sich der HSC in der Betriebsart Zählen oder Periode befindet, wird der Zählwert automatisch in jedem Zyklus erfasst und im Prozessabbild (Speicherbereich E) aktualisiert. Wenn sich der HSC in der Betriebsart Frequenz befindet, ist der Wert im Prozessabbild die Frequenz in Hz.

Neben dem Zählen und Messen kann der HSC auch Prozessalarmereignisse generieren, abhängig vom Zustand der physischen Eingänge arbeiten und einen Ausgangsimpuls entsprechend einem angegebenen Zählerereignis (nur bei CPUs ab V4.2) erzeugen. Über die Technologieanweisung CTRL_HSC_EXT kann das Anwenderprogramm den HSC über das Programm steuern. Wenn CTRL_HSC_EXT ausgeführt wird, aktualisiert die Anweisung die HSC-Parameter und gibt die aktuellsten Werte aus. Die Anweisung CTRL_HSC_EXT können Sie in den Betriebsarten Zählen, Periode und Frequenz verwenden.

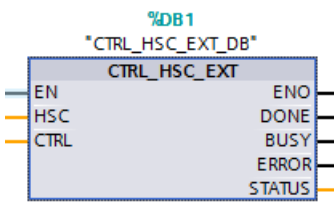
Hinweis

Die Anweisung CTRL_HSC_EXT ersetzt die bisherige Anweisung CTRL_HSC bei Projekten für CPUs ab V4.2. Mit der Anweisung CTRL_HSC_EXT stehen der gesamte Funktionsumfang der Anweisung CTRL_HSC sowie einige zusätzliche Funktionen zur Verfügung. Die bisherige Anweisung CTRL_HSC ist lediglich aus Kompatibilitätsgründen für ältere S7-1200 Projekte verfügbar und sollte in neuen Projekten nicht verwendet werden.

10.1.1 Anweisung CTRL_HSC_EXT (Schnellen Zähler steuern)

10.1.1.1 Übersicht über die Anweisung

Tabelle 10-1 Anweisung CTRL_HSC_EXT

KOP/FUP	SCL	Beschreibung
	<pre>"CTRL_HSC_1_DB" (hsc:=_hw_hsc_in_, done:=_done_out_, busy:=_busy_out_, error:=_error_out_, status:=_status_out_, ctrl:=_variant_in_);</pre>	<p>Jede Anweisung CTRL_HSC_EXT (Schnellen Zähler steuern (erweitert)) nutzt eine in einem benutzerdefinierten globalen DB abgelegte systemdefinierte Datenstruktur, um Zählerdaten zu speichern. Sie weisen die Datentypen HSC_Count, HSC_Period und HSC_Frequency als Eingangsparameter der Anweisung CTRL_HSC_EXT zu.</p>

- ¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- ² Im SCL-Beispiel ist "CTRL_HSC_1_DB" der Name des Instanz-DBs.

Tabelle 10-2 Datentypen für die Parameter

Parameter	Deklaration	Datentyp	Beschreibung
HSC	IN	HW_HSC	HSC-Kennung
CTRL	IN_OUT	Variant	SFB-Eingang und Rückgabedaten. Hinweis: Weitere Informationen finden Sie unter "Systemdatentypen für die Anweisung CTRL_HSC_EXT (SDT) (Seite 548)".
DONE	OUT	Bool	1 = Zeigt an, dass der SFB beendet ist. Immer 1, weil der SFB synchron ist.
BUSY	OUT	Bool	Immer 0, die Funktion ist nie besetzt.
ERROR	OUT	Bool	1 = Weist auf einen Fehler hin.
STATUS	OUT	Word	Ausführungsbedingung Hinweis: Siehe nachstehende Tabelle "Ausführungsbedingungen" mit weiteren Informationen.

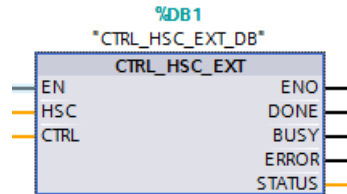
Tabelle 10-3 Ausführungsbedingungen

STATUS (W#16#)	Beschreibung
0	Kein Fehler
80A1	HSC-Kennung adressiert keinen HSC
80B1	Ungültiger Wert in NewDirection
80B4	Ungültiger Wert in NewPeriod
80B5	Ungültiger Wert in NewOpModeBehavior
80B6	Ungültiger Wert in NewLimitBehavior
80D0	SFB 124 nicht verfügbar

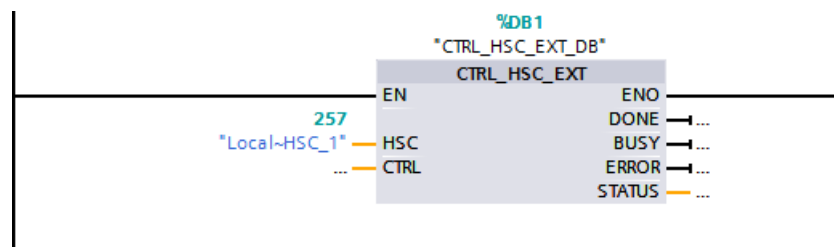
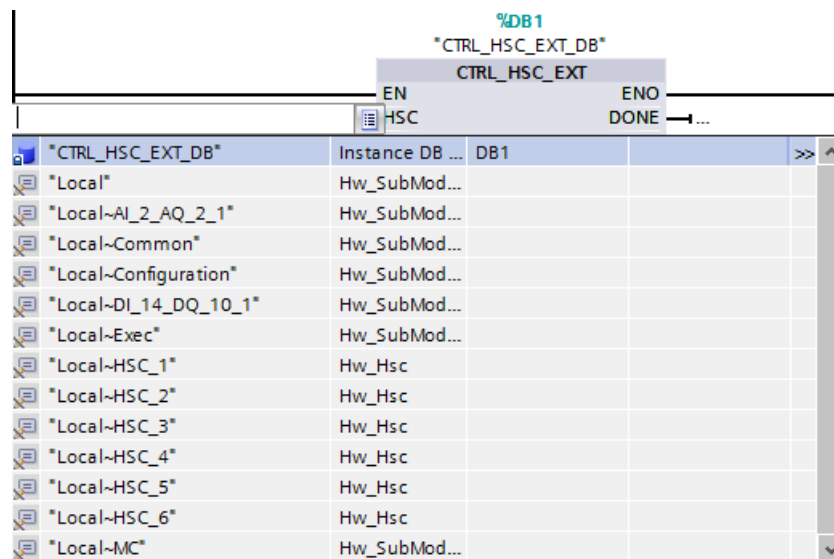
10.1.1.2 Beispiel

Zur Verwendung der Anweisung CTRL_HSC_EXT gehen Sie wie folgt vor:

1. Positionieren Sie die Anweisung CTRL_HSC_EXT in dem KOP-Netzwerk, das auch den folgenden Instanz-Datenbaustein erstellt: "CTRL_HSC_EXT_DB":



2. Fügen Sie die Hardwarekennung des HSC, die Sie in den Eigenschaften des HSC finden, an den Anschluss "HSC" der KOP-Anweisung an. Sie können auch eines der sechs "Hw_Hsc"-Objekte aus dem Dropdown-Menü für diesen Eingangsanschluss auswählen. Der Standardvariablenname für HSC1 ist "Local~HSC_1":



10.1 Zählen (schnelle Zähler)

3. Erstellen Sie einen globalen Datenbaustein mit dem Namen "Data_block_1" (Sie können auch einen vorhandenen globalen Datenbaustein verwenden):
 - Suchen Sie in "Data_block_1" eine leere Zeile und fügen Sie eine Variable mit dem Namen "MyHSC" hinzu.
 - Fügen Sie in der Spalte "Datentyp" einen der folgenden Systemdatentypen (SDT) hinzu. Wählen Sie den SDT aus, der der konfigurierten Zählweise des HSC entspricht. Weitere Informationen zu SDTs von HSCs finden Sie weiter unten in diesem Abschnitt. Die Klappliste enthält diese Typen nicht. Achten Sie deshalb besonders darauf, dass Sie den SDT-Namen genau wie gezeigt eingeben: HSC_Count, HSC_Period oder HSC_Frequency
 - Nachdem Sie den Datentyp eingegeben haben, können Sie die Variable "MyHSC" erweitern, um alle in der Datenstruktur enthaltenen Felder anzuzeigen. Hier finden Sie den Datentyp sämtlicher Felder und können die Standardstartwerte ändern:

Data_block_1			
	Name	Datentyp	Startwert
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. Weisen Sie die Variable "'Data_block_1'. MyHSC" dem Eingangsanschluss CTRL der Anweisung CTRL_HSC_EXT zu:
 - Wählen Sie "Data_Block_1" aus.

#Initial_Call	Bool		Initial call of this OB
#Remanence	Bool		=True, if remanent data
~-Port_1	Hw_Interface		
~-Port_2	Hw_Interface		
"Automatic update"	Pip		
"CTRL_HSC_EXT_DB"	Instance DB of HSC [SF...	DB1	
"Data_block_1"	Global DB	DB2	
"Local"	Hw_SubModule		
"Local~AI_2_AQ_2_1"	Hw_SubModule		
"Local~Common"	Hw_SubModule		
"Local~Configuration"	Hw_SubModule		
"Local~Device"	Hw_Device		
"Local~DI_14_DQ_10_1"	Hw_SubModule		
"Local~Exec"	Hw_SubModule		
"Local~HSC_1"	Hw_Hsc		
"Local~HSC_2"	Hw_Hsc		
"Local~HSC_3"	Hw_Hsc		
"Local~HSC_4"	Hw_Hsc		
"Local~HSC_5"	Hw_Hsc		
"Local~HSC_6"	Hw_Hsc		
"Local~MC"	Hw_SubModule		

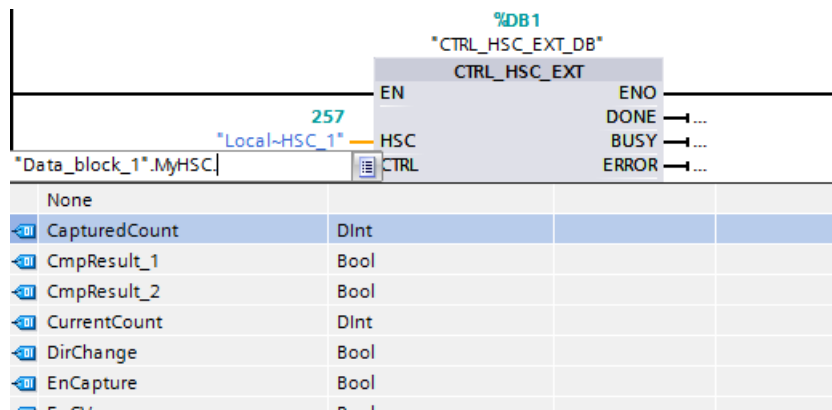
- Wählen Sie "MyHSC" aus.

None			
MyHSC	HSC_Count		

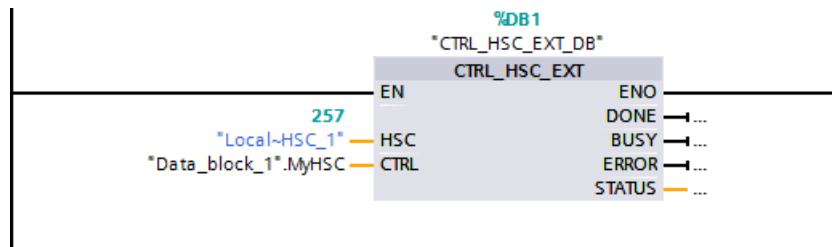
- Löschen Sie den Punkt ("."), der auf "'Data_Block_1'.MyHSC" folgt. Klicken Sie dann entweder außerhalb des Felds oder drücken Sie einmal die ESC-Taste und drücken Sie dann die Eingabetaste.

Hinweis

Nach dem Löschen des Punkts ("."), der auf "'Data_Block_1'.MyHSC" folgt, drücken Sie nicht nur die Eingabetaste. Dadurch wird der Punkt (".") wieder im Feld angeordnet.



– Die vollständige Eingabe für CTRL wird nachfolgend gezeigt.



Nachdem Sie den HSC im PLC konfiguriert haben, können Sie die Anweisung CTRL_HSC_EXT ausführen. Bei einem Fehler wird ENO auf 0 gesetzt und der Ausgang STATUS gibt den Bedingungscode an.

Siehe auch

Systemdatentypen (SDT) der Anweisung CTRL_HSC_EXT (Seite 548)

10.1.1.3 Systemdatentypen (SDT) der Anweisung CTRL_HSC_EXT

Die folgenden Systemdatentypen (SDTs) werden nur für den Anschluss CTRL der Anweisung CTRL_HSC_EXT verwendet. Um sie zu verwenden, erstellen Sie einen Anwenderdatenbaustein und fügen ein Objekt mit dem Datentyp des SDT hinzu, der der konfigurierten Betriebsart (Zählart) des HSC entspricht. In STEP 7 werden diese Datentypen nicht im Dropdown-Menü angezeigt. Geben Sie den Namen des SDT also genau wie dargestellt ein.

Die Eingänge des SDT des HSC sind durch das Präfix "En" oder "New" gekennzeichnet. Eingänge mit dem Präfix "En" aktivieren entweder eine HSC-Funktion oder aktualisieren den entsprechenden Parameter. Das Präfix "New" identifiziert den Aktualisierungswert. Damit der neue Wert wirksam werden kann, muss das entsprechend "En"-Bit wahr sein und der Wert "New" muss gültig sein. Wenn die Anweisung CTRL_HSC_EXT ausgeführt wird, wendet das Programm Eingangsänderungen an und aktualisiert die Ausgänge mit der entsprechenden SDT-Referenz.

SDT: HSC_Count

Der Datentyp "HSC_Count" entspricht einem HSC, der für die Betriebsart "Zählen" konfiguriert ist. Die Betriebsart Zählen bietet die folgenden Funktionen:

- Zugriff auf den aktuellen Impulszählwert
- Halten des aktuellen Impulszählwerts bei einem Eingangereignis
- Zurücksetzen des aktuellen Impulszählwerts auf den Startwert bei einem Eingangereignis
- Zugriff auf Statusbits, die das Auftreten bestimmter HSC-Ereignisse melden
- Deaktivieren des HSC über einen Software- oder Hardwareeingang
- Wechseln der Zählrichtung über einen Software- oder Hardwareeingang
- Ändern des aktuellen Impulszählwerts
- Ändern des Startwerts (wird verwendet, wenn die CPU in RUN wechselt oder die Synchronisierungsfunktion ausgelöst wird)
- Ändern zweier unabhängiger Referenzwerte (bzw. voreingestellter Werte) für Vergleiche
- Ändern der oberen und unteren Zählgrenzen
- Ändern der Funktionsweise des HSC für den Fall, dass der Impulszählwert diese Grenzwerte erreicht
- Generieren eines Prozessalarmereignisses, wenn der aktuelle Impulszählwert einen Referenzwert (voreingestellten Wert) erreicht
- Generieren eines Prozessalarmereignisses, wenn der Synchronisierungseingang (Rücksetzeingang) aktiviert wird
- Generieren eines Prozessalarmereignisses, wenn sich die Zählrichtung basierend auf einem externen Eingang ändert
- Generieren eines einzelnen Ausgangsimpulses bei einem angegebenen Zählereignis

Wenn bei ausgeführter Anweisung CTRL_HSC_EXT ein Ereignis auftritt, setzt die Anweisung ein Statusbit. Bei der nachfolgenden Ausführung der Anweisung löscht die Anweisung das Statusbit, sofern das Ereignis nicht erneut vor der Ausführung der Anweisung auftritt.

Tabelle 10-4 Struktur HSC_Count

Strukturelement	Deklaration	Datentyp	Beschreibung
CurrentCount	OUT	Dint	Gibt den aktuellen Zählwert des HSC aus
CapturedCount	OUT	Dint	Gibt den am angegebenen Eingangereignis erfassten Zählwert aus
SyncActive	OUT	Bool	Statusbit: Synchronisierungseingang wurde aktiviert
DirChange	OUT	Bool	Statusbit: Zählrichtung wurde gewechselt
CmpResult1	OUT	Bool	Statusbit: Ereignis CurrentCount gleich Reference1 aufgetreten
CmpResult2	OUT	Bool	Statusbit: Ereignis CurrentCount gleich Reference2 aufgetreten
OverflowNeg	OUT	Bool	Statusbit: CurrentCount hat LowerLimit erreicht
OverflowPos	OUT	Bool	Statusbit: CurrentCount hat UpperLimit erreicht
EnHSC	IN	Bool	Ermöglicht dem HSC, wenn wahr, Impulse zu zählen; wenn falsch, wird das Zählen deaktiviert

Strukturelement	Deklaration	Datentyp	Beschreibung
EnCapture	IN	Bool	Aktiviert den Eingang Capture, wenn wahr; wenn falsch, hat der Eingang Capture keine Wirkung
EnSync	IN	Bool	Aktiviert den Eingang Sync, wenn wahr; wenn falsch, hat der Eingang Sync keine Wirkung
EnDir	IN	Bool	Ermöglicht dem Wert von NewDirection, wirksam zu werden
EnCV	IN	Bool	Ermöglicht dem Wert von NewCurrentCount, wirksam zu werden
EnSV	IN	Bool	Ermöglicht dem Wert von NewStartValue, wirksam zu werden
EnReference1	IN	Bool	Ermöglicht dem Wert von NewReference1, wirksam zu werden
EnReference2	IN	Bool	Ermöglicht dem Wert von NewReference2, wirksam zu werden
EnUpperLmt	IN	Bool	Ermöglicht dem Wert von NewUpperLimit, wirksam zu werden
EnLowerLmt	IN	Bool	Ermöglicht dem Wert von New_Lower_Limit, wirksam zu werden
EnOpMode	IN	Bool	Ermöglicht dem Wert von NewOpModeBehavior, wirksam zu werden
EnLmtBehavior	IN	Bool	Ermöglicht dem Wert von NewLimitBehavior, wirksam zu werden
EnSyncBehavior	IN	Bool	Dieser Wert wird nicht verwendet.
NewDirection	IN	Int	Zählrichtung: 1 = Vorwärtszählen; -1 = Rückwärtszählen; alle anderen Werte sind reserviert
NewOpModeBehavior	IN	Int	Funktionsweise des HSC bei Überlauf: 1 = HSC hört auf zu zählen (der HSC muss deaktiviert und erneut aktiviert werden, um weiterzuzählen); 2 = HSC setzt den Betrieb fort; alle anderen Werte sind reserviert
NewLimitBehavior	IN	Int	Ergebnis des Werts CurrentCount bei Überlauf: 1 = CurrentCount auf entgegengesetzten Grenzwert setzen; 2 = CurrentCount auf StartValue setzen; alle anderen Werte sind reserviert
NewSyncBehavior	IN	Int	Dieser Wert wird nicht verwendet.
NewCurrentCount	IN	Dint	Wert von CurrentCount
NewStartValue	IN	Dint	StartValue: Anfangswert des HSC
NewReference1	IN	Dint	Wert von Reference1
NewReference2	IN	Dint	Wert von Reference2
NewUpperLimit	IN	Dint	Oberer Zählgrenzwert
New_Lower_Limit	IN	Dint	Unterer Zählgrenzwert

SDT: HSC_Period

Der Datentyp "HSC_Period" entspricht einem HSC, der für die Betriebsart "Periode" konfiguriert ist. Die Anweisung CTRL_HSC_EXT bietet Programmzugriff auf die Anzahl von Eingangsimpulsen über ein angegebenes Messintervall. Diese Anweisung ermöglicht die Berechnung des Zeitraums zwischen den Eingangsimpulsen mit einer Auflösung im Nanosekundenbereich.

Tabelle 10-5 HSC_Period -Struktur

Strukturelement	Deklaration	Datentyp	Beschreibung
ElapsedTime	OUT	UDInt	Siehe Beschreibung unten.
EdgeCount	OUT	UDInt	Siehe Beschreibung unten.
EnHSC	IN	Bool	Aktiviert, wenn wahr, den HSC für die Periodenmessung; deaktiviert die Periodenmessung, wenn falsch.

Strukturelement	Deklaration	Datentyp	Beschreibung
EnPeriod	IN	Bool	Ermöglicht dem Wert von NewPeriod, wirksam zu werden.
NewPeriod	IN	Int	Gibt die Messintervallzeit in Millisekunden an. Zulässig sind nur die Werte 10, 100 oder 1000 ms.

ElapsedTime gibt die Zeit in Nanosekunden zwischen den letzten Zählereignissen von aufeinander folgenden Messintervallen an. Traten während eines Messintervalls keine Zählereignisse auf, gibt ElapsedTime die kumulierte Zeit seit dem letzten Zählereignis aus. ElapsedTime hat einen Bereich von 0 bis 4.294.967.280 Nanosekunden (0x0000 0000 bis 0xFFFF FFF0). Der Rückgabewert 4.294.967.295 (0xFFFF FFFF) meldet, dass ein Periodenüberlauf aufgetreten ist. Ein Überlauf deutet darauf hin, dass die Zeit zwischen Impulsflanken größer als 4,295 Sekunden ist und die Periode mit dieser Anweisung nicht berechnet werden kann. Die Werte von 0xFFFF FFF1 bis 0xFFFF FFFE sind reserviert.

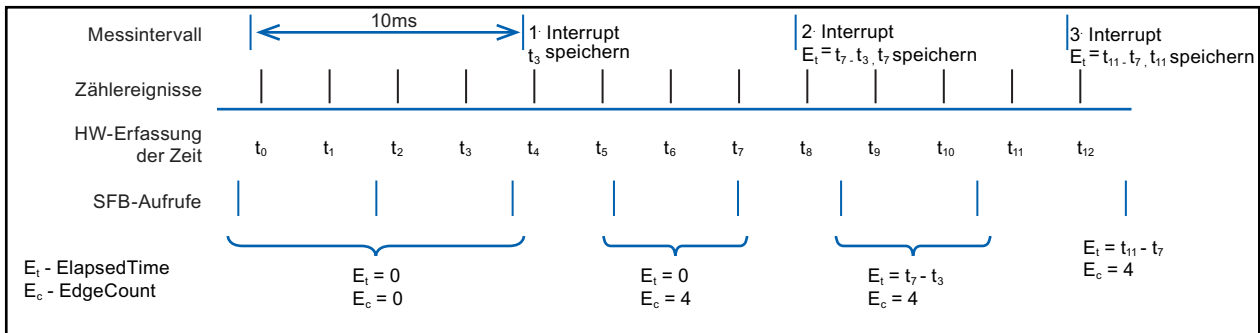
EdgeCount gibt die Anzahl der während des Messintervalls empfangenen Zählereignisse aus. Die Periode kann nur berechnet werden, wenn der Wert von EdgeCount größer als Null ist. Wenn ElapsedTime entweder 0 (keine Eingangsimpulse empfangen) oder 0xFFFF FFFF (Periodenüberlauf) ist, ist EdgeCount nicht gültig.

Wenn EdgeCount gültig ist, berechnen Sie die Periode in Nanosekunden mit Hilfe der folgenden Formel: $\text{Periode} = \text{ElapsedTime} / \text{EdgeCount}$

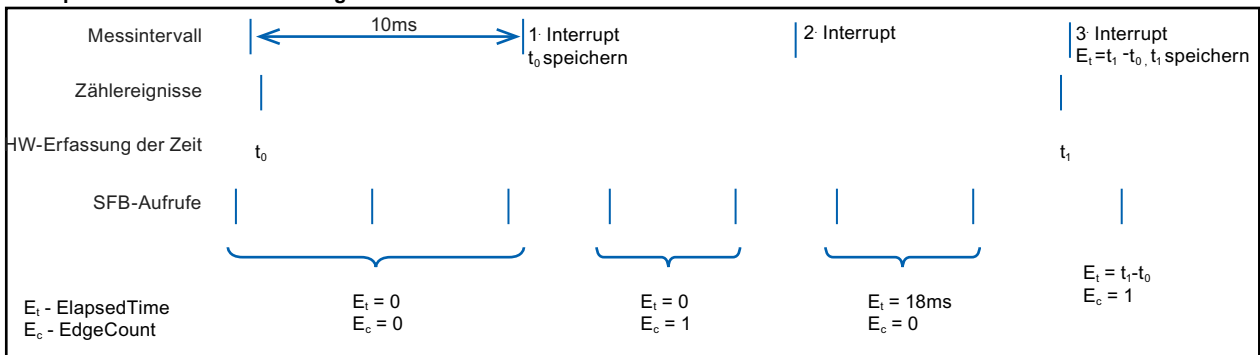
Der berechnete Periodenwert ist ein Mittelwert aus den Zeitperioden sämtlicher Impulse, die während des Messintervalls auftreten. Wenn die Periode eines kommenden Impulses größer als das Messintervall ist (10, 100 oder 1000 ms), benötigt die Periodenberechnung mehrere Messintervalle.

Die folgenden Beispiele zeigen, wie die Anweisung Periodenmessungen durchführt:

Beispiel 1: Mehrere Zählereignisse in einem Messintervall



Beispiel 2: Null und ein Zählereignis in mehreren Messintervallen



Regeln:

1. Wenn $E_t = 0$, ist der Zeitraum ungültig
2. Andernfalls, Zeitraum = E_t / E_c

SDT: HSC_Frequency

Der Datentyp "HSC_Frequency" entspricht einem HSC, der für die Betriebsart "Frequenz" konfiguriert ist. Die Anweisung CTRL_HSC_EXT bietet Programmmzugriff auf die Frequenz von Eingangsimpulsen, die über einen angegebenen Zeitraum gemessen werden.

Die Anweisung CTRL_HSC_EXT bietet in der Betriebsart Frequenz die folgenden Möglichkeiten:

Tabelle 10-6 Struktur von HSC_Frequency

Strukturelement	Deklaration	Datentyp	Beschreibung
Frequenz	OUT	DInt	Gibt eine Frequenz in Hz aus, gemessen über die Zeit des Messintervalls. Wenn der HSC rückwärts zählt, gibt die Anweisung eine negative Frequenz aus.
EnHSC	IN	Bool	Aktiviert, wenn wahr, den HSC für die Frequenzmessung; deaktiviert die Frequenzmessung, wenn falsch.
EnPeriod	IN	Bool	Ermöglicht dem Wert von NewPeriod, wirksam zu werden.
NewPeriod	IN	Int	Gibt die Messintervallzeit in Millisekunden an. Zulässig sind nur die Werte 10, 100 oder 1000 ms.

Die Anweisung CTRL_HSC_EXT misst die Frequenz mit der gleichen Messtechnik wie die Betriebsart Periode, um ElapsedTime und EdgeCount zu ermitteln. Die Anweisung berechnet die

Frequenz als ganzzahligen Wert mit Vorzeichen in Hz mit Hilfe der Formel: $\text{Frequenz} = \text{EdgeCount} / \text{ElapsedTime}$

Wenn Sie für die Frequenz einen Gleitpunktwert benötigen, können Sie die oben genannte Formel verwenden, wenn sich der HSC in der Betriebsart Periode befindet. Beachten Sie, dass der Wert von ElapsedTime in der Betriebsart Periode in Nanosekunden ausgegeben wird und gegebenenfalls skaliert werden muss.

10.1.2 Betrieb des schnellen Zählers

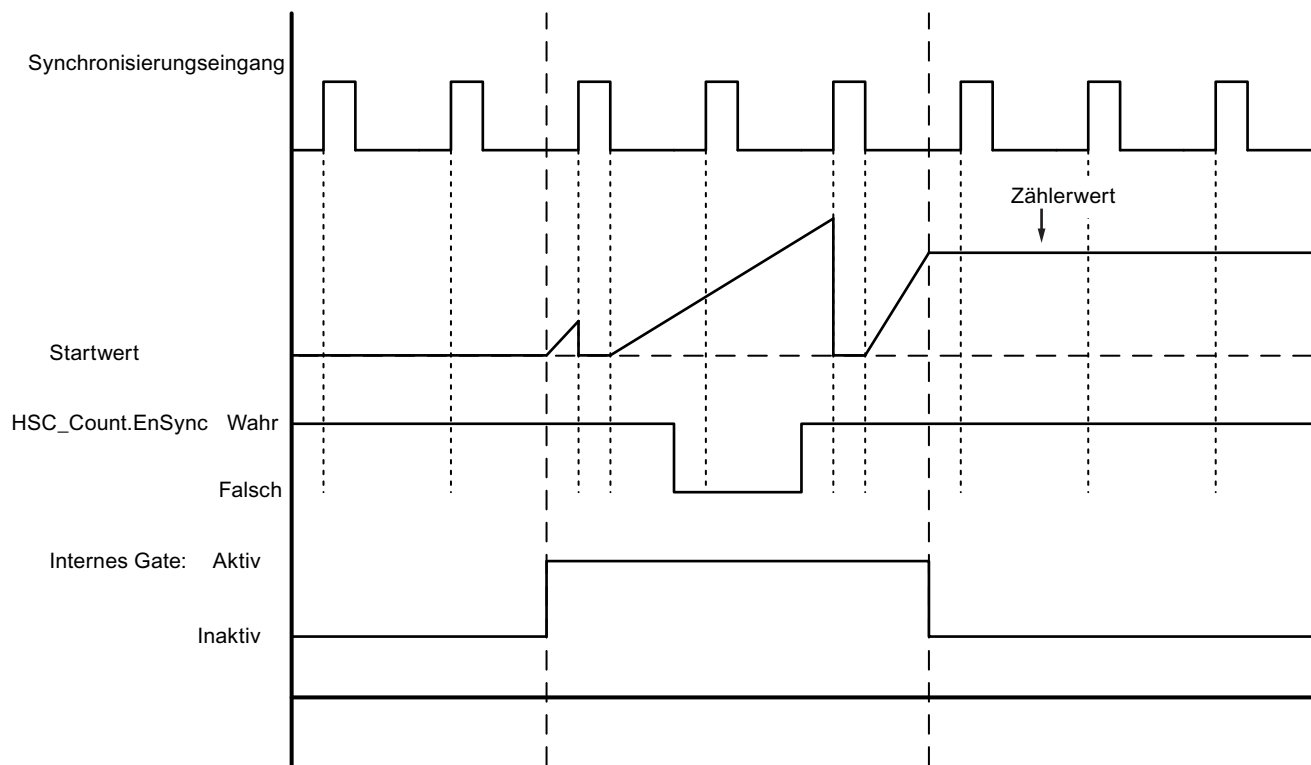
10.1.2.1 Synchronisierungsfunktion

Mit der Synchronisierungsfunktion setzen Sie den Zähler mit einem externen Eingangssignal auf den Startwert. Sie können den Startwert durch Ausführen der Anweisung CTRL_HSC_EXT ändern. Dadurch lässt sich der aktuelle Zählwert bei Auftreten eines externen Eingangssignals mit einem gewünschten Wert synchronisieren.

Die Synchronisierung erfolgt immer bei Auftreten des Eingangssignals und ist unabhängig vom Status des internen Gates wirksam. Sie müssen das Bit "HSC_Count.EnSync" auf wahr setzen, um die Synchronisierungsfunktion zu aktivieren.

Die Anweisung CTRL_HSC_EXT setzt das Statusbit HSC_Count.SyncActive nach Abschluss der Synchronisierung auf wahr. Die Anweisung CTRL_HSC_EXT setzt das Statusbit HSC_Count.SyncActive auf falsch, wenn seit der letzten Ausführung der Anweisung keine Synchronisierung aufgetreten ist.

Die folgende Abbildung zeigt ein Beispiel für die Synchronisierung, wenn das Eingangssignal für den Pegel "Aktiv High" konfiguriert ist:



Hinweis

Die konfigurierten Eingangsfiler verzögern das Steuersignal des Digitaleingangs.

Diese Eingangsfunktion ist nur verfügbar, wenn der HSC für die Betriebsart Zählen konfiguriert ist.

Informationen zur Konfiguration der Synchronisierungsfunktion finden Sie unter Eingangsfunktionen (Seite 566).

10.1.2.2 Gate-Funktion

In vielen Anwendungen müssen Zählprozesse in Abstimmung mit anderen Ereignissen gestartet oder gestoppt werden. In solchen Fällen wird das Zählen über die Funktion des internen Gates gestartet und gestoppt. Jeder HSC-Kanal hat zwei Gates: ein Software-Gate und ein Hardware-Gate. Der Zustand dieser Gates bestimmt den Zustand des internen Gates (siehe nachfolgende Tabelle).

Das interne Gate ist geöffnet, wenn das Software-Gate geöffnet ist und das Hardware-Gate geöffnet ist oder nicht konfiguriert wurde. Wenn das interne Gate geöffnet ist, wird das Zählen

gestartet. Wenn das interne Gate geschlossen ist, werden alle anderen Zählimpulse ignoriert und das Zählen gestoppt.

Tabelle 10-7 Zustände der Gate-Funktion

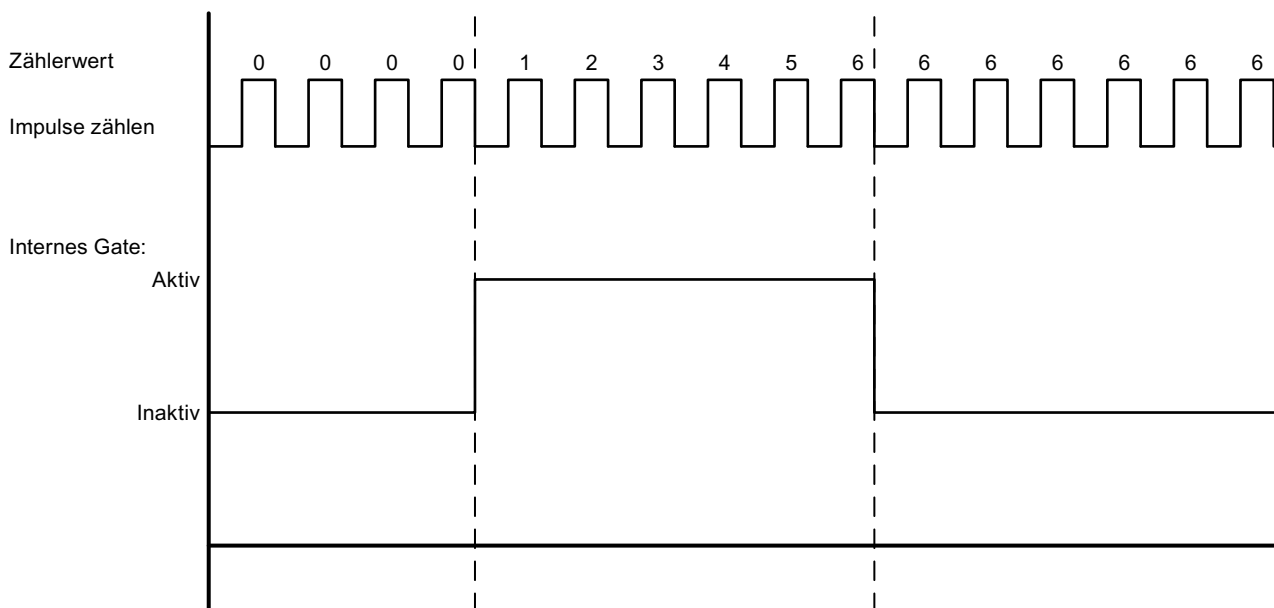
Hardware-Gate	Software-Gate	Internes Gate
Geöffnet/nicht konfiguriert	Geöffnet	Geöffnet
Geöffnet/nicht konfiguriert	Geschlossen	Geschlossen
Geschlossen	Geöffnet	Geschlossen
Geschlossen	Geschlossen	Geschlossen

Der Begriff "geöffnet" ist als aktiver Zustand des Gates definiert. Ebenso ist der Begriff "geschlossen" als inaktiver Zustand des Gates definiert.

Sie steuern das Software-Gate mit dem Freigabebit "HSC_Count.EnHSC" im SDT der Anweisung CTRL_HSC_EXT. Zum Öffnen des Software-Gates setzen Sie das Bit "HSC_Count.EnHSC" auf wahr; zum Schließen des Software-Gates setzen Sie das Bit "HSC_Count.EnHSC" auf falsch. Führen Sie die Anweisung CTRL_HSC_EXT aus, um den Zustand des Software-Gates auszuführen.

Das Hardware-Gate ist optional, und Sie können es im Bereich für die Eigenschaften des HSC aktivieren oder deaktivieren. Um einen Zählvorgang nur mit dem Hardware-Gate zu steuern, muss das Software-Gate geöffnet bleiben. Wenn Sie kein Hardware-Gate konfigurieren, wird das Hardware-Gate als ständig geöffnet betrachtet und der Gate-Zustand ist der gleiche wie der des Software-Gates.

Die folgende Abbildung zeigt ein Beispiel für das Öffnen und Schließen des Hardware-Gates mit einem Digitaleingang. Der Digitaleingang ist für den Pegel "Aktiv High" konfiguriert:



Hinweis

Die konfigurierten Eingangsfilter verzögern das Steuersignal des Digitaleingangs.

Die Funktion des Hardware-Gates ist nur verfügbar, wenn der HSC für die Betriebsart Zählen konfiguriert ist. In den Betriebsarten Periode und Frequenz ist der Zustand des internen Gates der gleiche wie der des Software-Gates.

In der Betriebsart Periode wird das Software-Gate von "HSC_Period.EnHSC" gesteuert.

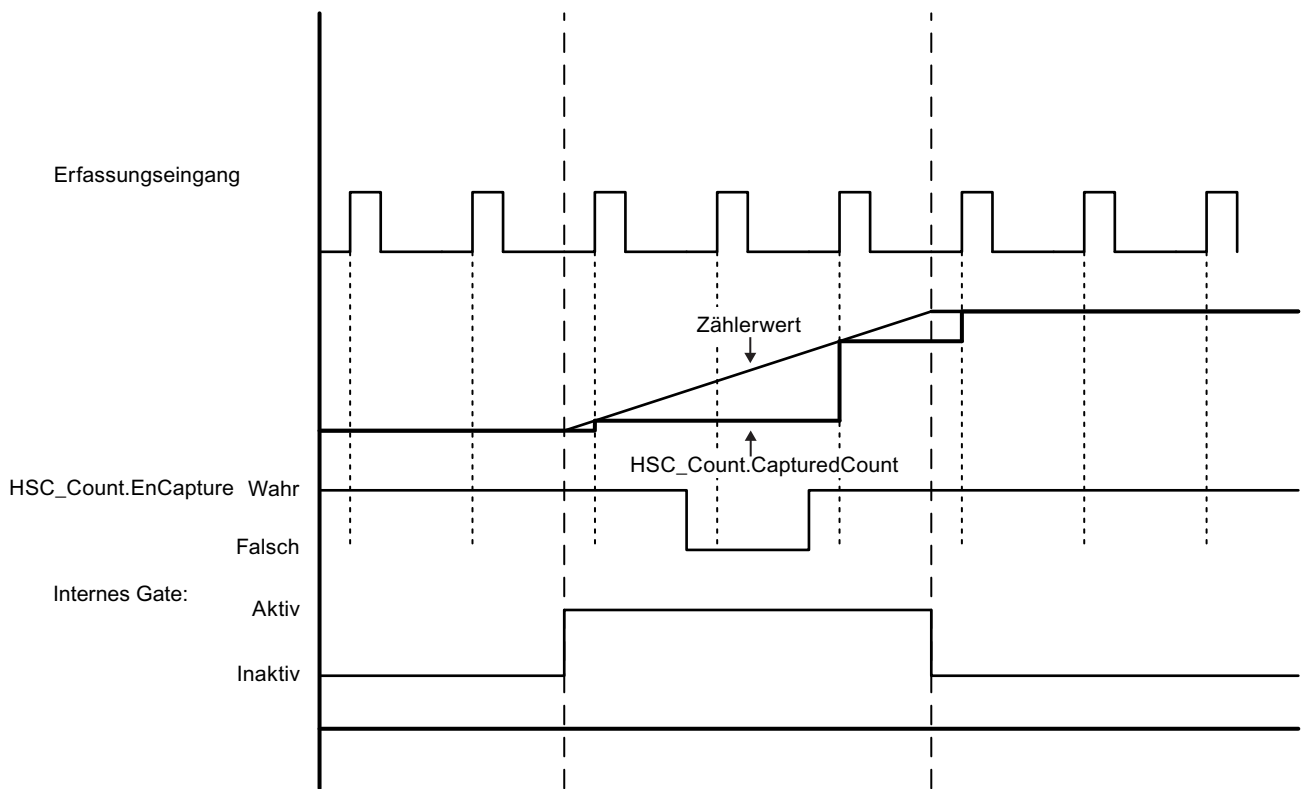
In der Betriebsart Frequenz wird das Software-Gate von "HSC_Frequency.EnHSC" gesteuert.

Informationen zur Konfiguration der Gate-Funktion finden Sie unter Eingangsfunktionen (Seite 566).

10.1.2.3 Erfassungsfunktion

Mit der Erfassungsfunktion speichern Sie den aktuellen Zählerwert mit einem externen Referenzsignal. Wenn über das Bit "HSC_Count.EnCapture" konfiguriert und aktiviert, bewirkt die Erfassungsfunktion, dass bei Auftreten einer externen Eingangsflanke der aktuelle Zählwert erfasst wird. Die Erfassungsfunktion ist unabhängig vom Zustand des internen Gates wirksam. Wenn das Gate geschlossen ist, speichert das Programm den unveränderten Zählerwert. Nach der Ausführung der Anweisung CTRL_HSC_EXT speichert das Programm den in "HSC_Count.CapturedCount" erfassten Wert.

Die nachfolgende Abbildung zeigt ein Beispiel für die Erfassungsfunktion, die für die Erfassung bei einer steigenden Flanke konfiguriert ist. Der Erfassungseingang stößt keine Erfassung des aktuellen Zählwerts an, wenn das Bit "HSC_Count.EnCapture" über die Anweisung CTRL_HSC_EXT auf falsch gesetzt wird.



Hinweis

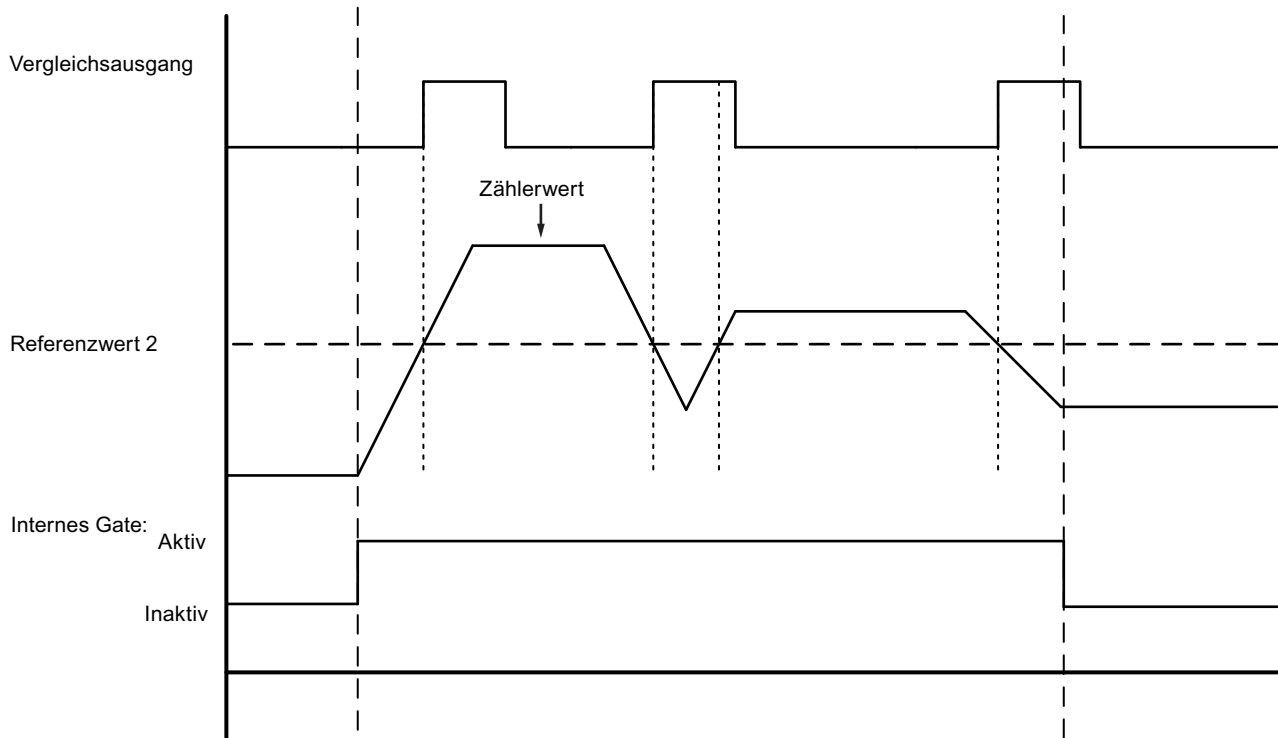
Die konfigurierten Eingangsfiler verzögern das Steuersignal des Digitaleingangs.

Diese Eingangsfunktion kann nur verwendet werden, wenn der HSC für die Betriebsart Zählen konfiguriert ist.

Informationen zur Konfiguration der Erfassungsfunktion finden Sie unter Eingangsfunktionen (Seite 566).

10.1.2.4 Vergleichsfunktion

Wenn aktiviert, generiert die Vergleichsausgangsfunktion bei jedem Auftreten des konfigurierten Ereignisses einen einzelnen konfigurierbaren Impuls. Zu den Ereignissen gehören: Zählerwert entspricht einem der Referenzwerte oder Überlauf des Zählers. Wenn ein Impuls in Verarbeitung ist und das Ereignis erneut auftritt, wird für das Ereignis kein Impuls erzeugt.



Hinweis

Diese Ausgangsfunktion kann nur verwendet werden, wenn der HSC für die Betriebsart Zählen konfiguriert ist.

Informationen zur Konfiguration der Vergleichsfunktion finden Sie unter Ausgangsfunktionen (Seite 567).

10.1.2.5 Anwendungen

In einer typischen Anwendung wird über den HSC die Rückmeldung eines Winkelschrittgebers überwacht. Der Geber liefert eine angegebene Anzahl von Zählerwerten pro Umdrehung, die Sie als Taktgeneratoreingang für den HSC verwenden können. Außerdem gibt es einen Rücksetzimpuls, der einmal pro Umdrehung auftritt und den Sie als Synchronisierungseingang für den HSC verwenden können.

Zum Starten sollte das Anwenderprogramm den ersten Referenzwert in den HSC laden und die Ausgänge auf ihren Anfangszustand setzen. Die Ausgänge bleiben für den Zeitraum, in dem der aktuelle Zählerwert kleiner als der Referenzwert ist, in diesem Zustand. Der HSC löst einen Alarm aus, wenn der aktuelle Zählerwert gleich dem Referenzwert ist, wenn das

Synchronisierungseignis (Rücksetzen) auftritt oder wenn es zu einem Richtungswechsel kommt.

Wenn jeder Zählerwert dem Referenzwert entspricht, tritt ein Alarmereignis auf. Im Alarm-OB sollte das Anwenderprogramm den nächsten Referenzwert in den HSC laden und die Ausgänge auf ihren nächsten Zustand setzen.

Wenn der Synchronisierungseingang ausgelöst wird, wird der aktuelle Zählwert auf den Startwert gesetzt und ein Alarmereignis tritt auf. In diesem Alarm-OB sollte das Anwenderprogramm den ersten Referenzwert in den HSC laden und die Ausgänge auf ihren Anfangszustand setzen. Zu diesem Zeitpunkt ist der HSC zu seinem Anfangszustand zurückgekehrt und der Zyklus wird wiederholt, wobei der HSC weiterzählt.

Da die Alarmergebnisse sehr viel langsamer auftreten als der HSC zählt, können Sie eine präzise Steuerung der schnellen Anweisungen mit relativ geringem Einfluss auf den Zyklus der CPU implementieren. Da Sie Alarmergebnisse bestimmten Interruptprogrammen zuordnen können, kann jede neue Voreinstellung in einem getrennten Interruptprogramm geladen werden, damit der Zustand auf diese Weise einfach gesteuert werden kann. Alternativ können Sie alle Alarmereignisse in einem einzelnen Interruptprogramm verarbeiten.

Die Gate-Funktion, ausgelöst entweder vom Anwenderprogramm oder von einem externen Eingangssignal, kann das Zählen der Geberimpulse deaktivieren. Sie können jede Bewegung der Welle durch Deaktivierung des Gates ignorieren. Das bedeutet, dass der Geber zwar weiterhin Impulse an den HSC sendet, doch der Zählwert auf dem letzten Wert gehalten wird, bevor das Gate in den inaktiven Zustand wechselt. Wenn das Gate wieder in den aktiven Zustand wechselt, wird das Zählen ab dem letzten Wert vor der Deaktivierung des Zählers fortgesetzt.

Wenn aktiviert, bewirkt die Erfassungsfunktion, dass bei Auftreten eines externen Eingangs der aktuelle Zählwert erfasst wird. Ein Prozess (z.B. eine Kalibrierungsroutine) kann mit Hilfe dieser Funktion ermitteln, wie viele Impulse zwischen den Ereignissen auftreten.

Wenn aktiviert, generiert die Vergleichsausgangsfunktion jedes Mal, wenn der aktuelle Zählwert einen der Referenzwerte oder Überläufe (Überschreiten der Zählgrenzen) erreicht, einen einzelnen konfigurierbaren Impuls. Diesen Impuls können Sie als Signal verwenden, um bei jedem Auftreten eines bestimmten HSC-Ereignisses einen anderen Prozess zu starten.

Die Zählrichtung wird entweder vom Anwenderprogramm oder einem externen Eingangssignal gesteuert.

Um die Drehzahl der Welle abzurufen, können Sie den HSC für die Betriebsart Frequenz konfigurieren. Diese Funktion liefert einen ganzzahligen Wert mit Vorzeichen in der Einheit Hz. Da das Rücksetzsignal einmal pro Umdrehung auftritt, bietet das Messen der Frequenz des Rücksetzsignals einen schnellen Hinweis auf die Drehzahl der Welle in Umdrehungen pro Sekunde.

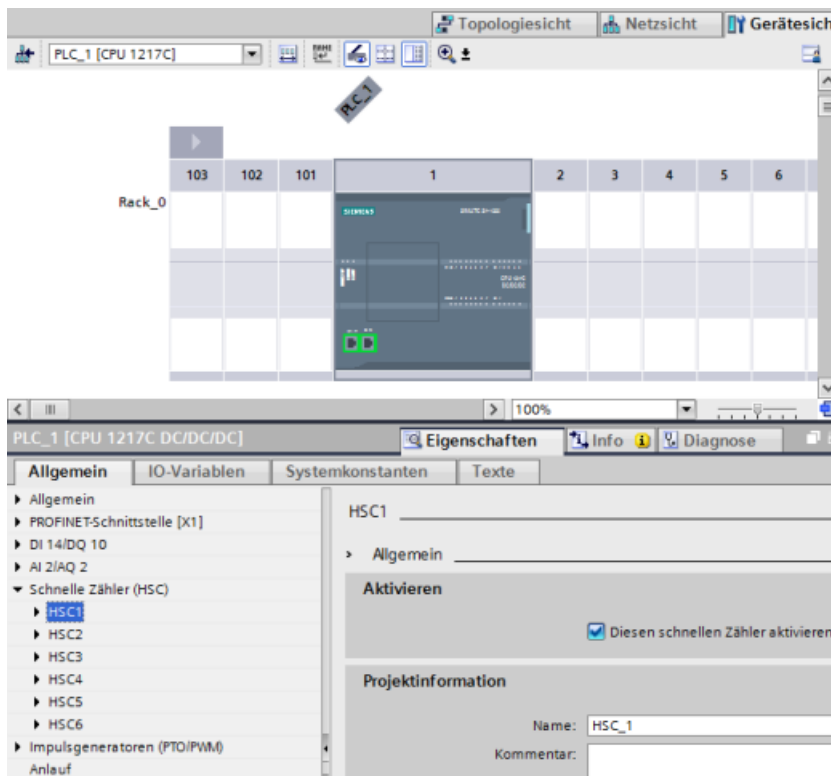
Wenn Sie die Frequenz als Gleitpunktwert benötigen, konfigurieren Sie den HSC für die Betriebsart Periode. Mit den in der Betriebsart Periode ausgegebenen Werten von ElapsedTime und EdgeCount können Sie die Frequenz berechnen.

10.1.3 Konfigurieren eines schnellen Zählers

Zum Einrichten des schnellen Zählers (HSC) führen Sie Folgendes aus:

- Wählen Sie in der Projektnavigation die Gerätekonfiguration aus.
- Wählen Sie die CPU aus, die Sie konfigurieren möchten.
- Klicken Sie im Inspektorfenster auf das Register "Eigenschaften" (siehe Abbildung unten).
- Wählen Sie den HSC, den Sie aktivieren möchten, in der Liste im Register "Allgemein" aus (siehe Abbildung unten).

Sie können bis zu sechs schnelle Zähler konfigurieren (HSC1 bis HSC6). Aktivieren Sie einen HSC durch Auswahl der Option "Diesen schnellen Zähler aktivieren". Nach der Aktivierung weist STEP 7 diesem HSC einen eindeutigen Standardnamen zu. Diesen Namen können Sie im Bearbeitungsfeld "Name:" ändern, der Name muss jedoch eindeutig sein. Namen von aktivierten HSCs werden zu Variablen mit dem Datentyp "HW_Hsc" in der Variablen-tabelle "Systemkonstanten" und können als Parameter "HSC" der Anweisungen CTRL_HSC_EXT genutzt werden. Weitere Informationen finden Sie unter Konfigurieren des CPU-Betriebs (Seite 153).



Nach der Aktivierung des HSC legt STEP 7 den Einphasenzähler als Standardkonfiguration fest. Sobald Sie den digitalen Eingangsfiter des HSC-Taktgeneratoreingangs festgelegt haben, kann das Programm in den PLC geladen werden, und die CPU ist bereit, den Zähler einzusetzen. Wie Sie die Konfiguration des HSCs ändern, erfahren Sie im nächsten Abschnitt "Zählarten".

Die folgende Tabelle bietet eine Übersicht über die Eingänge und Ausgänge, die für jede Konfiguration zur Verfügung stehen:

Tabelle 10-8 Zählarten für den HSC

Typ	Eingang 1	Eingang 2	Eingang 3	Eingang 4	Eingang 5	Ausgang 1	Funktion
Einphasenzähler mit interner Richtungssteuerung	Takt	-	-	-	-	-	Zählen, Frequenz oder Periode
			Synchronisieren	Gate	Erfassen	Vergleichen	Zählen
Einphasenzähler mit externer Richtungssteuerung	Takt	Richtung	-	-	-	-	Zählen, Frequenz oder Periode
			Synchronisieren	Gate	Erfassen	Vergleichen	Zählen
Zweiphasenzähler	Vorwärtszählen	Rückwärtszählen	-	-	-	-	Zählen, Frequenz oder Periode
			Synchronisieren	Gate	Erfassen	Vergleichen	Zählen
A/B-Zähler	Phase A	Phase B	-	-	-	-	Zählen, Frequenz oder Periode
			Synchronisieren ¹	Gate	Erfassen	Vergleichen	Zählen
Vierfacher A/B-Zähler	Phase A	Phase B	-	-	-	-	Zählen, Frequenz oder Periode
			Synchronisieren ¹	Gate	Erfassen	Vergleichen	Zählen

¹ Bei einem Winkelschrittgeber: Phase Z, Referenzpunkt

10.1.3.1 Zählarten

Es gibt vier Zählarten bzw. Betriebsarten. Wenn Sie die Betriebsart ändern, ändern sich auch die verfügbaren Konfigurationsoptionen für den HSC:

- **Zählen:** Zählt die Anzahl von Impulsen und inkrementiert oder dekrementiert den Zählwert abhängig vom Zustand der Richtungssteuerung. Externe E/A können den Zählwert zurücksetzen, das Zählen deaktivieren, die Erfassung des aktuellen Zählwerts auslösen oder bei einem angegebenen Ereignis einen einzelnen Impuls erzeugen. Die Ausgangswerte sind der aktuelle Zählwert und der Zählwert im Moment eines Erfassungsereignisses.
- **Periode:** Zählt die Anzahl der Eingangsimpulse über einen angegebenen Zeitraum. Gibt den Impulszählwert und die Zeitdauer in Nanosekunden (ns) aus. Die Werte werden am Ende des von der Frequenzmessperiode angegebenen Zeitraums erfasst und berechnet. Der Periodenbetrieb steht für Anweisung CTRL_HSC_EXT, aber nicht für Anweisung CTRL_HSC zur Verfügung.

10.1 Zählen (schnelle Zähler)

- Frequenz: Misst die Eingangsimpulse und die Zeitdauer und berechnet die Frequenz der Impulse. Das Programm gibt die Frequenz als doppelte Ganzzahl mit Vorzeichen in der Einheit Hz aus. Der Wert ist negativ, wenn rückwärts gezählt wird. Die Werte werden am Ende des von der Frequenzmessperiode angegebenen Zeitraums erfasst und berechnet.
- Bewegungssteuerung: Wird vom Technologieobjekt für die Bewegungssteuerung verwendet und steht nicht für die HSC-Anweisungen zur Verfügung. Weitere Informationen finden Sie unter "Bewegungssteuerung".

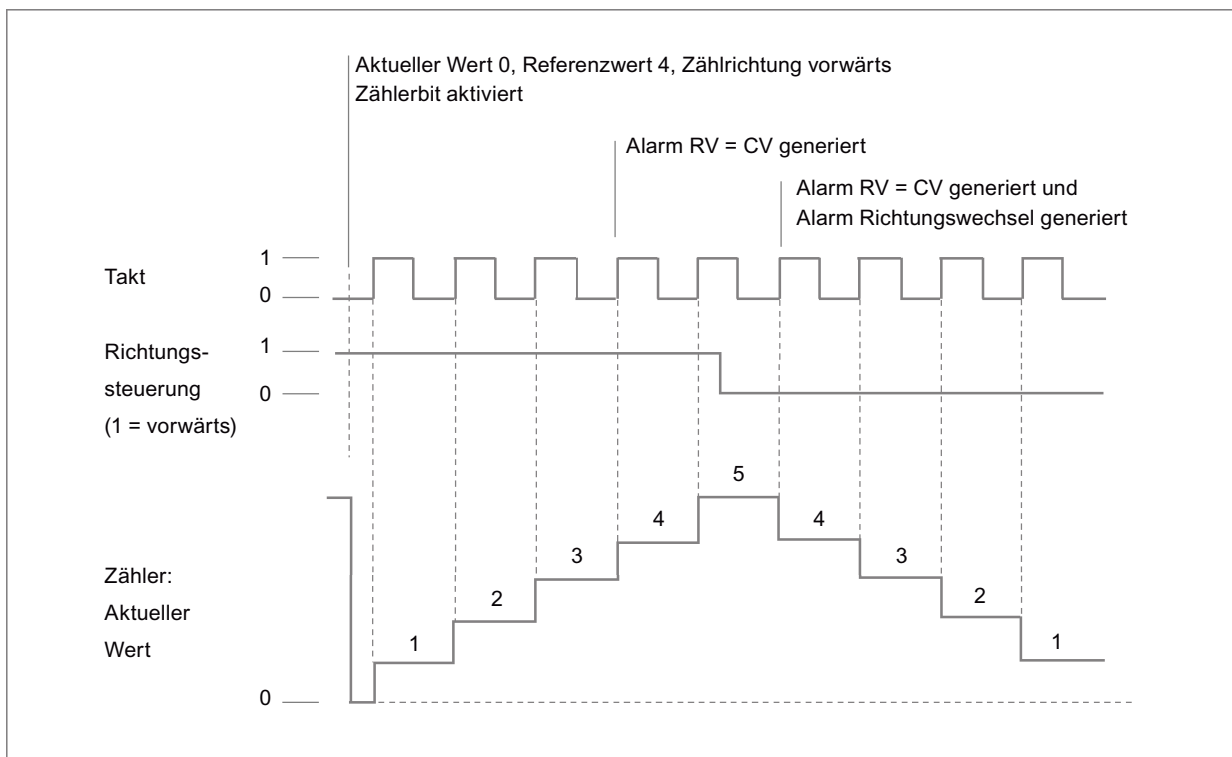
10.1.3.2 Betriebsphase

Wählen Sie die gewünschte Betriebsphase des HSC aus. Die vier Abbildungen unten zeigen, wann sich der Zählerwert ändert, wann das Ereignis auftritt, dass der aktuelle Wert (CV) gleich dem Referenzwert (RV) ist, und wann das Ereignis des Richtungswechsels auftritt.

Einphasenzähler

Einphasenzähler (nicht erhältlich für Bewegungssteuerung) zählt Impulse:

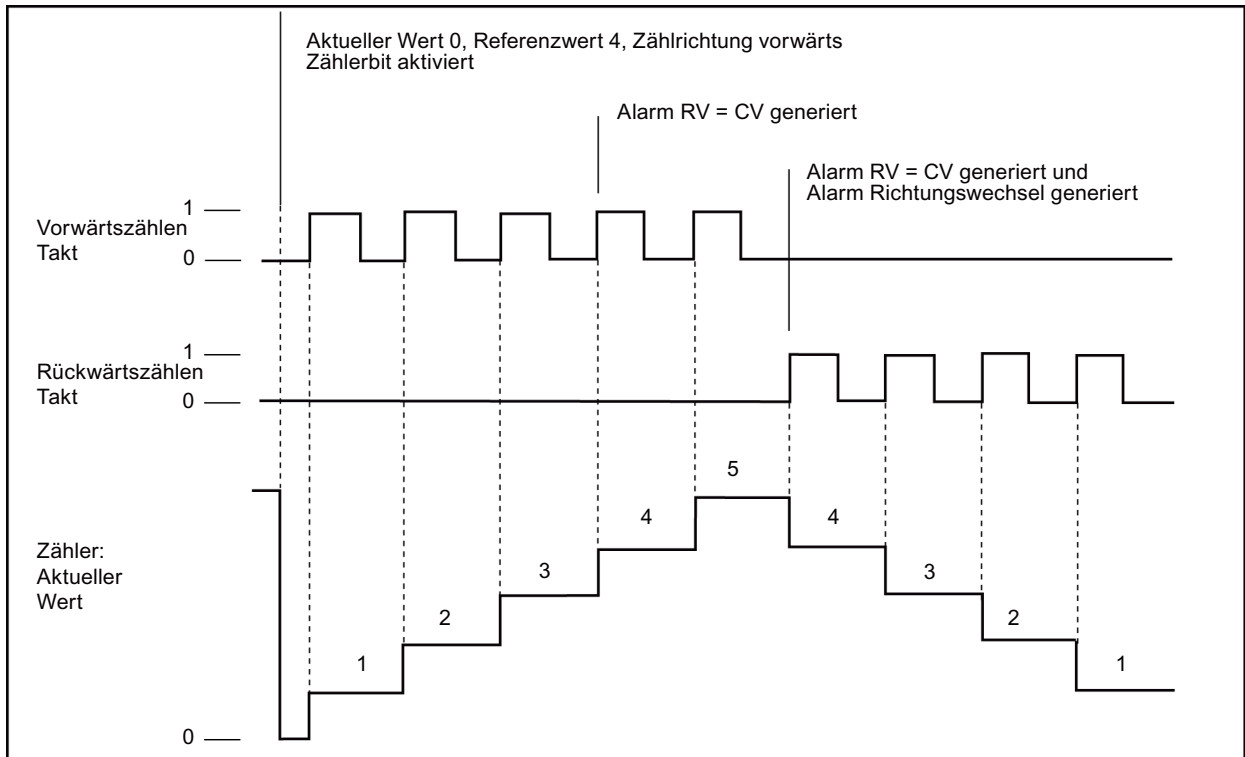
- Anwenderprogramm (interne Richtungssteuerung):
 - 1 ist vorwärts
 - -1 ist rückwärts
- Hardwareeingang (externe Richtungssteuerung):
 - Pegel High ist vorwärts.
 - Pegel Low ist rückwärts.



Zweiphasenzähler

Der Zweiphasenzähler zählt:

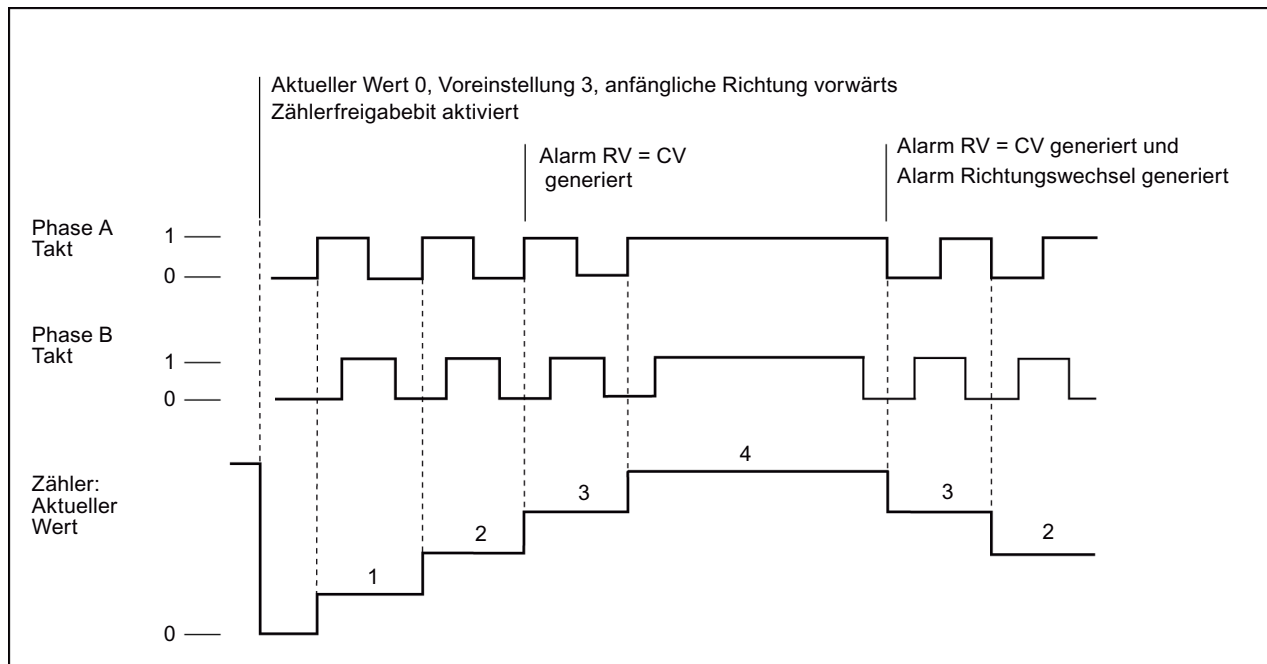
- Vorwärts beim Vorwärtzzähleingang
- Rückwärts beim Rückwärtzzähleingang



A/B-Zähler

Der A/B-Zähler zählt:

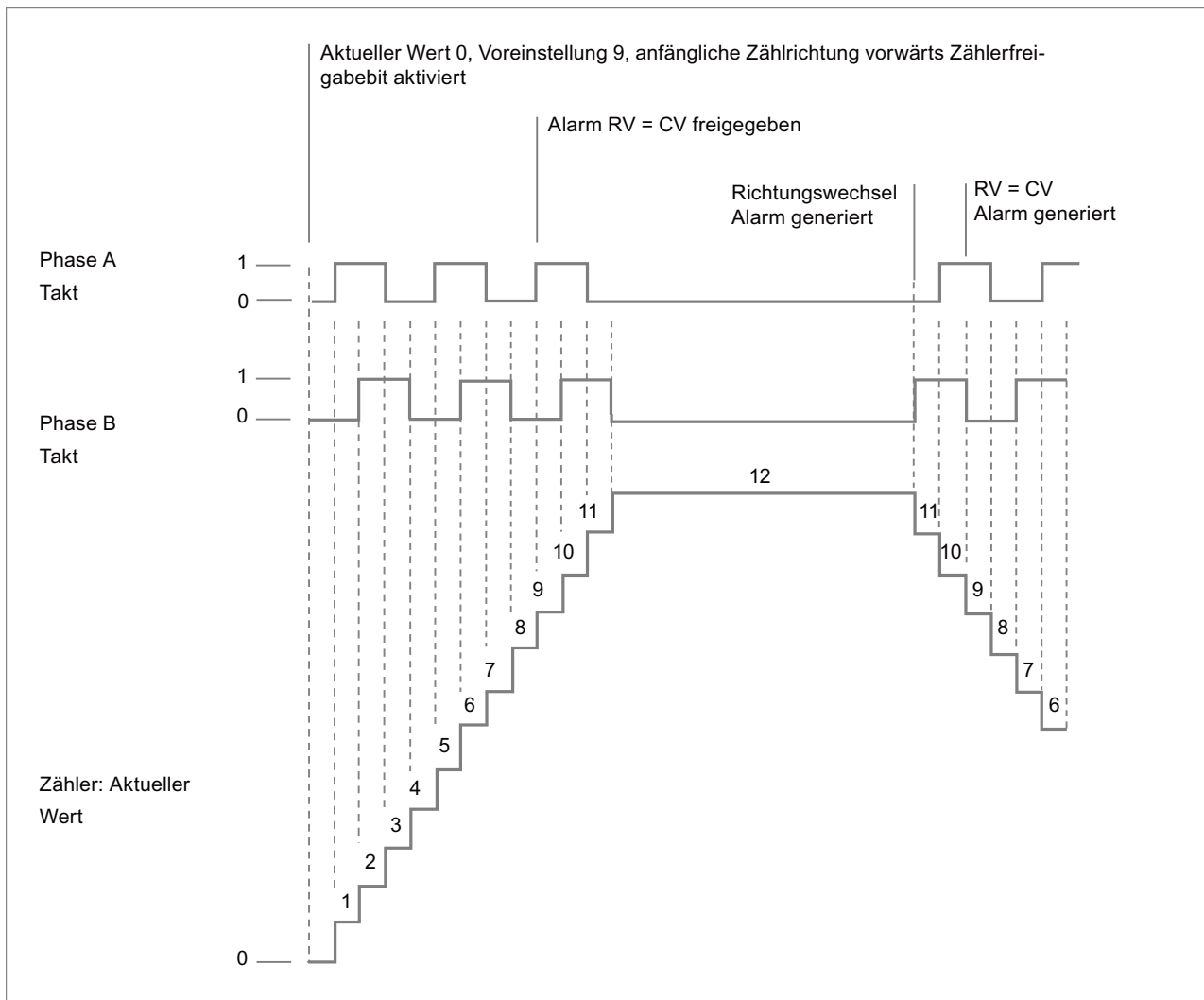
- Vorwärts bei einer steigenden Flanke am Zähl Eingang A, wenn der Zähl Eingang B Low ist
- Rückwärts bei einer fallenden Flanke am Zähl Eingang A, wenn der Zähl Eingang B Low ist



Vierfacher A/B-Zähler

Der vierfache A/B-Zähler zählt:

- Vorwärts bei einer steigenden Flanke am Zähleringang A, wenn der Zähleringang B Low ist
- Vorwärts bei einer fallenden Flanke am Zähleringang A, wenn der Zähleringang B High ist
- Vorwärts bei einer steigenden Flanke am Zähleringang B, wenn der Zähleringang A High ist
- Vorwärts bei einer fallenden Flanke am Zähleringang B, wenn der Zähleringang A Low ist
- Rückwärts bei einer steigenden Flanke am Zähleringang B, wenn der Zähleringang A Low ist
- Rückwärts bei einer fallenden Flanke am Zähleringang B, wenn der Zähleringang A High ist
- Rückwärts bei einer steigenden Flanke am Zähleringang A, wenn der Zähleringang B High ist
- Rückwärts bei einer fallenden Flanke am Zähleringang A, wenn der Zähleringang B Low ist



10.1.3.3 Anfangswerte

Bei jedem Wechsel der CPU in RUN lädt sie die Anfangswerte. Die Anfangswerte werden nur in der Betriebsart Zählen verwendet:

- Anfänglicher Zählerwert: Das Programm setzt den aktuellen Zählwert auf den anfänglichen Zählerwert, wenn die CPU von STOP nach RUN wechselt oder wenn das Programm den Synchronisierungseingang auslöst.
- Anfänglicher Referenzwert: Wenn der aktuelle Zählwert den Referenzwert erreicht, generiert das Programm, sofern die entsprechenden Funktionen eingerichtet sind, einen Alarm und/oder einen Ausgangsimpuls.
- Anfänglicher Referenzwert 2: Wenn der aktuelle Zählwert den Referenzwert 2 erreicht, generiert das Programm, sofern die Funktion eingerichtet ist, einen Ausgangsimpuls.

10.1 Zählen (schnelle Zähler)

- Anfänglicher oberer Grenzwert: Maximaler Zählwert. Der Standardwert ist der größtmögliche Wert von +2.147.483.647 Impulsen.
- Anfänglicher unterer Grenzwert: Minimaler Zählwert. Der Standardwert ist der kleinstmögliche Wert von -2.147.483.648 Impulsen.

Die oben genannten Werte und das Verhalten des Zählers bei Erreichen eines Grenzwerts sind nur in der Betriebsart Zählen verfügbar. Sie können die Werte und das Verhalten mit dem SDT HSC_Count der Anweisung CTRL_HSC_EXT anpassen.

10.1.3.4 Eingangsfunktionen

Die Takt- und Richtungseingänge bestimmen Zählereignisse und Zählrichtung basierend auf der Betriebsphase. Sie können nur die Synchronisierungs-, Erfassungs- und Gate-Eingänge in der Betriebsart Zählen verwenden und die Eingänge einzeln für die verschiedenen Arten von Auslösungen aktivieren und konfigurieren.

Synchronisierungseingang

Der Synchronisierungseingang setzt den aktuellen Zählwert auf den Startwert (den Anfangswert des Zählers). Den Synchronisierungseingang verwenden Sie üblicherweise zum Zurücksetzen des Zählers auf 0. Sie können die Synchronisierung auslösen, wenn sich der Eingangsanschluss in einem der folgenden Zustände befindet:

- High
- Low
- Wechselt von Low nach High
- Wechselt von High nach Low
- Wechselt von High nach Low oder von Low nach High

Erfassungseingang

Der Erfassungseingang setzt den erfassten Zählwert auf den Zählwert, der im Moment der Auslösung des Erfassungseingangs gespeichert wurde. Sie können die Erfassung auslösen, wenn sich der Eingangsanschluss in einem der folgenden Zustände befindet:

- Wechselt von Low nach High
- Wechselt von High nach Low
- Wechselt von High nach Low oder von Low nach High

Gate-Eingang

Der Gate-Eingang stoppt das Zählen des HSC. Sie können das Gate auslösen, wenn sich der Eingangsanschluss in einem der folgenden Zustände befindet:

- High
- Low

10.1.3.5 Ausgangsfunktion

Die Vergleichsausgangsfunktion ist die einzige Funktion für den HSC und steht nur in der Betriebsart Zählen zur Verfügung.

Vergleichsausgang

Sie können den Vergleichsausgang konfigurieren, um bei Auftreten eines der folgenden Ereignisse einen einzelnen Impuls zu generieren:

- Zähler entspricht Referenzwert (Zählrichtung vorwärts)
- Zähler entspricht Referenzwert (Zählrichtung rückwärts)
- Zähler entspricht Referenzwert (Zählrichtung vorwärts oder rückwärts)
- Zähler entspricht Referenzwert 2 (Zählrichtung vorwärts)
- Zähler entspricht Referenzwert 2 (Zählrichtung rückwärts)
- Zähler entspricht Referenzwert 2 (Zählrichtung vorwärts oder rückwärts)
- Positiver Überlauf
- Negativer Überlauf

Sie können die Ausgangsimpulse mit einer Zykluszeit von 1 bis 500 ms konfigurieren; die Standardzykluszeit beträgt 10 ms. Sie können die Impulsdauer bzw. die Einschaltdauer in einem Bereich von 1 bis 100 % festlegen; die Standardimpulsdauer beträgt 50 %.

Wenn innerhalb der angegebenen Zykluszeit mehrere Vergleichsausgangsereignisse auftreten, gehen die Ausgangsimpulse dieser Ereignisse verloren, weil der aktuelle Impuls seinen Zyklus noch nicht beendet hat. Sobald der Impuls beendet ist (die konfigurierte Zykluszeit abgelaufen ist), ist der Impulsgenerator zum Erzeugen eines neuen Impulses verfügbar.

10.1.3.6 Alarmereignisse

Im Bereich für die Ereigniskonfiguration können Sie einen Prozessalarm-OB aus dem Dropdown-Menü auswählen (oder einen neuen OB erstellen) und ihn einem HSC-Ereignis zuordnen. Die Priorität des Alarms liegt im Bereich von 2 bis 26, wobei 2 die niedrigste Priorität und 26 die höchste Priorität ist. Abhängig von der Konfiguration des HSC sind die folgenden Ereignisse verfügbar:

- Ereignis Zählerwert entspricht dem Referenzwert: Das Ereignis Zählerwert entspricht dem Referenzwert tritt auf, wenn der Zählwert des HSC den Referenzwert erreicht. Sie können den Referenzwert während der Konfiguration im Bereich für den Anfangsreferenzwert festlegen oder indem Sie mit der Anweisung CTRL_HSC_EXT den Wert "NewReference1" aktualisieren. Weitere Informationen finden Sie unter "Betriebsphase (Seite 562)".
- Synchronisierungsereignis: Eine Synchronisierung tritt immer dann auf, wenn Sie den Synchronisierungseingang aktivieren und auslösen.
- Richtungswechselereignis: Ein Richtungswechselereignis tritt auf, wenn die Zählrichtung wechselt. Weitere Informationen finden Sie unter "Betriebsphase (Seite 562)".

10.1.3.7 Anschlussbelegung des Hardwareeingangs


Wählen Sie für jeden HSC-Eingang, den Sie aktivieren, den gewünschten Eingang aus, entweder an der CPU oder auf dem optionalen Signalboard (Kommunikations- und Signalmodule unterstützen keine HSC-Eingänge). Wenn Sie einen Eingang auswählen, zeigt STEP 7 den maximalen Frequenzwert neben der Auswahl an. Die Einstellungen des digitalen Eingangsfilters müssen möglicherweise angepasst werden, damit alle Signalfrequenzen den Filter durchlaufen können. Wie Sie HSC-Eingangsfiler einrichten, erfahren Sie unter "Konfigurieren der Filterzeiten von Digitaleingängen (Seite 155)".

Hinweis

Eingangskanäle von CPU und SB (ab Firmware V4) haben konfigurierbare Eingangsfilerzeiten

Bei früheren Firmwareversionen waren die HSC-Eingangskanäle und auch die Filterzeiten festgelegt und konnten nicht geändert werden.

Ab Version V4 können Sie die Eingangskanäle und Filterzeiten festlegen. Die Standardeinstellung für den Eingangsfiler beträgt 6,4 ms, was die maximale Zählfrequenz auf 78 Hz begrenzt. Sie können die Filtereinstellungen ändern, um abhängig vom Aufbau Ihres Systems höhere oder niedrigere Frequenzen zu zählen.

 **WARNUNG**

Gefahren beim Ändern der Filterzeiteinstellung für digitale Eingangskanäle

Wenn die bisherige Einstellung der Filterzeit eines Digitaleingangskanals verändert wird, muss möglicherweise bis zu 20,0 ms lang ein neuer Eingangswert mit Pegel 0 vorhanden sein, damit der Filter vollständig auf neue Eingänge reagiert. Während dieses Zeitraums werden kurze Impulse mit Pegel "0", die kürzer als 20,0 ms sind, möglicherweise nicht erkannt oder gezählt.

Das Verändern der Filterzeiten kann ein unerwartetes Verhalten der Maschine oder des Prozesses verursachen, was zu schweren oder lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.

Damit die neue Filterzeit sofort wirksam wird, ist die CPU aus- und wieder einzuschalten.

Nutzen Sie bei der Auswahl die folgende Tabelle und achten Sie darauf, dass die verwendeten Eingangskanäle von CPU und SB die maximalen Impulsraten in Ihren Prozesssignalen unterstützen.

Tabelle 10-9 CPU-Eingang: Höchstfrequenz

CPU	CPU-Eingangskanal	Betriebsphase: Einphasenzähler oder Zweiphasenzähler	Betriebsphase: A/B-Zähler oder A/B-Zähler vierfach
1211C	Ea.0 bis Ea.5	100 kHz	80 kHz
1212C	Ea.0 bis Ea.5	100 kHz	80 kHz
	Ea.6, Ea.7	30 kHz	20 kHz
1214C und 1215C	Ea.0 bis Ea.5	100 kHz	80 kHz
	Ea.6 bis Eb.5	30 kHz	20 kHz

CPU	CPU-Eingangskanal	Betriebsphase: Einphasenzähler oder Zweiphasenzähler	Betriebsphase: A/B-Zähler oder A/B-Zähler vierfach
1217C	Ea.0 bis Ea.5	100 kHz	80 kHz
	Ea.6 bis Eb.1	30 kHz	20 kHz
	Eb.2 bis Eb.5 (0,2+, 0,2- bis 0,5+, 0,5-)	1 MHz	1 MHz

Tabelle 10-10 SB-Eingang: Höchstfrequenz (optionales Board)

Signalboard (SB)	SB-Eingangskanal	Betriebsphase: Einphasenzähler oder Zweiphasenzähler	Betriebsphase: A/B-Zähler oder A/B-Zähler vierfach
SB 1221, 200 kHz	Ee.0 bis Ee.3	200 kHz	160 kHz
SB 1223, 200 kHz	Ee.0, Ee.1	200 kHz	160 kHz
SB 1223	Ee.0, Ee.1	30 kHz	20 kHz

Beim Zuweisen eines Eingangs zu einer HSC-Funktion können Sie denselben Eingang mehreren HSC-Funktionen zuweisen. Beispiel: Die Zuweisung von E0.3 zum Synchronisierungseingang von HSC1 und zum Synchronisierungseingang von HSC2 für die Synchronisierung des Zählwerts beider HSCs gleichzeitig ist eine gültige Konfiguration. Sie erzeugt jedoch eine Warnung bei der Übersetzung.

Vermeiden Sie es nach Möglichkeit, einem Eingang mehrere Eingangsfunktionen desselben HSC zuzuweisen. Beispiel: Die Zuweisung von E0.3 zum Synchronisierungseingang und dem Gate-Eingang von HSC 1 für die Synchronisierung des Zählwerts und die Deaktivierung der Zählens gleichzeitig ist ebenfalls eine gültige Konfiguration. Sie können diese Konfiguration vornehmen, doch sie könnte zu unbeabsichtigten Ergebnissen führen.

 **WARNUNG**

Risiken bei der Zuweisung mehrerer Funktionen zu einem einzigen Digitaleingangskanal

Die Zuweisung mehrerer Eingangsfunktionen eines HSC zu einem gemeinsamen Eingang kann zu nicht vorhersehbaren Ergebnissen führen. Wenn an einem Eingang eine Auslösung auftritt, der mehrere Funktionen zugewiesen sind, kann die Reihenfolge, in der die Funktionen vom PLC ausgeführt werden, nicht ermittelt werden. Das ist als Wettlaufsituation bekannt und häufig nicht erwünscht.

Verändern der Filterzeiten kann ein unerwartetes Verhalten der Maschine oder des Prozesses verursachen, was zu schweren oder lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.

Um eine Wettlaufsituation zu vermeiden, weisen Sie nie mehr als zwei Eingangsfunktionen desselben HSC demselben Eingangsanschluss zu. Wenn bei einem HSC zwei Eingangsfunktionen demselben Anschluss zugewiesen sind, legen Sie die Auslösungen so fest, dass sie nie gleichzeitig auftreten können. Beachten Sie, dass eine fallende Flanke zum gleichen Zeitpunkt auftritt, zu dem ein Pegel Low beginnt, und dass eine steigende Flanke zum gleichen Zeitpunkt auftritt, zu dem ein Pegel High beginnt.

Hinweis

Sie weisen die von schnellen Zählern (HSCs) verwendeten digitalen Eingänge und Ausgänge während der Gerätekonfiguration der CPU zu. Wenn Sie HSC-Geräten Eingänge und Ausgänge zuweisen, können Sie die Werte dieser Eingänge und Ausgänge nicht mit Hilfe der Force-Funktion in einer Beobachtungstabelle ändern. Der HSC hat die volle Kontrolle über diese Eingänge und Ausgänge.

10.1.3.8 Anschlussbelegung des Hardwareausgangs

Wenn Sie den Vergleichsausgang aktivieren, wählen Sie einen freien Ausgang aus. Sobald Sie einen Ausgang zur Verwendung durch einen HSC (oder andere Technologieobjekte wie einen Impulsgenerator) konfiguriert haben, gehört dieser Ausgang exklusiv diesem Objekt. Keine andere Komponente kann den Ausgang verwenden, und der Ausgang kann nicht auf einen Wert geforct werden. Wenn Sie einen einzigen Ausgang für mehrere HSCs oder zur Verwendung in einem HSC und einem Impulsausgang konfigurieren, generiert das Programm einen Übersetzungsfehler.

10.1.3.9 HSC-Eingangsadressen

Jeder HSC nutzt ein Doppelwort im Speicherbereich E, in dem der aktuelle Zählwert gespeichert wird. Wenn Sie den HSC für die Betriebsart Frequenz konfigurieren, wird die Frequenz an dieser Eingangsadresse gespeichert. Der verfügbare Bereich der Eingangsadressen liegt von E0.0 bis E1023.7 (maximale Anfangsadresse ist E1020.0). Der HSC kann keine Eingangsadresse verwenden, die sich mit einer Eingangsadresse überschneidet, die einer anderen Komponente zugeordnet ist. Weitere Informationen zum Prozessabbild finden Sie unter "Ausführung des Anwenderprogramms (Seite 67)".

Die folgende Tabelle zeigt die Standardadressen für jeden HSC:

Tabelle 10-11 HSC-Standardadressen

Schneller Zähler (HSC)	Datentyp des aktuellen Werts	Standardadresse des aktuellen Werts
HSC1	DInt	ID 1000
HSC2	DInt	ID 1004
HSC3	DInt	ID 1008
HSC4	DInt	ID 1012
HSC5	DInt	ID 1016
HSC6	DInt	ID 1020

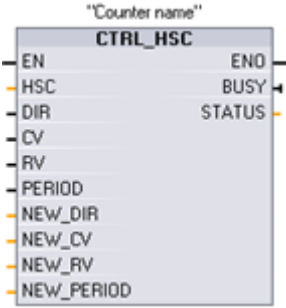
10.1.3.10 Hardwarekennung

Jeder HSC hat eine eindeutige Hardwarekennung, die von den Anweisungen HSC_CTRL und HSC_CTRL_EXT verwendet wird. Sie finden die PLC-Variable für die Hardwarekennung unter "Systemkonstanten". Ein HSC mit dem Namen "HSC_1" hat die Variable "Local~HSC_1" und den Datentyp "Hw_Hsc". Diese Variable wird auch bei der Auswahl des HSC-Eingangs für die Anweisungen CTRL_HSC_EXT im Dropdown-Menü angezeigt.

10.1.4 Frühere Anweisung CTRL_HSC (Schnellen Zähler steuern)

10.1.4.1 Übersicht über die Anweisung

Tabelle 10-12 Anweisung CTRL_HSC (für allgemeine Zählaufgaben)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Jede Anweisung CTRL_HSC (Schnellen Zähler steuern) nutzt eine in einem DB abgelegte Struktur, um Zählerdaten zu speichern. Sie weisen den DB zu, wenn Sie die Anweisung CTRL_HSC im Editor einfügen.</p>

- ¹ Wenn Sie die Anweisung einfügen, zeigt STEP 7 den Dialog "Aufrufoptionen" zum Erstellen des zugehörigen DB an.
- ² Im SCL-Beispiel ist "CTRL_HSC_1_DB" der Name des Instanz-DBs.

Tabelle 10-13 Datentypen für die Parameter

Parameter	Deklaration	Datentyp	Beschreibung
HSC	IN	HW_HSC	HSC-Kennung
DIR ^{1, 2}	IN	Bool	1 = Neue Richtung anfordern
CV ¹	IN	Bool	1 = Anforderung zum Setzen eines neuen Zählwerts
RV ¹	IN	Bool	1 = Anforderung zum Setzen eines neuen Referenzwerts
PERIOD ¹	IN	Bool	1 = Anforderung zum Setzen eines neuen Zeitintervalls (nur bei Frequenzmessung)
NEW_DIR	IN	Int	Neue Richtung: 1= vorwärts, -1= rückwärts
NEW_CV	IN	DInt	Neuer Zählwert
NEW_RV	IN	DInt	Neuer Referenzwert

10.1 Zählen (schnelle Zähler)

Parameter	Deklaration	Datentyp	Beschreibung
NEW_PERIOD	IN	Int	Der neue Zeitintervallwert wird in Millisekunden angegeben (nur bei Frequenzmessung). Zulässig sind nur die Werte 10, 100 oder 1000 Millisekunden: 1000 = 1 Sekunde 100 = 0,1 Sekunde 10 = 0,01 Sekunde
BUSY ³	OUT	Bool	Funktion besetzt
STATUS	OUT	Word	Ausführungsbedingung

- ¹ Wird keine Aktualisierung eines Parameterwerts angefordert, so werden die entsprechenden Eingangswerte ignoriert.
- ² Der Parameter DIR ist nur gültig, wenn für die konfigurierte Zählrichtung die Programmsteuerung (interne Richtungssteuerung) vorgegeben ist. Sie legen fest, wie dieser Parameter in der HSC-Gerätekonfiguration genutzt wird.
- ³ Bei einem HSC in der CPU oder im SB hat der Parameter BUSY immer den Wert 0.

Sie konfigurieren die Parameter für jeden HSC in der Gerätekonfiguration der CPU: Zähl-/Frequenzfunktion, Rücksetzoptionen, Alarmereigniskonfiguration, Hardware-E/A und Adresse des Zählwerts.

Einige der Parameter für den HSC können von Ihrem Anwenderprogramm geändert werden, um die Programmsteuerung für den Zählvorgang vorzugeben:

- Setzen der Zählrichtung auf einen Wert NEW_DIR
- Setzen des aktuellen Zählwerts auf einen neuen Wert NEW_CV
- Setzen des Referenzwerts auf einen neuen Wert NEW_RV
- Setzen des Zeitintervallwerts (nur bei Frequenzmessung) auf einen neuen Wert NEW_PERIOD

Sind die folgenden Booleschen Merker auf 1 gesetzt, wenn die Anweisung CTRL_HSC ausgeführt wird, so wird der entsprechende Wert NEW_xxx in den Zähler geladen. Mehrere Anforderungen (mehrere Merker sind gleichzeitig gesetzt) werden in einer Ausführung der Anweisung CTRL_HSC verarbeitet.

- DIR = 1 ist eine Anforderung zum Laden eines Werts NEW_DIR, 0 = keine Änderung
- CV = 1 ist eine Anforderung zum Laden eines Werts NEW_CV, 0 = keine Änderung
- RV = 1 ist eine Anforderung zum Laden eines Werts NEW_RV, 0 = keine Änderung
- PERIOD = 1 ist eine Anforderung zum Laden eines Werts NEW_PERIOD, 0 = keine Änderung

Bei einem Fehler wird ENO auf 0 gesetzt und der Ausgang STATUS gibt einen Bedingungscode an:

Tabelle 10-14 Execution condition codes

STATUS (W#16#)	Beschreibung
0	Kein Fehler
80A1	HSC-Kennung adressiert keinen HSC
80B1	Ungültiger Wert in NEW_DIR
80B2	Ungültiger Wert in NEW_CV
80B3	Ungültiger Wert in NEW_RV
80B4	Ungültiger Wert in NEW_PERIOD

STATUS (W#16#)	Beschreibung
80C0	Mehrmaliger Zugriff auf den schnellen Zähler Dieser Fehler kann auftreten, wenn für die Zählart (Seite 561) "Periode" oder "Bewegungssteuerung" eingestellt ist. Diese Zählarten sind für die Anweisung CTRL_HSC ungültig und werden nur von der Anweisung CTRL_HSC_EXT unterstützt.
80D0	Schneller Zähler (HSC) nicht in der CPU-Hardwarekonfiguration aktiviert

10.1.4.2 Verwenden von CTRL_HSC

Die Anweisung CTRL_HSC wird typischerweise in einem Prozessalarm-OB platziert, der ausgeführt wird, wenn das Prozessalarmereignis des Zählers ausgelöst wird. Wird zum Beispiel der Zählerinterrupt durch ein Ereignis CV=RV ausgelöst, so führt ein Prozessalarm-OB die Anweisung CTRL_HSC aus und kann den Referenzwert durch Laden eines Werts NEW_RV ändern.

Der aktuelle Zählwert ist in den Parametern für CTRL_HSC nicht vorgesehen. Die Adresse des Prozessabbilds, in der der aktuelle Zählwert gespeichert wird, wird bei der Hardwarekonfiguration dem schnellen Zähler zugewiesen. Sie können den Zählwert über die Programmlogik direkt auslesen. Der an Ihr Programm ausgegebene Wert ist ein korrekter Zählwert für den Moment, in dem der Zähler gelesen wurde. Der Zähler setzt die Zählung schneller Ereignisse fort. Der tatsächliche Zählwert kann sich deshalb ändern, bevor Ihr Programm einen Prozess mit einem alten Zählwert beendet.

10.1.4.3 Aktueller Zählwert des HSC

Die CPU speichert den aktuellen Wert jedes HSC in der Adresse eines Eingangs (E). Die folgende Tabelle zeigt die Standardadressen für den aktuellen Wert jedes HSC. Sie können die E-Adresse für den aktuellen Wert ändern, indem Sie die Eigenschaften der CPU in der Gerätekonfiguration ändern.

Schnelle Zähler speichern den aktuellen Zählwert in einem DInt-Wert. Ein DInt-Zählwert hat einen Bereich von -2147483648 bis +2147483647. Ab den CPUs mit der Firmware V4.2 können Sie die Bereichsgrenzen konfigurieren. Weitere Informationen finden Sie unter "Anfangswerte (Seite 565)".

Beim Vorwärtszählen schlägt der Zähler vom maximalen positiven Wert zum maximalen negativen Wert um und beim Rückwärtszählen vom maximalen negativen Wert zum maximalen positiven Wert. Die Frequenz wird in der Einheit Hz ausgegeben (Beispiel: 123,4 Hz wird als 123 ausgegeben).

Tabelle 10-15 HSC-Standardadressen

HSC	Datentyp des aktuellen Werts	Standardadresse des aktuellen Werts
HSC1	DInt	ED1000
HSC2	DInt	ED1004
HSC3	DInt	ED1008
HSC4	DInt	ED1012
HSC5	DInt	ED1016
HSC6	DInt	ED1020

10.2 Bewegungssteuerung

10.2.1 Übersicht über die Bewegungssteuerung

Das TIA Portal unterstützt Sie zusammen mit der Bewegungssteuerungsfunktionalität der S7-1200 CPU beim Steuern von Schrittmotoren und Servomotoren:

- Sie konfigurieren die Positionierachse und die Technologieobjekte der Befehlstabelle im TIA Portal. Die S7-1200 CPU steuert mit diesen Technologieobjekten die Ausgänge, die wiederum die Antriebe steuern.
- Im Anwenderprogramm steuern Sie die Achse mit Hilfe von Bewegungssteuerungsanweisungen und veranlassen die Bewegungsbefehle Ihres Antriebs.

10.2.2 Hardwarekomponenten für die Bewegungssteuerung

Die grundlegende Hardwarekonfiguration für eine Bewegungssteuerungsanwendung mit der S7-1200 CPU besteht aus den folgenden Elementen.

S7-1200 CPU

Die S7-1200 CPU verbindet die Funktionalität einer speicherprogrammierbaren Steuerung mit der Bewegungssteuerungsfunktionalität für den Betrieb von Antrieben. Die Bewegungssteuerungsfunktion übernimmt die Steuerung und Überwachung der Antriebe.

Signalboard

Mit Signalboards erweitern Sie die CPU um zusätzliche Eingänge und Ausgänge. Die Digitalausgänge können Sie nach Bedarf als Impulsgeberausgänge zum Steuern von Antrieben verwenden. Bei CPUs mit Relaisausgängen kann das Impulssignal nicht an den integrierten Ausgängen ausgegeben werden, weil die Relais die benötigten Schaltfrequenzen nicht unterstützen. Um an diesen CPUs mit der Impulsfolge (PTO) zu arbeiten, müssen Sie ein Signalboard mit digitalen Ausgängen nutzen.

Die Analogausgänge können Sie nach Bedarf zum Steuern analoger Antriebe verwenden.

PROFINET

Stellen Sie über die PROFINET-Schnittstelle eine Online-Verbindung zwischen der S7-1200 CPU und dem Programmiergerät her. Neben den Online-Funktionen der CPU stehen für die Bewegungssteuerung weitere Inbetriebnahme- und Diagnosefunktionen zur Verfügung.

PROFINET unterstützt weiterhin das PROFIdrive-Profil für die Verbindung von PROFIdrive-fähigen Antrieben und Gebern.

Antriebe und Geber

Antriebe gestatten die Bewegung der Achse. Geber liefern die Istposition für die Lageregelung der Achse.

Die nachstehende Tabelle zeigt die Verbindungsmöglichkeiten für Antriebe und Geber:

Antriebsverbindung	Regelung/Steuerung der Achse	Geberverbindung
Impulsfolge (PTO) (Schrittmotoren und Servomotoren mit Impulsschnittstelle)	Lagegeregelt	-
Analogausgang (AQ)	Lagegeregelt	<ul style="list-style-type: none"> • Geber am schnellen Zähler (HSC) • Geber am Technologiemodul (TM) • Geber auf PROFINET
PROFINET	Lagegeregelt	<ul style="list-style-type: none"> • Geber am Antrieb • Geber am schnellen Zähler (HSC) • Geber am Technologiemodul (TM) • Geber auf PROFINET

10.2.3 Bewegungssteuerungsanweisungen

10.2.3.1 Übersicht MC-Anweisungen

Die Bewegungssteuerungsanweisungen verwenden einen zugehörigen Technologie-Datenbaustein und die dafür zugewiesene Impulsfolge (PTO) der CPU, um die Bewegung einer Achse zu steuern:

- MC_Power (Seite 576) aktiviert und deaktiviert eine Achse für die Bewegungssteuerung.
- MC_Reset (Seite 577) setzt alle Bewegungssteuerungsfehler zurück. Alle Bewegungssteuerungsfehler, die quittiert werden können, werden quittiert.
- MC_Home (Seite 578) stellt die Beziehung zwischen dem Achsensteuerungsprogramm und dem mechanischen Positionierungssystem der Achse her.
- MC_Halt (Seite 578) storniert alle Bewegungsvorgänge und bewirkt den Stopp der Achsenbewegung. Die Halteposition ist nicht definiert.
- MC_MoveAbsolute (Seite 579) startet die Bewegung zu einer absoluten Position. Der Auftrag ist beendet, wenn die Zielposition erreicht ist.
- MC_MoveRelative (Seite 579) startet eine Positionierbewegung relativ zur Startposition.
- MC_MoveVelocity (Seite 580) bewirkt, dass sich die Achse mit der angegebenen Geschwindigkeit bewegt.
- MC_MoveJog (Seite 580) führt den Tippbetrieb zu Test- und Anlaufzwecken aus.
- MC_CommandTable (Seite 581) führt Achssteuerungsbefehle als Bewegungsfolge aus.
- MC_ChangeDynamic (Seite 582) ändert die Dynamikeinstellungen der Achse.

- MC_WriteParam (Seite 581) schreibt bestimmte Parameter, um die Funktionalität der Achse im Anwenderprogramm zu ändern.
- MC_ReadParam (Seite 582) liest bestimmte Parameter, die die aktuelle Position, Geschwindigkeit usw. der im Achseneingang definierten Achse angeben.

CPU-Firmwarestände

Wenn Sie mit einer S7-1200 CPU ab Firmware V4.1 arbeiten, wählen Sie Version V5.0 der Bewegungsanweisungen.

Wenn Sie mit einer S7-1200 CPU mit Firmwareversion V4.0 oder niedriger arbeiten, wählen Sie die entsprechende Version V4.0, V3.0, V2.0 oder V1.0 der Bewegungsanweisungen.

Wenn Sie mit einer S7-1200 CPU ab Firmware V4.2 arbeiten, wählen Sie Version V6.0 der Bewegungsanweisungen.

Wenn Sie mit einer S7-1200 CPU ab Firmware V4.4 arbeiten, wählen Sie Version V7.0 der Bewegungsanweisungen.

Wenn Sie mit einer S7-1200 CPU ab Firmware V4.5 arbeiten, wählen Sie Version V8.0 der Bewegungsanweisungen.

Hinweis

Die Anweisungen der Bewegungssteuerung V1.0 bis V3.0 steuern aktiv den Ausgang der Anweisung. Tritt im Baustein ein Fehler auf, wird der Freigabeausgang (ENO) in den AUS-Zustand geschaltet. Ein Fehler wird über die Ausgänge ERROR, ErrorID und ErrorInfo am Baustein gemeldet. Über den ENO-Ausgang kann der Zustand der Anweisung ausgewertet werden, und nachfolgende Anweisungen können der Reihe nach ausgeführt werden.

Bei den Anweisungen der Bewegungssteuerung V4.0 und V5.0 wird der EN-Eingang immer auf den ENO-Ausgang durchgereicht, unabhängig vom Fehlerzustand des Bausteins. Dies kann dazu führen, dass ein Programm, das mit der Bewegungssteuerung V3.0 oder früher arbeitet und vom ENO-Zustand abhängig ist, nicht richtig funktioniert. Zur Behebung dieser Situation werten Sie, wenn Sie die Bewegungssteuerung V4.0 oder höher verwenden, den Zustand der Anweisung mit den Ausgängen DONE und ERROR und nicht mit dem ENO-Ausgang aus.

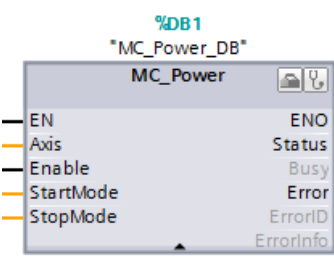
10.2.3.2 MC_Power (Achse freigeben/sperrn)

Die Anweisung MC_Power aktiviert und deaktiviert eine Achse für die Bewegungssteuerung.

Hinweis

Wenn die Achse wegen eines Fehlers ausgeschaltet wird, wird sie nach Behebung und Quittierung des Fehlers automatisch wieder aktiviert. Hierfür ist erforderlich, dass der Eingangsparameter Enable den Wert WAHR während dieses Vorgangs gespeichert hat.

Tabelle 10-16 Anweisung MC_Power


KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Die Bewegungssteuerungsanweisung MC_Power aktiviert oder deaktiviert eine Achse. Bevor Sie die Achse aktivieren oder deaktivieren können, prüfen Sie die folgenden Bedingungen:</p> <ul style="list-style-type: none"> • Das Technologieobjekt wurde korrekt konfiguriert. • Es steht kein Freigabe verhindernder Fehler an. <p>Die Ausführung von MC_Power kann nicht von einer Bewegungssteuerungsaufgabe abgebrochen werden. Durch Deaktivieren der Achse (Eingangsparameter Enable = FALSCH) werden alle Bewegungssteuerungsaufgaben für das zugehörige Technologieobjekt abgebrochen.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_Power_DB" der Name des Instanz-DBs.

10.2.3.3 MC_Reset (Fehler bestätigen)

Die Anweisung MC_Reset setzt alle Bewegungssteuerungsfehler zurück. Alle Bewegungssteuerungsfehler, die quittiert werden können, werden quittiert.

Tabelle 10-17 Anweisung MC_Reset

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_Reset quittieren Sie "Betriebsfehler mit Achsenstopp" und "Konfigurationsfehler". Die Fehler, die quittiert werden müssen, finden Sie in der "Liste von ErrorIDs und ErrorInfos" unter "Abhilfe".</p> <p>Bevor Sie die Anweisung MC_Reset verwenden, müssen Sie die Ursache eines anstehenden zu quittierenden Konfigurationsfehlers behoben haben (indem Sie beispielsweise einen ungültige Beschleunigungswert im Technologieobjekt "Achse" in einen gültigen Wert ändern).</p> <p>Ab Version 3.0 ermöglicht der Befehl Restart das Laden der Achsenkonfiguration in den Arbeitsspeicher im Betriebszustand RUN.</p>

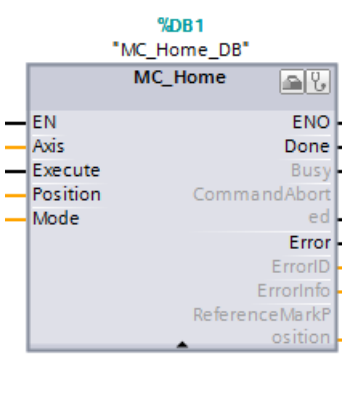
- 1 STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_Reset_DB" der Name des Instanz-DBs.

Die Aufgabe MC_Reset kann von keiner anderen Bewegungssteuerungsaufgabe abgebrochen werden. Die neue Aufgabe MC_Reset bricht keine anderen aktiven Bewegungssteuerungsaufgaben ab.

10.2.3.4 MC_Home (Referenzpunktfahrt der Achse durchführen)

Die Anweisung MC_Home stellt die Beziehung zwischen dem Achsensteuerungsprogramm und dem mechanischen Positionierungssystem der Achse her.

Tabelle 10-18 Anweisung MC_Home

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_Home passen Sie die Achsenkoordinaten an die reale, physikalische Position des Antriebs an. Für die absolute Positionierung der Achse ist eine Referenzpunktfahrt erforderlich:</p> <p>Um die Anweisung MC_Home zu verwenden, muss die Achse zunächst freigegeben werden.</p>

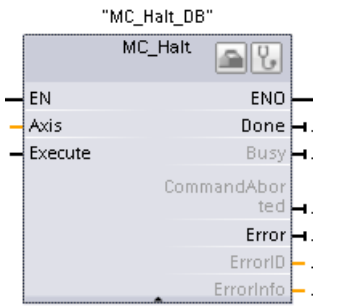
¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "MC_Home_DB" der Name des Instanz-DBs.

10.2.3.5 MC_Halt (Achse pausieren)

Die Anweisung MC_Halt bricht alle Bewegungsvorgänge ab und bewirkt den Stopp der Achsenbewegung. Die Halteposition ist nicht definiert.

Tabelle 10-19 Anweisung MC_Halt

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_Halt stoppen Sie alle Bewegungen und bringen die Achse zum Stillstand. Die Stillstandposition ist nicht definiert.</p> <p>Um die Anweisung MC_Halt zu verwenden, muss die Achse zunächst freigegeben werden.</p>

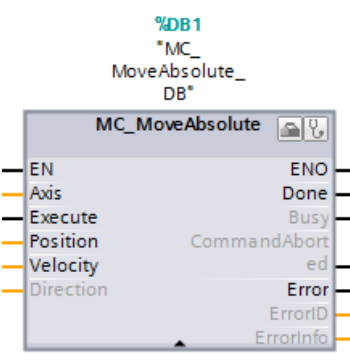
¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "MC_Halt_DB" der Name des Instanz-DBs.

10.2.3.6 MC_MoveAbsolute (Achse absolut positionieren)

Die Anweisung MC_MoveAbsolute startet die Bewegung zu einer absoluten Position. Der Auftrag ist beendet, wenn die Zielposition erreicht ist.

Tabelle 10-20 Anweisung MC_MoveAbsolute

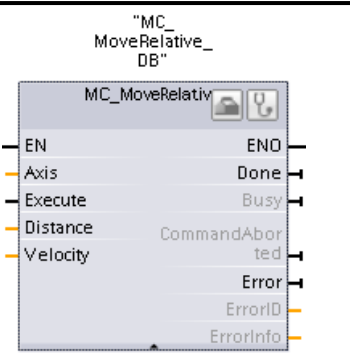
KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveAbsolute_DB" (Axis:= _multi_fb_in_, Execute:= _bool_in_, Position:= _real_in_, Velocity:= _real_in_, Direction:= _int_in_, Done=> _bool_out_, Busy=> _bool_out_, CommandAborted=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>Mit der Anweisung MC_MoveAbsolute starten Sie eine Positionierbewegung der Achse zu einer absoluten Position.</p> <p>Um die Anweisung MC_MoveAbsolute zu verwenden, muss die Achse zunächst freigegeben und zum Referenzpunkt gefahren werden.</p>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC_MoveAbsolute_DB" der Name des Instanz-DBs.

10.2.3.7 MC_MoveRelative (Achse relativ positionieren)

Die Anweisung MC_MoveRelative startet eine Positionierbewegung relativ zur Startposition.

Tabelle 10-21 Anweisung MC_MoveRelative

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_MoveRelative starten Sie eine Positionierbewegung relativ zur Startposition.</p> <p>Um die Anweisung MC_MoveRelative zu verwenden, muss die Achse zunächst freigegeben werden.</p>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC_MoveRelative_DB" der Name des Instanz-DBs.

10.2.3.8 MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen)

Die Anweisung MC_MoveVelocity bewirkt, dass sich die Achse mit der angegebenen Geschwindigkeit bewegt.

Tabelle 10-22 Anweisung MC_MoveVelocity

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_MoveVelocity bewegen Sie die Achse konstant mit der angegebenen Geschwindigkeit.</p> <p>Um die Anweisung MC_MoveVelocity zu verwenden, muss die Achse zunächst freigegeben werden.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "MC_MoveVelocity_DB" der Name des Instanz-DBs.

10.2.3.9 MC_MoveJog (Achse im Tipbetrieb bewegen)

Die Anweisung MC_MoveJog führt den Tipbetrieb zu Test- und Anlaufzwecken aus.

Tabelle 10-23 Anweisung MC_MoveJog

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung MC_MoveJog bewegen Sie die Achse konstant mit der angegebenen Geschwindigkeit im Tipbetrieb. Diese Anweisung dient üblicherweise zu Test- und Inbetriebnahmezwecken.</p> <p>Um die Anweisung MC_MoveJog zu verwenden, muss die Achse zunächst freigegeben werden.</p>


¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

² Im SCL-Beispiel ist "MC_MoveJog_DB" der Name des Instanz-DBs.

10.2.3.10 MC_CommandTable (Achssteuerungsbefehle als Bewegungsfolge ausführen)

Die Anweisung MC_CommandTable führt Achssteuerungsbefehle als Bewegungsfolge aus.

Tabelle 10-24 Anweisung MC_CommandTable

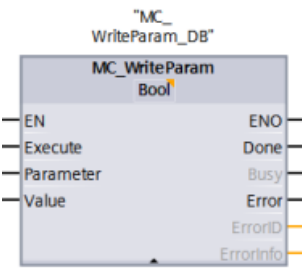
KOP/FUP	SCL	Beschreibung
	<pre>"MC_CommandTable_DB" (Axis:= multi_fb_in_, CommandTable:= multi_fb_in_, Execute:= bool_in_, StartStep:= uint_in_, EndStep:= uint_in_, Done=> bool_out_, Busy=> bool_out_, CommandAborted=> bool_out_, Error=> bool_out_, ErrorID=> word_out_, ErrorInfo=> word_out_, CurrentStep=> uint_out_, StepCode=> word_out_);</pre>	<p>Führt eine Reihe einzelner Bewegungen für eine Motorsteuersachse aus, die zu einer Bewegungsfolge verbunden werden können.</p> <p>Einzelne Bewegungen werden in einer Befehlstabelle eines Technologieobjekts für die Impulsfolge konfiguriert (TO_CommandTable_PTO).</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_CommandTable_DB" der Name des Instanz-DBs.

10.2.3.11 MC_WriteParam (Parameter eines Technologieobjekts schreiben)

Mit der Anweisung MC_WriteParam können Sie bestimmte Parameter schreiben, um die Funktionalität der Achse im Anwenderprogramm zu ändern.

Tabelle 10-25 Anweisung MC_WriteParam

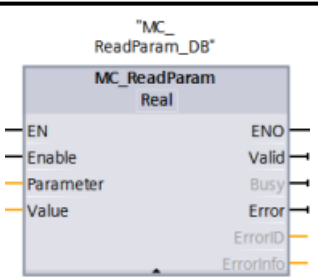
KOP/FUP	SCL	Beschreibung
	<pre>"MC_WriteParam_DB" (Parameter:= variant_in_, Value:= variant_in_, Execute:= bool_in_, Done:= bool_out_, Error:= real_out_, ErrorID:= word_out_, ErrorInfo:= word_out_);</pre>	<p>Mit der Anweisung MC_WriteParam können Sie in öffentliche Parameter schreiben (z. B. Beschleunigung und Werte in Anwender-DBs).</p>

- 1 STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_WriteParam_DB" der Name des Instanz-DBs.

10.2.3.12 Anweisung MC_ReadParam (Parameter des Technologieobjekts lesen)

Mit der Anweisung MC_ReadParam lesen Sie eine ausgewählte Anzahl von Parametern, die die aktuelle Position, Geschwindigkeit usw. der im Eingang "Axis" definierten Achse angeben.

Tabelle 10-26 Anweisung MC_ReadParam

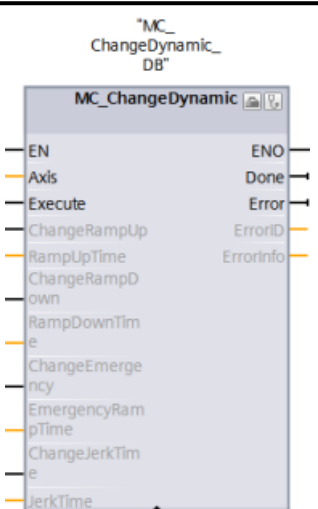
KOP/FUP	SCL	Beschreibung
	<pre>"MC_ReadParam_DB" (Enable:=_bool_in_, Parameter:=_variant_in_, Value:=_variant_in_out_, Valid:=_bool_out_, Busy:=_bool_out_, Error:=_real_out_, ErrorID:=_word_out_, ErrorInfo:=_word_out_);</pre>	<p>Mit der Anweisung MC_ReadParam lesen Sie einzelne Statuswerte unabhängig vom Steuerpunkt im Zyklus.</p>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_ReadParam_DB" der Name des Instanz-DBs.

10.2.3.13 MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern)

Mit der Anweisung MC_ChangeDynamic ändern Sie die Dynamikeinstellungen einer Bewegungssteuerungsachse.

Tabelle 10-27 Anweisung MC_ChangeDynamic

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Ändert die Dynamikeinstellungen einer Bewegungssteuerungsachse:</p> <ul style="list-style-type: none"> • Wert der Hochlaufzeit (Beschleunigung) ändern • Wert der Rücklaufzeit (Verzögerung) ändern • Wert der Not-Aus-Rücklaufzeit (Not-Aus-Verzögerung) ändern • Wert der Glättungszeit (Ruck) ändern

- 1 STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_ChangeDynamic_DB" der Name des Instanz-DBs.

10.2.4 Weitere Informationen zur Bewegungssteuerung mit S7-1200

Weitere Informationen zur S7-1200 Bewegungssteuerung finden Sie unter SIMATIC STEP 7 S7-1200 Motion Control V6.0 im TIA Portal V15 (<https://support.industry.siemens.com/cs/us/en/view/109773400>).

In diesem Handbuch finden Sie ausführliche Informationen zu den folgenden Themen:

- Impulsgenerator konfigurieren
- Ungeregelte Bewegungssteuerung
- Achse konfigurieren
- Inbetriebnahme
- Geregeltete Bewegungssteuerung
- Achse konfigurieren
- ServoOBs
- Drehzahl geregelter Betrieb
- Unterstützung von Telegramm 4
- Simulationsachse
- Datenübernahme
- Achssteuerung mit dem Technologiemodul (TM) Pulse
- TO_CommandTable_PTO konfigurieren
- Funktionsweise der Bewegungssteuerung bei der S7-1200
- Für Bewegungssteuerung verwendete CPU-Ausgänge
- Hardware- und Softwareendschalter für die Bewegungssteuerung
- Referenzpunktfahrt
- Ruckbegrenzung
- Aktive Befehle überwachen
- MC-Anweisungen mit einem Ausgangsparameter "Done" überwachen
- MC_Velocity überwachen
- MC_MoveJog überwachen
- ErrorIDs und ErrorInfos für die Bewegungssteuerung

Siehe auch

SIMATIC STEP 7 S7-1200 Motion Control V6.0 in TIA Portal V15 (<https://support.industry.siemens.com/cs/us/en/view/109773400>)

10.3 PID-Regelung

10.3.1 PID-Funktionalität

STEP 7 bietet die folgenden PID-Anweisungen (Proportional-Integral-Differenzial) für die S7-1200 CPU:

- Die Anweisung PID_Compact dient zum Regeln technischer Prozesse mit kontinuierlichen Eingangs- und Ausgangsvariablen.
- Die Anweisung PID_3Step dient zum Regeln von motorbetätigten Geräten wie Ventilen, die digitale Signale zum Öffnen und Schließen benötigen.
- Die Anweisung PID_Temp bietet einen universellen PID-Regler für die speziellen Anforderungen der Temperaturregelung.

10.3.2 PID-Anweisungen

10.3.2.1 PID-Funktionalität

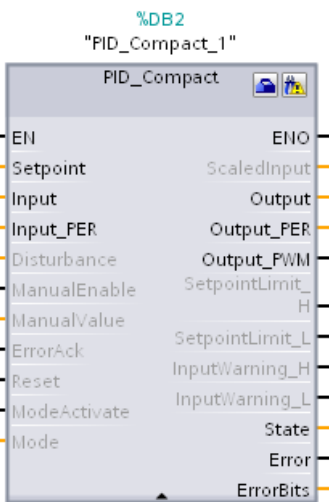
STEP 7 bietet die folgenden PID-Anweisungen (Proportional-Integral-Differenzial) für die S7-1200 CPU:

- Die Anweisung PID_Compact dient zum Regeln technischer Prozesse mit kontinuierlichen Eingangs- und Ausgangsvariablen.
- Die Anweisung PID_3Step dient zum Regeln von motorbetätigten Geräten wie Ventilen, die digitale Signale zum Öffnen und Schließen benötigen.
- Die Anweisung PID_Temp bietet einen universellen PID-Regler für die speziellen Anforderungen der Temperaturregelung.

10.3.2.2 Anweisung PID_Compact

Die Anweisung PID_Compact bietet einen universellen PID-Regler mit integrierter Selbsteinstellung für den Automatik- und Handbetrieb.

Tabelle 10-28 Anweisung PID_Compact

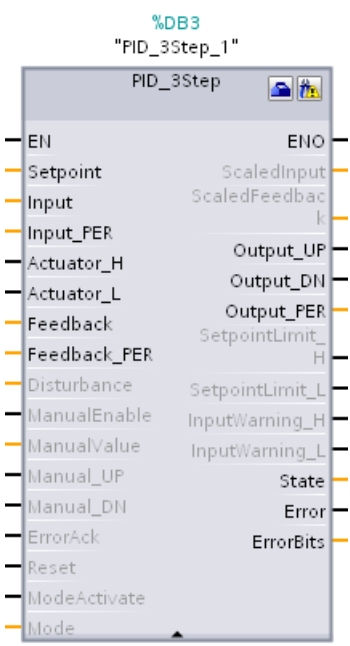
KOP/FUP	SCL	Beschreibung
	<pre>"PID_Compact_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, Disturbance:=_real_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, ErrorAck:=_bool_in_, Reset:=_bool_in_, ModeActivate:=_bool_in_, Mode:=_int_in_, ScaledInput=>_real_out_, Output=>_real_out_, 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_);</pre>	<p>PID_Compact stellt einen PID-Regler mit Selbsteinstellung für den Automatik- und Handbetrieb bereit. PID_Compact ist ein PIDT1-Regler mit Anti-Windup und Gewichtung des P- und D-Anteils.</p>

- ¹ STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.
- ² Im SCL-Beispiel ist "PID_Compact_1" der Name des Instanz-DBs.

10.3.2.3 Anweisung PID_3Step

Die Anweisung PID_3Step konfiguriert einen PID-Regler mit Selbsteinstellungsfunktionen, der für motorbetätigte Ventile und Stellglieder optimiert wurde.

Tabelle 10-29 Anweisung PID_3Step

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>PID_3Step konfiguriert einen PID-Regler mit Selbsteinstellungsfunktionen, der für motorbetätigte Ventile und Stellglieder optimiert wurde. Er bietet zwei Boolesche Ausgänge.</p> <p>PID_3Step ist ein PID T1-Regler mit Anti-Windup und Gewichtung des P- und D-Anteils.</p>

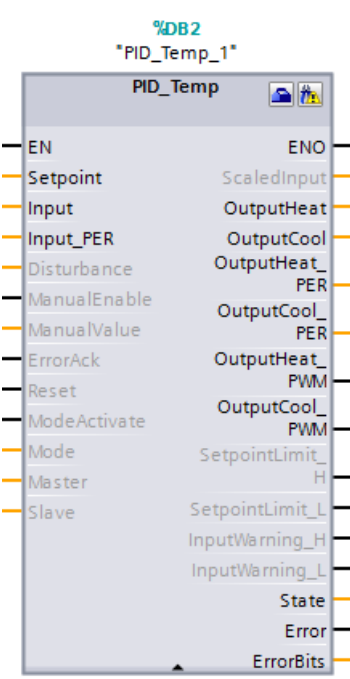
¹ STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.

² Im SCL-Beispiel ist "PID_3Step_1" der Name des Instanz-DBs.

10.3.2.4 Anweisung PID_Temp

Die Anweisung PID_Temp bietet einen universellen PID-Regler für die speziellen Anforderungen der Temperaturregelung.

Tabelle 10-30 Anweisung PID_Temp

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>PID_Temp bietet die folgenden Möglichkeiten:</p> <ul style="list-style-type: none"> • Heizen und Kühlen im Prozess mit verschiedenen Aktoren • Integrierte Selbsteinstellung für die Temperaturregelung • Kaskadenschaltung von mehreren Temperaturen im Prozess, die vom gleichen Aktor geregelt werden

¹ STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.

² Im SCL-Beispiel ist "PID_Temp_1" der Name des Instanz-DBs.

10.3.3 Weitere Informationen zum PID-Regler der S7-1200

Weitere Informationen zum PID-Regler der S7-1200 finden Sie unter Handbuch SIMATIC S7-1200, S7-1500 PID-Regelung (<https://support.industry.siemens.com/cs/us/en/view/108210036>).

In diesem Handbuch finden Sie ausführliche Informationen zu den folgenden Themen:

- Prinzipien für die Regelung
- Konfigurieren eines Software-Reglers

10.3 PID-Regelung

- Arbeiten mit PID_Compact:
 - Prozesswertgrenzen
 - ErrorBit-Parameter
 - Warnparameter
- Arbeiten mit PID_3Step:
 - ErrorBit-Parameter
 - Warnparameter
- Arbeiten mit PID_Temp:
 - ErrorBit-Parameter
 - Warnparameter
- Konfigurieren der Regler PID_Compact, PID_3Step und PID_Temp
- Inbetriebnehmen der Regler PID_Compact, PID_3Step und PID_Temp

Kommunikation

11.1 Überblick

Die S7-1200 bietet mehrere Arten der Kommunikation zwischen CPUs und Programmiergeräten, HMI-Geräten und anderen CPUs.

PROFINET

PROFINET wird für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern über Ethernet verwendet:

- Bei der S7-1200 unterstützt PROFINET 16 IO-Geräte mit maximal 256 Submodulen, und PROFIBUS gestattet 3 unabhängige PROFIBUS DP-Master und unterstützt 32 Slaves je DP-Master, mit maximal 512 Modulen je DP-Master.
- S7-Kommunikation
- User Datagram Protocol (UDP)
- ISO on TCP (RFC 1006)
- Transport Control Protocol (TCP)

PROFINET IO-Controller

Als IO-Steuerung über PROFINET IO kommuniziert die CPU mit bis zu 16 PN-Geräten im lokalen PN-Netzwerk oder über einen PN/PN-Koppler (Verbund). Ausführliche Informationen hierzu finden Sie unter PROFIBUS and PROFINET International, PI (www.profinet.com).

PROFIBUS

PROFIBUS wird für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern über das PROFIBUS-Netzwerk verwendet:

- Beim CM 1242-5 fungiert die CPU als PROFIBUS DP-Slave.
- Beim CM 1243-5 fungiert die CPU als PROFIBUS DP-Master der Klasse 1.
- PROFIBUS DP-Slaves, PROFIBUS DP-Master und AS-i (die 3 Kommunikationsmodule auf der linken Seite) und PROFINET sind voneinander getrennte Kommunikationsnetzwerke, die sich nicht gegenseitig einschränken.

ASi

Der S7-1200 CM 1243-2 AS-i Master ermöglicht die Anbindung eines AS-i-Netzwerks an eine S7-1200 CPU.

CPU-zu-CPU-Kommunikation über S7

Sie können eine Kommunikationsverbindung mit einer Partnerstation aufbauen und mithilfe der Anweisungen GET und PUT mit S7-CPU's kommunizieren.

TeleService-Kommunikation

Bei TeleService über GPRS kommuniziert eine Engineering-Station, auf der STEP 7 installiert ist, über das GSM-Netz und das Internet mit einer SIMATIC S7-1200 Station mit einem CP 1242-7. Die Verbindung läuft über einen Telecontrol-Server, der als Vermittler dient und mit dem Internet verbunden ist.

IO-Link

Der S7-1200 SM 1278 4xIO-Link-Master ermöglicht die Verbindung von IO-Link-Geräten mit einer S7-1200 CPU.

Sicherheitsrisiken durch physische Netzwerkangriffe vermeiden



WARNUNG

Sicherheitsrisiken durch physische Netzwerkangriffe vermeiden

Wenn sich ein Angreifer physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.

Einige andere Arten der Kommunikation (E/A-Austausch über PROFIBUS, PROFINET, AS-i oder einen anderen E/A-Bus, GET/PUT, T-Block und Kommunikationsmodule (CM)) haben keine Sicherheitsfunktionen. Sie müssen diese Arten der Kommunikation durch Begrenzung des physischen Zugriffs schützen. Wenn sich ein Angreifer über diese Arten der Kommunikation physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.

Wenn Sie diese Arten der Kommunikation nicht schützen, kann das zu Tod oder schwerem Personenschaden führen.

Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren "Operational Guidelines für Industrial Security" (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) auf der Website "Service & Support" von Siemens.

Siehe auch

Secure Communication und Legacy-Kommunikation (Seite 591)

11.2 Secure Communication und Legacy-Kommunikation

Bei der S7-1200 CPU V4.5 ist Secure Communication zwischen PLCs und dem TIA Portal, SIMATIC Automation Tool und HMI-Geräten implementiert. Diese Implementierung verwendet den Industriestandardprotokoll TLS 1.3 (Transport Layer Security). Bei der S7-1200 CPU gibt es seit V4.0 Erweiterungen für Secure Communication. Für die Kompatibilität mit älteren Geräten und Clients gibt es weiterhin die Legacy-Kommunikation. Sie haben die Wahl, ob Sie Secure Communication oder die Legacy-Kommunikation verwenden. Siemens empfiehlt unbedingt die Verwendung von Secure Communication für Clients und Geräte, die diese Kommunikation unterstützen.

Vorteile von Secure Communication

Mit Secure Communication erreichen Sie folgende Ziele:

- **Geheimhaltung:** Die Daten sind geheim und können nicht von Unbefugten gelesen werden.
- **Integrität:** Die Nachricht, die den Empfänger erreicht, ist die gleiche Nachricht, unverändert, die der Absender gesendet hat. Die Nachricht wurde in keiner Weise modifiziert.
- **Authentifizierung am Endpunkt:** Der Kommunikationspartner am Endpunkt ist genau derjenige, der er vorgibt zu sein, und derjenige, der erreicht werden soll. Die Identität des Partners ist verifiziert.

Heute sind vernetzte industrielle Maschinen und Leitsysteme mit sensiblen Daten einem hohen Risiko ausgesetzt und stellen daher hohe Sicherheitsanforderungen an den Datenaustausch.

Schutz über eine Firewall, VPN-Verbindung oder ein Security-Modul war in der Vergangenheit üblich und bleibt üblich. Zusätzlich zur physischen Sicherheit wird es jedoch erforderlich, Daten mittels Secure Communication in verschlüsselter Form an externe Kommunikationspartner zu übertragen.

Die CPU verwendet X.509-Zertifikate, um Secure Communication zwischen der CPU und Clients zu bieten. Clients wie STEP 7 und das SIMATIC Automation Tool können voraussetzen, dass Sie dem Zertifikat in der CPU vertrauen. Gehen Sie beim Herunterladen eines Zertifikats in eine CPU vorsichtig vor, damit Sie dem Zertifikat vertrauen können, wenn Sie dazu aufgefordert werden.

Hinweise zur Nutzung von Secure Communication

Beachten Sie im Hinblick auf Secure Communication folgende Hinweise:

- Die Kommunikationsgeschwindigkeiten können bei Secure Communication im Vergleich zur Legacy-Kommunikation langsamer sein.
- Für die Secure PG/PC- und HMI-Kommunikation sind TIA Portal V17 und S7-1200 CPUs ab V4.5 erforderlich.

Die Fähigkeiten für Secure Communication sind von mehreren Faktoren abhängig:

- Tatsächliche Firmware-Version der physikalisch CPU
- Konfigurierte Firmware-Version der CPU im STEP 7-Projekt

11.2 Secure Communication und Legacy-Kommunikation

Die folgende Tabelle zeigt die unterstützte Kommunikation für eine S7-1200 CPU mit Firmware V4.5:

Gerätekonfiguration der CPU im STEP 7-Projekt	Client-Versionen			
	TIA Portal V17 / SIMATIC Automation Tool V4.0 SP3	TIA Portal < V17 / SIMATIC Automation Tool < V4.0 SP3	HMI V17 ¹	HMI < V17
Im STEP 7-Projekt konfigurierte Firmware-Version: V4.5 Konfigurationseinstellung: "Nur Secure PG/PC- und HMI-Kommunikation zulassen" ausgewählt	Sicher	Keine Verbindung	Sicher	Keine Verbindung
Im STEP 7-Projekt konfigurierte Firmware-Version: V4.5 Konfigurationseinstellung: "Nur Secure PG/PC- und HMI-Kommunikation zulassen" abgewählt	Standard ist Secure, Legacy ist konfigurierbar	Legacy	Sicher	Legacy
Im STEP 7-Projekt konfigurierte Firmware-Version: V4.4	Legacy			
Kein Projekt	Sicher oder Legacy	Legacy	-/-	

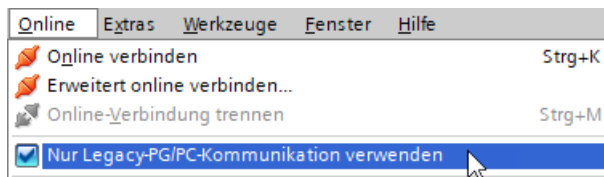
¹ Nicht zutreffend für HMI-Geräte mit S7-Kommunikation

Sie müssen für die Firmware-Version der CPU im STEP 7-Projekt V4.5 konfigurieren, um TLS 1.3 nutzen zu können. Die CPU verwendet TLS 1.2, wenn die konfigurierte Firmware-Version im Projekt vor V4.5 ist.

Secure Communication oder Legacy-Kommunikation konfigurieren

Sie konfigurieren die Sicherheitsfunktionen Ihrer CPU in der Gerätekonfiguration (Seite 160) der CPU im TIA Portal unter Schutz & Security. Die Dialoge im TIA Portal bieten Zugriff auf das Informationssystem im TIA Portal, um Sie durch die Konfigurierung zu führen. Zur Vereinfachung der Konfigurierung bietet TIA Portal ab V17 einen Security-Assistent (Seite 160), der Sie durch die Security-Optionen führt.

Ab TIA Portal V17 wird als Standardwert Secure PG/PC- und HMI-Kommunikation genutzt. Aus Inbetriebnahmegründen können Sie jedoch das TIA Portal zwingen, Legacy-PG/PC-Kommunikation zu verwenden, indem Sie im Online-Menü "Nur Legacy-PG/PC-Kommunikation verwenden" auswählen.



Weitere Informationen

Weitere Einzelheiten über die Implementierung von Secure Communication finden Sie im Informationssystem im TIA Portal. Weiterführende Informationen über Zertifikate finden Sie insbesondere in den folgenden Themen im TIA Portal-Informationssystem:

- Vertraulichkeit durch Verschlüsselung
- Verwalten von Zertifikaten mit STEP 7
- Beispiele für die Verwaltung von Zertifikaten
- Authentizität und Integrität durch Signaturen

Siehe auch

Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161)

Sichere PG/PC-Kommunikation und HMI-Kommunikation aktivieren und Zertifikate erstellen (Seite 166)

Eine CPU mit Schutz vertraulicher PLC-Konfigurationsdaten ersetzen (Seite 1455)

11.3 Kommunikationsprotokolle und Ports für die Ethernet-Kommunikation

Hierbei handelt es sich um eine Übersicht der für die Kommunikation über PN/IE-Schnittstellen unterstützten Protokolle und Ports. Bei den angegebenen Ports handelt es sich um die von der S7-1200 PLC verwendeten Standard-Portnummern. Zahlreiche Kommunikationsprotokolle und Implementierungen ermöglichen es Ihnen, andere Portnummern zu verwenden. Die folgenden Tabellen zeigen verschiedene von der S7-1200 PLC verwendete Schichten, Protokolle und Ports.

Tabelle 11-1 Ports und Protokolle für die Transportschicht der S7-1200

Port(s)	Richtung	Protokoll	Anwendung	Beschreibung
25	Ausgehend	TCP	SMTP	SMTP wird zum Versenden von E-Mails verwendet.
80	Eingehend	TCP	HTTP	HTTP wird für die Kommunikation mit dem CPU-internen Webserver verwendet.
102 ¹	Eingehend/Ausgehend	TCP	ISO-on-TCP	ISO-on-TCP (nach RFC 1006) wird für meldungsorientierten Datenaustausch mit Remote-CPU, S7-Kommunikation mit ES, HMI verwendet.
123	Ausgehend	UDP	NTP	NTP wird für die Synchronisation der CPU-Systemzeit mit der Uhrzeit eines NTP-Servers verwendet.
161 ¹	Eingehend	UDP	SNMP	SNMP wird zum Lesen und Einstellen von Netzwerkverwaltungsdaten (über SNMP-verwaltete Objekte) durch den SNMP-Manager verwendet.
443	Eingehend	TCP	HTTPS	HTTP wird für die Kommunikation mit dem CPU-internen Webserver über TLS verwendet.
465, 587	Ausgehend	TCP	SMTPS	SMTP wird zum Versenden von E-Mails über sichere Verbindungen verwendet.
502	Eingehend/Ausgehend	TCP	Modbus	Modbus/TCP wird von MB_CLIENT/MB_SERVER-Anweisungen im Anwenderprogramm verwendet.

11.3 Kommunikationsprotokolle und Ports für die Ethernet-Kommunikation

Port(s)	Richtung	Protokoll	Anwendung	Beschreibung
4840 ²	Eingehend	TCP	OPC UA	Kommunikationsstandard von der Unternehmens- zur Feldebene.
34964 ¹	Eingehend/Ausgehend	UDP	PROFINET Context Manager	Der PROFINET Context Manager bietet eine Endpunktzuordnung, um eine Application Relation herzustellen (PROFINET AR).

- ¹ Diese Ports sind in der im Auslieferungszustand sofort einsetzbaren Konfiguration mit konfigurierten IP-Adressen offen und zugänglich. Andere Anwendungen müssen als Teil des S7-1200 Anwenderprogramms aktiviert/konfiguriert werden.
- ² Port 4840 ist der Standardport, er kann jedoch auch konfiguriert werden.

Tabelle 11-2 Portbereiche, die von offener Benutzerkommunikation (Open User Communication, OUC) und anderen Protokollen verwendet werden können. Als Teil des S7-1200 Anwenderprogramms werden vom Anwender exakte Kommunikationsparameter definiert.

Portbereich	Richtung	Protokoll	Anwendung	Beschreibung
1 bis 999	Variiert	TCP/UDP	OUC	Portbereich kann mit Ausnahme bereits verwendeter Ports in eingeschränktem Umfang verwendet werden.
2000-5000	Variiert	TCP/UDP	OUC	Empfohlener Portbereich für OUC
5001-49151	Variiert	TCP/UDP	OUC	Portbereich kann mit Ausnahme bereits verwendeter Ports in eingeschränktem Umfang verwendet werden.
49152-65535	Ausgehend	TCP/UDP	Variiert	Dynamischer Portbereich für den aktiven Verbindungsendpunkt, wenn die Anwendung die lokale Portnummer nicht bestimmt.

Tabelle 11-3 Von S7-1200 in der Sicherungs- und Netzwerkschicht (Schichten 2 und 3) des OSI-Modells verwendete Protokolle

Protokoll	Richtung	Ethertyp	Beschreibung
PROFINET DCP	Eingehend/Ausgehend	0x8892	DCP wird von PROFINET verwendet, um PROFINET-Geräte zu erkennen und Grundeinstellungen bereitzustellen.
LLDP	Ausgehend	0x88CC	LLDP wird von PROFINET verwendet, um Beziehungen zwischen benachbarten PROFINET-Geräten zu erkennen und zu verwalten. LLDP verwendet die besondere Multicast-MAX-Adresse: 01-80-C2-00-00-0E.
PROFINET IO	Eingehend/Ausgehend	0x8892	Die PROFINET IO-Telegramme dienen dazu, IO-Daten zyklisch zwischen PROFINET IO-Controller und IO-Devices via Ethernet zu übertragen.
ICMP	Eingehend	0x800	Das Internet Control Message Protocol wird für Diagnose- oder Steuerungszwecke verwendet.

11.4 Asynchrone Kommunikationsverbindungen

Überblick über die Kommunikationsdienste

Die CPU unterstützt die folgenden Kommunikationsdienste:

Kommunikationsdienst	Funktionalität	Bei PROFIBUS DP		Bei Ethernet
		CM 1243-5 DP-Master-modul	CM 1242-5 DP-Slave-modul	
PG-Kommunikation	Inbetriebnahme, Test, Diagnose	Ja	Nein	Ja
HMI-Kommunikation	Bedienen und Beobachten	Ja	Nein	Ja
S7-Kommunikation	Datenaustausch über konfigurierte Verbindungen	Ja	Nein	Ja
Routing von PG-Funktionen	Beispielsweise Test und Diagnose außerhalb der Netzwerkgrenzen	Nein	Nein	Nein
PROFIBUS DP	Datenaustausch zwischen Master und Slave	Ja	Ja	Nein
PROFINET IO	Datenaustausch zwischen E/A-Steuerungen und E/A-Geräten	Nein	Nein	Ja
Webserver	Diagnose	Nein	Nein	Ja
SNMP ¹ (Simple Network Management Protocol)	Standardprotokoll für Netzwerkdagnose und Parametrierung	Nein	Nein	Ja
S7-Routing	Mit Hilfe von Routingtabellen können Kommunikationspartner mit jedem Gerät kommunizieren, selbst wenn sich die Geräte in unterschiedlichen S7-Subnetzen befinden.	Nein	Nein	Ja
Offene Kommunikation über TCP/IP	Datenaustausch über Industrial Ethernet mit TCP/IP-Protokoll (mit ladbaren FBs)	Nein	Nein	Ja
Offene Kommunikation über ISO-on-TCP	Datenaustausch über Industrial Ethernet mit ISO-on-TCP-Protokoll (mit ladbaren FBs)	Nein	Nein	Ja
Offene Kommunikation über UDP	Datenaustausch über Industrial Ethernet mit UDP-Protokoll (mit ladbaren FBs)	Nein	Nein	Ja
OPC UA-Server	Datenaustausch über Industrial Ethernet mit OPC UA Clients	Nein	Nein	Ja ²

¹ Die CPU unterstützt SNMP V1 ohne TRAPs.

² OPC UA wird nur über den integrierten Ethernet-Port der CPU unterstützt.


Verfügbare Verbindungen

Die CPU unterstützt maximal die folgende Anzahl von gleichzeitigen, asynchronen Kommunikationsverbindungen für PROFINET und PROFIBUS. Die maximale Anzahl der jeder Kategorie zugeordneten Verbindungsressourcen ist festgelegt. Sie können die Werte nicht ändern. Die frei verfügbaren Verbindungen können jedoch genutzt werden, um die Anzahl der Verbindungen in jeder Kategorie nach dem Bedarf Ihrer Anwendung zu erhöhen.

Einige Verbindungstypen haben eine feste Anzahl reservierter Ressourcen (auch "garantiert" genannt). Das bedeutet, dass die CPU garantiert maximal diese Anzahl reservierter Ressourcen für den Verbindungstyp unterstützt. So können beispielsweise mindestens 12 HMI-Verbindungen gleichzeitig mit der CPU hergestellt werden. Weitere Verbindungen über die Anzahl reservierter Ressourcen hinaus können für einen Verbindungstyp hergestellt werden, doch diese Verbindungsressourcen müssen aus dem Pool "dynamischer" Ressourcen stammen.

Die dynamischen Ressourcen (auch "frei" genannt) sind eine Sammlung von Ressourcen, die für jeden Verbindungstyp genutzt werden können. Diese Ressourcen werden von Verbindungen verwendet, die keine reservierten Ressourcen haben (wie OPC UA), oder von Verbindungen, die alle ihre reservierten Ressourcen aufgebraucht haben.

Ab V17 verfügen CPUs V4.5 über 34 dynamischen Ressourcen.

Verbindungsressourcen				
	Ressourcen der Station			Ressourcen des Moduls
	Reserviert		Dynami... 	PLC_1 [CPU 1214C DC/DC]
Maximale Anzahl der Ressourcen:	34		34	68
	Maximum	Konfigurierte	Konfigurierte	Konfigurierte
PG-Kommunikation:	4	-	-	-
HMI-Kommunikation:	12	0	0	0
S7-Kommunikation:	8	0	0	0
Open User Communication:	8	0	0	0
Web-Kommunikation:	2	-	-	-
OPC UA-Client/Server-Kommunikati...	0	-	-	-
Sonstige Kommunikation:	-	-	0	0
Insgesamt verwendete Ressourcen:	0		0	0
Verfügbare Ressourcen:	34		34	68

Hinweis

Wenn Sie CM/CP-Module ergänzen, erhöht sich die Gesamtzahl der S7-1200 Kommunikationsverbindungen nicht.

Hinweis

OPC UA-Verbindungen

OPC UA-Verbindungen verbrauchen Ressourcen aus den dynamischen Ressourcen. Stellen Sie sicher, dass Sie ausreichend Verbindungen für Ihre Anwendung verfügbar haben.

Basierend auf den zugeordneten Verbindungsressourcen steht die folgende Anzahl von Verbindungen je Gerät zur Verfügung:

Typ	Reserviert	Maximum ¹
Kommunikation über Programmiergerät (PG)	4	4
Kommunikation über Mensch-Maschine-Schnittstelle (HMI)	12	18
S7-Kommunikation	8	14
Open User Communication	8	14
Webkommunikation	2	30
OPC UA Client/Server-Kommunikation	0	10
Sonstige Kommunikation	0	-

¹ Da die dynamischen Verbindungen gemeinsam genutzt werden, ist es nicht möglich, alle Verbindungen gleichzeitig zu verwenden.

Beispiel: Die CPU hat vier verfügbare PG Verbindungsressourcen. Je nach aktuell verwendeten PG-Funktionen verwendet das PG möglicherweise tatsächlich eine, zwei, drei oder vier der verfügbaren Verbindungsressourcen. Sie können immer ein PG verwenden.

Ein weiteres Beispiel ist die Anzahl der HMI-Geräte, wie in der nachstehenden Abbildung gezeigt. HMI-Geräte haben 12 verfügbare Verbindungsressourcen. Je nach Typ oder Modell Ihres HMI-Geräts und den verwendeten HMI-Funktionen kann jedes HMI-Gerät tatsächlich eine, zwei oder drei seiner verfügbaren Verbindungsressourcen nutzen. Abhängig von der Anzahl der verwendeten verfügbaren Verbindungsressourcen können möglicherweise mehr als vier HMI-Geräte gleichzeitig verwendet werden. Es besteht jedoch jederzeit die Garantie für mindestens vier HMI-Geräte. Ein HMI-Gerät kann seine verfügbaren Verbindungsressourcen (je 1 von insgesamt 3) für die folgenden Funktionen nutzen:

- Lesen
- Schreiben
- Alarmer und Diagnosen

Dies ist nur ein Beispiel. Die tatsächliche Anzahl der Verbindungen kann je nach Typ und Version des HMI-Geräts unterschiedlich sein.

Beispiel	HMI 1	HMI 2	HMI 3	HMI 4	HMI 5	Gesamtzahl verfügbarer Verbindungsressourcen
Verwendete Verbindungsressourcen	2	2	2	3	3	12

Hinweis

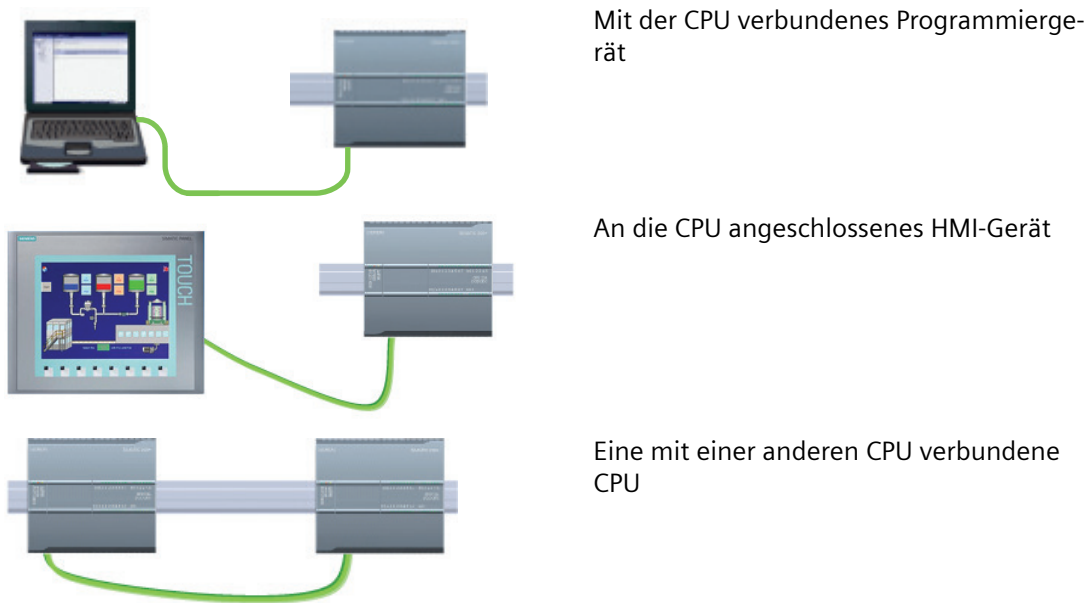
Webserver-Verbindungen: Die CPU bietet Verbindungen für mehrere Webbrowser. Die Anzahl von Browsern, die von der CPU gleichzeitig unterstützt werden können, richtet sich danach, wie viele Verbindungen ein Webbrowser anfordert/verwendet und wie viele dynamische Verbindungen in der CPU verfügbar sind.

Hinweis

Die Kommunikationsverbindungen über Open User Communication, S7-Verbindung, HMI-Gerät, Programmiergerät, OPC UA und Webserver können basierend auf den verwendeten Funktionen mehrere Verbindungsressourcen nutzen.

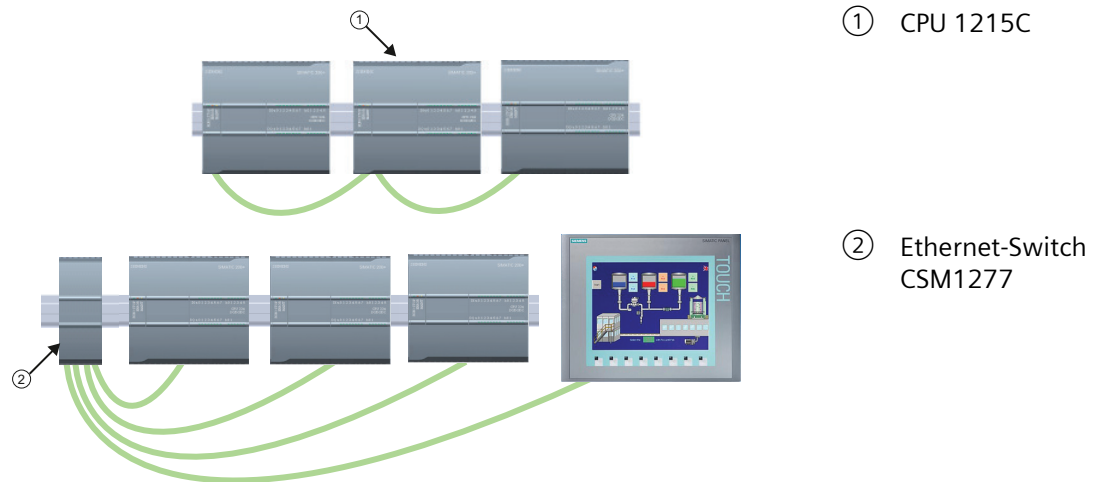
11.5 PROFINET

Die CPU kann mit anderen CPUs, mit Programmiergeräten, mit HMI-Geräten und Siemens-fremden Geräten über herkömmliche TCP-Kommunikationsprotokolle kommunizieren.



Ethernet-Switch

Die CPU 1211C, 1212C und 1214C haben einen Ethernet-Port und verfügen nicht über einen integrierten Ethernet-Switch. Eine direkte Verbindung zwischen einem Programmiergerät oder HMI-Gerät und einer CPU erfordert keinen Ethernet-Switch. Ein Netzwerk mit mehr als zwei CPUs oder HMI-Geräten jedoch benötigt einen Ethernet-Switch.



Die CPUs 1215C und 1217C besitzen einen integrierten Ethernet-Switch mit 2 Ports. Sie können ein Netzwerk mit einer CPU 1215C und zwei anderen S7-1200 CPUs aufbauen. Zum Anschließen mehrerer CPUs und HMI-Geräte können Sie auch einen auf dem Baugruppenträger montierten Ethernet-Switch CSM1277 mit 4 Ports verwenden.

11.5.1 Netzwerkverbindung erstellen

In der Netzsicht der Gerätekonfiguration können Sie die Netzwerkverbindungen zwischen den Geräten in Ihrem Projekt herstellen. Nach dem Herstellen der Netzwerkverbindung können Sie im Register "Eigenschaften" des Inspektorfensters die Netzwerkparameter konfigurieren.

Tabelle 11-4 Netzwerkverbindung erstellen

Handlungsanweisung	Ergebnis
<p>Wählen Sie die "Netzwerkansicht" für die Anzeige der Geräte, die verbunden werden sollen.</p>	
<p>Wählen Sie die Schnittstelle eines Geräts und ziehen Sie die Verbindung zur Schnittstelle des zweiten Geräts.</p>	
<p>Lassen Sie die Maustaste los, damit die Netzwerkverbindung erstellt wird.</p>	

11.5.2 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

Eine Verbindung zwischen lokaler CPU und entfernter Partner-CPU definiert eine logische Zuweisung von zwei Kommunikationspartnern für die Herstellung von Kommunikationsdiensten. Eine Verbindung definiert folgende Elemente:

- Beteiligte Kommunikationspartner (einer aktiv, einer passiv)
- Art der Verbindung (z. B. eine PLC-, HMI- oder Geräteverbindung)
- Verbindungspfad

Die Kommunikationspartner führen die Anweisungen zum Einrichten und Herstellen der Kommunikationsverbindung aus. Mit Parametern geben Sie die aktiven und passiven

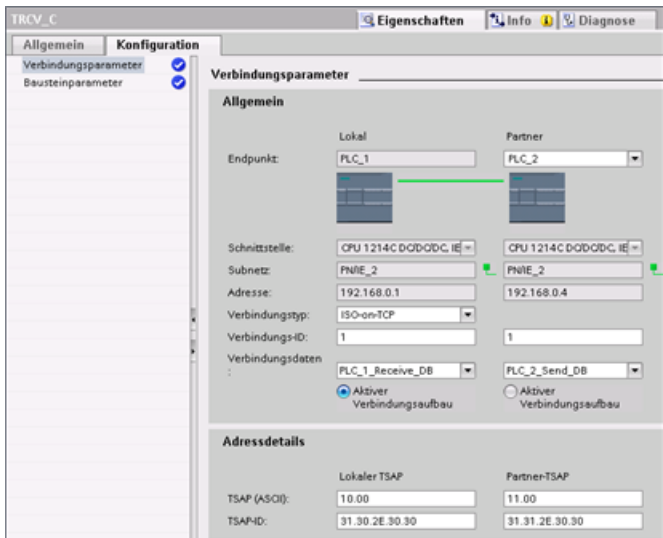
Endpunkte der Kommunikation an. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.

Wird die Verbindung beendet (z. B. durch eine Leitungsstörung), so versucht der aktive Partner, sie wieder aufzubauen. Sie müssen die Kommunikationsanweisung nicht erneut ausführen.

Verbindungspfade

Nach dem Einfügen einer Anweisung TSEND_C, TRCV_C oder TCON in das Anwenderprogramm werden im Inspektorfenster die Eigenschaften der Verbindung angezeigt, wenn Sie einen Teil der Anweisung auswählen. Sie legen die Kommunikationsparameter über das Register "Konfiguration" im Dialog "Eigenschaften" der Kommunikationsanweisung fest.

Tabelle 11-5 Verbindungspfad konfigurieren (über die Eigenschaften der Anweisung)

TCP, ISO-on-TCP und UDP	Verbindungseigenschaften
<p>Bei den TCP-, ISO-on-TCP- und UDP-Ethernet-Protokollen konfigurieren Sie die Verbindungen der lokalen und der Partner-CPU in den "Eigenschaften" der Anweisung (TSEND_C, TRCV_C oder TCON).</p> <p>Die Abbildung zeigt die "Verbindungseigenschaften" im Register "Verbindung" einer ISO-on-TCP-Verbindung.</p>	

Hinweis

Wenn Sie die Verbindungseigenschaften für eine CPU konfigurieren, können Sie in STEP 7 entweder einen bestimmten Verbindungs-DB in der Partner-CPU auswählen (sofern vorhanden) oder einen Verbindungs-DB für die Partner-CPU anlegen. Die Partner-CPU muss im Projekt bereits angelegt sein, es darf sich nicht um eine "nicht spezifizierte" CPU handeln.

Sie müssen trotzdem eine Anweisung TSEND_C, TRCV_C oder TCON in das Anwenderprogramm der Partner-CPU einfügen. Wenn Sie die Anweisung einfügen, wählen Sie den Verbindungs-DB aus, der von der Konfiguration angelegt wurde.

Tabelle 11-6 Verbindungspfad für die S7-Kommunikation konfigurieren (Gerätekonfiguration)

S7-Kommunikation (GET und PUT)	Verbindungseigenschaften
<p>Bei der S7-Kommunikation konfigurieren Sie die Verbindungen zwischen lokaler und Partner-CPU im Editor "Geräte & Netze" des Netzwerks. Sie können auf die Schaltfläche "Hervorgehoben: Verbindung" klicken, um die "Eigenschaften" aufzurufen. Im Register "Allgemein" werden mehrere Eigenschaften aufgeführt:</p> <ul style="list-style-type: none"> • "Allgemein" (abgebildet) • "Lokale ID" • "Spezielle Verbindungseigenschaften" • "Adressdetails" (abgebildet) 	

Im Abschnitt "PROFINET" unter "Protokolle" (Seite 617) und im Abschnitt "S7-Kommunikation" unter "S7-Verbindung erstellen" (Seite 803) finden Sie weitere Informationen sowie eine Liste der verfügbaren Kommunikationsanweisungen.

Tabelle 11-7 Parameter für die CPU-Verbindung

Parameter		Definition
Adresse		Zugewiesene IP-Adressen
Allgemein	Endpunkt	Name der Partner-CPU (Empfänger)
	Schnittstelle	Name der Schnittstellen
	Subnetz	Name der Subnetze
	Schnittstellentyp	Nur S7-Kommunikation: Typ der Schnittstelle
	Verbindungstyp	Typ des Ethernet-Protokolls
	Verbindungs-ID	ID-Nummer
	Verbindungsdaten	Datenspeicher für die lokale CPU und die Partner-CPU
	Aktive Verbindung herstellen	Optionsfeld zum Auswählen der lokalen CPU oder der Partner-CPU als aktive Verbindung

Parameter		Definition
Adressdetails	Endpunkt	Nur S7-Kommunikation: Name der Partner-CPU (Empfänger)
	Baugruppenträger/Steckplatz	Nur S7-Kommunikation: Baugruppenträger- und Steckplatzadresse
	Verbindungsressource	Nur S7-Kommunikation: Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU
	Port (dezimal):	TCP und UDP: Port der Partner-CPU im Dezimalformat
	TSAP ¹ und Subnetz-ID:	ISO on TCP (RFC 1006) und S7-Kommunikation: TSAPs der lokalen CPU und der Partner-CPU im ASCII- und Hexadezimalformat

¹ Verwenden Sie beim Konfigurieren einer Verbindung mit einer S7-1200 CPU über ISO-on-TCP in der TSAP-Erweiterung für die passiven Kommunikationsteilnehmer nur ASCII-Zeichen.

Transport Service Access Points (TSAPs)

Mit TSAPs gestatten das ISO-on-TCP-Protokoll und die S7-Kommunikation mehrere Verbindungen mit einer einzigen IP-Adresse. TSAPs ermitteln die eindeutige Zuordnung dieser Verbindungen der Kommunikationsendpunkte zu einer IP-Adresse.

Die zu verwendenden TSAPs definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der TSAP einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

Portnummern

Bei den TCP- und UDP-Protokollen müssen in der Konfiguration der Verbindungsparameter der lokalen (aktiven) CPU die dezentrale IP-Adresse und die Portnummer der (passiven) Partner-CPU angegeben werden.

Die zu verwendenden Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adressdetails". Der Port einer Verbindung in der CPU wird im Feld "Lokaler Port" eingegeben. Der Port für die Verbindung in der Partner-CPU wird im Feld "Partner-Port" eingegeben.

11.5.3 IP-Adressen zuweisen

11.5.3.1 IP-Adressen für Programmier- und Netzwerkgeräte zuweisen

Wenn Ihr Programmiergerät eine integrierte Adapterkarte nutzt, die an das LAN Ihrer Anlage angeschlossen ist, müssen sich das Programmiergerät und die CPU im gleichen Subnetz befinden. Sie weisen das Subnetz als Kombination aus IP-Adresse und Subnetzmaske für das Gerät zu. Wenn Sie Unterstützung benötigen, wenden Sie sich an Ihren Netzwerkadministrator vor Ort.

Die Netzwerk-ID sind die ersten drei Oktette der IP-Adresse, z.B. **211.154.184**.16. Sie legt fest, welches IP-Netzwerk Sie nutzen. Die Subnetzmaske hat normalerweise den Wert **255.255.255.0**. Da sich Ihr Computer jedoch in einem Anlagen-LAN befindet, kann die Subnetzmaske andere Werte annehmen (z.B. **255.255.254.0**), um eindeutige Subnetze

einzurichten. Die Subnetzmaske definiert in logischer UND-Verknüpfung mit der IP-Adresse des Geräts die Grenzen eines IP-Subnetzes.

Hinweis

Im Internet, wo Ihre Programmiergeräte, Netzwerkgeräte und IP-Router mit der Welt kommunizieren, müssen eindeutige IP-Adressen zugeordnet werden, um Konflikte mit anderen Netzwerknutzern zu vermeiden. Wenden Sie sich an die IT-Abteilung Ihrer Firma, deren Mitarbeiter die Netzwerke in Ihrem Unternehmen kennen, um sich IP-Adressen zuweisen zu lassen.



WARNUNG

Unbefugter Zugriff auf die CPU über den Webserver

Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Ein unbefugter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Schützen Sie die CPU-Zugriffsstufen (Seite 163) und Webserver-Benutzer-IDs (Seite 852) durch starke Passwörter.
- Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden:
 - Großbuchstaben
 - Kleinbuchstaben
 - Ziffern
 - Sonderzeichen
- Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234.
- Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d
- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht.
- Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil die Benutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.
- Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN).

Hinweis

Eine zweite Netzwerkkartenkarte ist nützlich, wenn Sie Ihre CPU nicht an das Firmen-LAN anschließen möchten. Dieser Aufbau ist besonders während anfänglicher Tests bzw. während der Inbetriebnahmeprüfungen nützlich.

IP-Adresse Ihres Programmiergeräts über "Netzwerkumgebung" (auf Ihrem Desktop) zuweisen oder prüfen

Um die IP-Adresse Ihres Programmiergeräts zuzuweisen oder zu prüfen, gehen Sie wie folgt vor:

1. Öffnen Sie die Systemsteuerung über das Start-Menü.
2. Öffnen Sie das „Netzwerk- und Freigabecenter“ und wählen Sie "LAN-Verbindung" für den an Ihre CPU angeschlossenen Netzwerkkartenadapter
3. Klicken Sie auf "Eigenschaften" im Dialog "LAN-Verbindungsstatus".
4. Wählen Sie im Dialog "Eigenschaften von LAN-Verbindung" im Feld "Diese Verbindung verwendet folgende Elemente:" den Eintrag "Internetprotokoll 4 (TCP/IPv4)".
5. Klicken Sie auf "Eigenschaften".
6. Wählen Sie "IP-Adresse automatisch beziehen" oder "Folgende IP-Adresse verwenden", um eine statische IP-Adresse einzugeben, aus.
7. Wenn Sie "Folgende IP-Adresse verwenden" gewählt haben, stellen Sie die IP-Adresse und Subnetzmaske ein:
 - Stellen Sie die IP-Adresse so ein, dass die gleiche Netzwerk-ID und die gleiche Subnetzmaske wie die CPU verwendet werden. Beispiel: Wenn die IP-Adresse der CPU **192.168.0.1** ist, können Sie als IP-Adresse **192.168.0.200** einstellen.
 - Wählen Sie die Subnetzmaske **255.255.255.0**.
 - Lassen Sie das Standard-Gateway leer.

Sie können sich jetzt mit der CPU verbinden.

Hinweis

Die Netzwerkschnittstellenkarte und die CPU müssen sich im gleichen Subnetz befinden, damit STEP 7 die CPU finden und mit ihr kommunizieren kann.

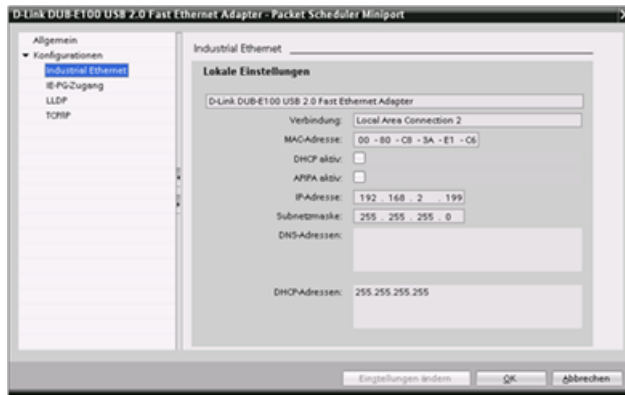
Wenden Sie sich an Ihre IT-Abteilung, wenn Sie bei der Einrichtung einer Netzwerkkonfiguration, die Ihnen eine Verbindung mit der S7-1200 CPU gestattet, Unterstützung benötigen.

11.5.3.2 Ermitteln der IP-Adresse Ihres Programmiergeräts

Die MAC- und die IP-Adresse Ihres Programmiergeräts ermitteln Sie mit den folgenden Menübefehlen:

1. Erweitern Sie in der Projektnavigation den Knoten "Online-Zugänge".
2. Klicken Sie mit der rechten Maustaste auf das erforderliche Netzwerk und wählen Sie "Eigenschaften".
3. Erweitern Sie im Dialog "Netzwerk" die "Konfigurationen" und wählen Sie "Industrial Ethernet".

Dort werden die MAC- und die IP-Adresse Ihres Programmiergeräts angezeigt.



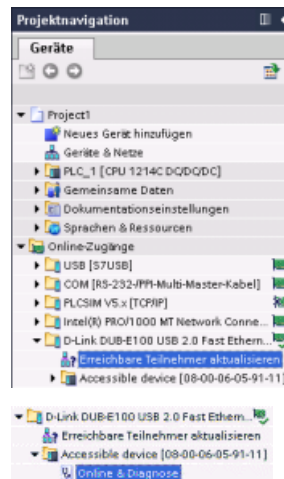
11.5.3.3 Online eine IP-Adresse zu einer CPU zuweisen

Sie können einem Netzwerkgerät online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich.

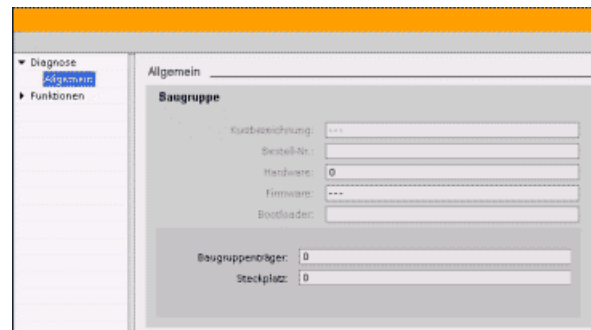
1. Prüfen Sie in der Projektnavigation, ob die CPU keine konfigurierte IP-Adresse besitzt. Erweitern Sie "Online-Zugriff" > <Adapterkarte für das Netzwerk, in dem sich das Gerät befindet, und doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren".

Wenn STEP 7 statt einer IP-Adresse eine MAC-Adresse anzeigt, wurde keine IP-Adresse zugewiesen.

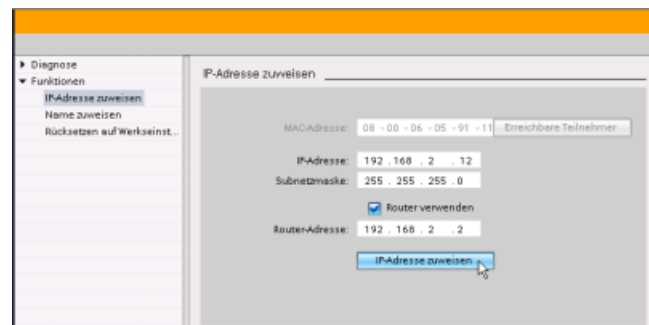
2. Doppelklicken Sie unter dem erforderlichen zugänglichen Gerät auf "Online & Diagnose".



3. Wählen Sie im Dialog "Online & Diagnose" Option "Funktionen" > "IP-Adresse zuweisen".

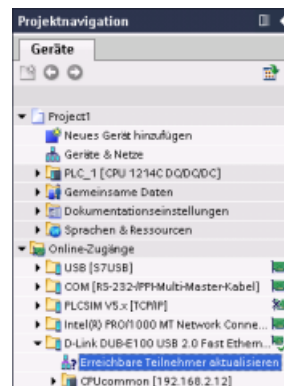


4. Geben Sie im Feld "IP-Adresse" Ihre neue IP-Adresse ein und klicken Sie auf die Schaltfläche "IP-Adresse zuweisen".



5. Prüfen Sie in der Projektnavigation, ob STEP 7 der CPU die neue IP-Adresse zugewiesen hat.

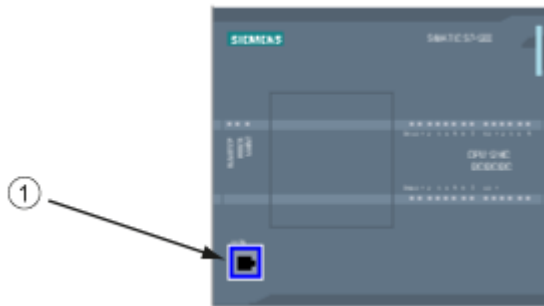
Doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren", um die zuvor konfigurierte IP-Adresse anzuzeigen.



11.5.3.4 IP-Adresse für eine CPU in Ihrem Projekt konfigurieren

Konfigurieren der PROFINET-Schnittstelle

Um Parameter für die PROFINET-Schnittstelle zu konfigurieren, wählen Sie das grüne PROFINET-Feld auf der CPU. Im Inspektorfenster wird das Register "Eigenschaften" für den PROFINET-Port angezeigt.



① PROFINET-Port

Konfigurieren der IP-Adresse

Ethernet-Adresse (MAC-Adresse): In einem PROFINET-Netzwerk ist jedem Gerät für die Identifikation eine MAC-Adresse (Media-Access-Control-Adresse) vom Hersteller zugewiesen. Eine MAC-Adresse besteht aus sechs Gruppen zu je zwei Hexadezimalziffern mit Bindestrichen (-) oder Doppelpunkten (:) als Trennzeichen in der Reihenfolge der Übertragung (zum Beispiel 01-23-45-67-89-AB oder 01:23:45:67:89:AB).

IP-Adresse: Jedes Gerät benötigt außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern.

Jede IP-Adresse ist in vier 8-Bit-Segmente unterteilt und wird in einem Dezimalformat mit Punkttrennzeichen dargestellt (zum Beispiel 211.154.184.16). Der erste Teil der IP-Adresse ist die Netzwerk-ID (Kennung des aktiven Netzwerks), der zweite Teil der Adresse ist die Host-ID (für jedes Gerät im Netzwerk eindeutig). Eine IP-Adresse von 192.168.x.y ist eine Standardkennzeichnung, die als Teil eines privaten Netzwerks, das sich nicht im Internet befindet, erkannt wird.

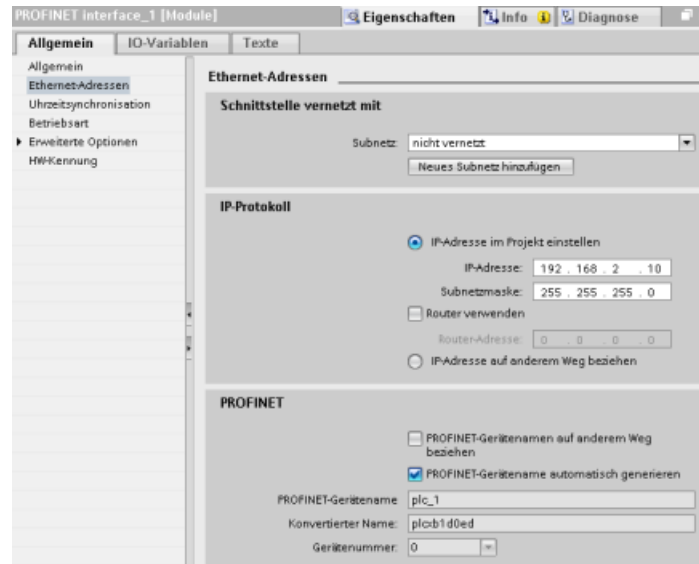
Subnetzmaske: Ein Subnetz ist eine logische Gruppierung miteinander verbundener Netzwerkgeräte. Die Teilnehmer eines Subnetzes befinden sich meist dicht nebeneinander auf einem LAN (Local Area Network). Eine Maske (die Subnetz- oder Netzmaske) legt die Grenzen eines IP-Subnetzes fest.

Eine Subnetzmaske von 255.255.255.0 eignet sich allgemein für ein kleines lokales Netzwerk. Das bedeutet, dass alle IP-Adressen in diesem Netzwerk dieselben ersten 3 Oktette haben müssen. Die einzelnen Geräte in diesem Netzwerk werden dann über das letzte Oktett (8-Bit-Feld) identifiziert. Ein Beispiel hierfür ist die Zuweisung einer Subnetzmaske von 255.255.255.0 und IP-Adressen von 192.168.2.0 bis 192.168.2.255 zu den einzelnen Geräten in einem kleinen lokalen Netzwerk.

Die einzige Verbindung zwischen verschiedenen Subnetzen läuft über einen Router. Werden Subnetze verwendet, so muss ein IP-Router eingesetzt werden.

IP-Router: Router sind die Verbindung zwischen LANs. Mit einem Router kann ein Rechner in einem LAN Meldungen an andere Netzwerke senden, die wiederum zu anderen LANs gehören. Liegt das Ziel der Daten nicht innerhalb des LANs, so leitet der Router die Daten an ein anderes Netzwerk oder eine Gruppen von Netzwerken weiter, wo die Daten ihrem Ziel zugestellt werden können.

Router benötigen IP-Adressen zum Senden und Empfangen von Datenpaketen.



Eigenschaften von IP-Adressen: Wählen Sie im Dialog "Eigenschaften" den Eintrag "Ethernet-Adressen". STEP 7 zeigt den Dialog für die Konfiguration der Ethernet-Adresse an, in dem Sie dem Softwareprojekt die IP-Adresse der CPU zuweisen, in die das Projekt geladen wird.

Tabelle 11-8 Parameter für die IP-Adresse

Parameter	Beschreibung	
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist. Um ein neues Subnetz anzulegen, klicken Sie auf Schaltfläche "Neues Subnetz hinzufügen". Die Voreinstellung ist "nicht verbunden". Zwei Arten von Verbindungen sind möglich: <ul style="list-style-type: none"> Die Voreinstellung "nicht verbunden" stellt eine lokale Verbindung her. Wenn Ihr Netzwerk über zwei oder mehr Geräte verfügt, ist ein Subnetz erforderlich. 	
IP-Protokoll	IP-Adresse	Zugewiesene IP-Adresse der CPU
	Subnetzmaske	Zugewiesene Subnetzmaske
	IP-Router verwenden	Aktivieren Sie das Optionskästchen, wenn ein IP-Router verwendet wird
	Router-Adresse	Zugewiesene IP-Adresse des Routers, sofern vorhanden

Hinweis

Alle IP-Adressen werden beim Laden des Projekts konfiguriert. Wenn die CPU keine vorkonfigurierte IP-Adresse besitzt, müssen Sie das Projekt mit der MAC-Adresse des Zielgeräts zuweisen. Ist Ihre CPU an einen Router oder ein Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.

Über das Optionsfeld "IP-Adresse auf anderem Weg einstellen" können Sie die IP-Adresse online oder über die Anweisung "T_CONFIG (Seite 717)" ändern, nachdem Sie das Programm in die CPU geladen haben. Diese Zuweisungsart der IP-Adresse gilt nur für die CPU.

 **WARNUNG****Laden einer Hardwarekonfiguration mit "IP-Adresse auf anderem Weg einstellen"**

Nach dem Laden einer Hardwarekonfiguration mit aktivierter Option "IP-Adresse auf anderem Weg einstellen" ist es nicht möglich, den Betriebszustand der CPU von RUN in STOP oder von STOP in RUN zu versetzen.

User equipment continues to run under these conditions and can result in unexpected machine or process operations, which could cause death, severe personal injury, or property damage if proper precautions are not taken.

Stellen Sie sicher, dass die IP-Adresse(n) Ihrer CPU festgelegt sind, bevor Sie die CPU in einer echten Automatisierungsumgebung einsetzen. Für diese Überprüfung können Sie das Programmierpaket STEP 7, das SIMATIC Automation Tool oder ein angeschlossenes HMI-Gerät in Verbindung mit der Anweisung T_CONFIG nutzen.

 **WARNUNG****Bedingung, unter der das PROFINET-Netzwerk möglicherweise stoppt**

Wenn Sie die IP-Adresse einer CPU online oder über das Anwenderprogramm ändern, kann dies dazu führen, dass das PROFINET-Netzwerk stoppt.

Wird die IP-Adresse einer CPU durch eine IP-Adresse außerhalb des Subnetzes ersetzt, verliert das PROFINET-Netzwerk die Kommunikation und der gesamte Datenaustausch stoppt. Die Anwendergeräte können so konfiguriert werden, dass sie unter diesen Bedingungen weiterarbeiten. Loss of PROFINET communication may result in unexpected machine or process operations, causing death, severe personal injury, or property damage if proper precautions are not taken.

Wenn eine IP-Adresse manuell geändert werden muss, stellen Sie sicher, dass sich die neue IP-Adresse innerhalb des Subnetzes befindet.

Konfigurieren des PROFINET-Ports

Standardmäßig konfiguriert die CPU die Ports der PROFINET-Schnittstelle für die Autonegotiation. Damit die Autonegotiation einwandfrei funktioniert, müssen Sie beide Stationen für die Autonegotiation einrichten. Wenn eine Station eine feste Konfiguration hat (z. B. Vollduplex bei 100 MBit/s) und die andere Station für die Autonegotiation eingerichtet ist, schlägt die Autonegotiation fehl, was zu Halbduplexbetrieb führt.

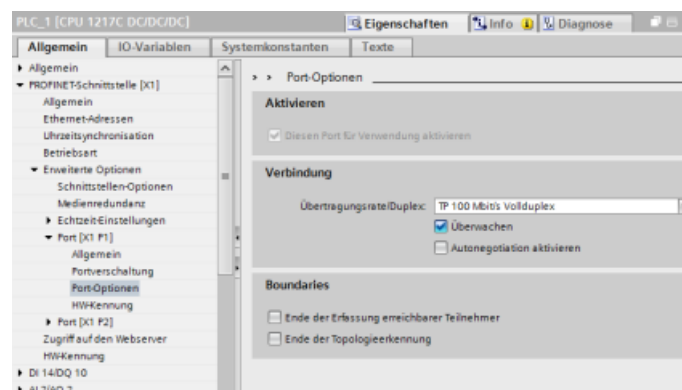
Um diese Einschränkung der Autonegotiation zu überwinden, bietet die S7-1200 eine Option zum Deaktivieren der Autonegotiation. Wenn Sie die Autonegotiation deaktivieren, ist die S7-1200 automatisch für den Vollduplexbetrieb bei 100 MBit/s konfiguriert.

Sie können die Übertragungsrate und Duplex für jeden Port auf einen festen Wert setzen:

1. Wählen Sie die erweiterten Optionen und den Port, den Sie konfigurieren möchten. Wählen Sie dann "Port-Optionen" aus.
2. Wählen Sie unter "Verbindung" im Feld "Übertragungsrate/Duplex" eine der folgenden Optionen aus:
 - Automatisch: Die CPU und das Partnergerät ermitteln Übertragungsrate und Duplex des Ports durch Autonegotiation.
 - TP 100 MBit/s Vollduplex: Wenn Sie die Autonegotiation deaktivieren, arbeitet der Port mit 100 MBit/s Vollduplex. Wenn Sie die Autonegotiation aktivieren, kann der Port mit 100 MBit/s Vollduplex oder einer anderen Übertragungsrate/Duplex arbeiten, die zwischen der CPU und dem Partnergerät durch Autonegotiation ermittelt wird (wenn Sie "Überwachen" auswählen, wird eine Meldung im Diagnosepuffer abgelegt (siehe unten)).
3. Überwachen: Wenn Sie dieses Kontrollkästchen aktivieren, wird eine Meldung im Diagnosepuffer abgelegt, sofern eine der folgenden Situationen am Port auftritt:
 - Am Port kann keine Verbindung hergestellt werden
 - Eine hergestellte Verbindung schlägt fehl
 - Sie wählen "TP 100 MBit/s Vollduplex" als Übertragungsrate/Duplex, und die CPU stellt mittels Autonegotiation eine Verbindung mit der ausgehandelten Übertragungsrate ungleich 100 MBit/s oder dem ausgehandelten Duplex ungleich Halbduplex her.
4. Autonegotiation aktivieren: Wenn Sie im Feld "Übertragungsrate/Duplex" die Option für Vollduplex bei 100 MBit/s auswählen, können Sie die Autonegotiation deaktivieren. Um die Autonegotiation zu deaktivieren, deaktivieren Sie das Kontrollkästchen "Autonegotiation aktivieren".

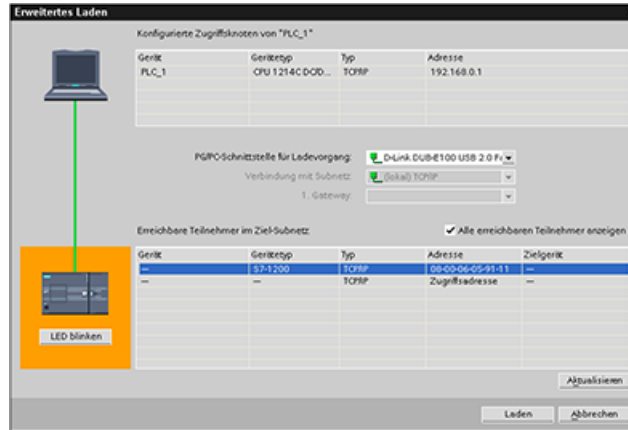
Hinweis

Wenn Sie die Autonegotiation nicht deaktivieren, ermitteln die CPU und das Partnergerät Übertragungsrate und Duplex des Ports durch Autonegotiation.



11.5.4 Testen des PROFINET-Netzwerks

Nachdem die Konfiguration beendet ist, laden Sie das Programm (Seite 199) in die CPU. Alle IP-Adressen werden beim Laden des Projekts konfiguriert.



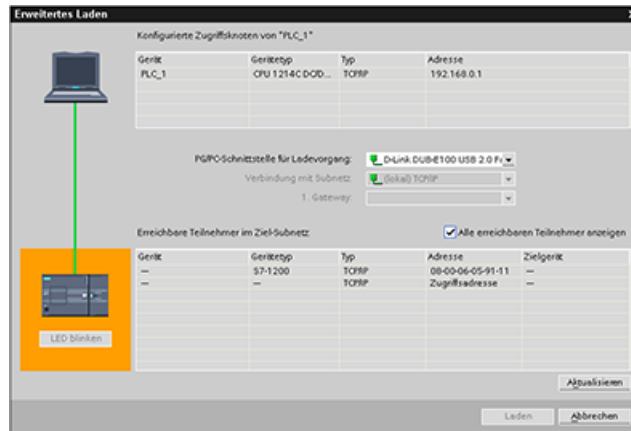
Online-Zuweisung einer IP-Adresse zu einem Gerät

Die S7-1200 CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher manuell eine IP-Adresse zuweisen.

- Wie Sie einem Gerät online eine IP-Adresse zuweisen, erfahren Sie unter "Gerätekonfiguration: IP-Adresse einer CPU online zuweisen" (Seite 606).
- Für die Zuweisung einer IP-Adresse in Ihrem Projekt muss die IP-Adresse in der Gerätekonfiguration eingerichtet, die Konfiguration gespeichert und ins Zielsystem geladen werden. Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 608).

Im Dialog "Erweitertes Laden" auf angeschlossene Netzwerkgeräte abfragen

Die Funktion "Laden in Gerät" der S7-1200 CPU und der zugehörigen Dialog "Erweitertes Laden" kann alle erreichbaren Netzwerkgeräte anzeigen und zusätzlich angeben, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Um alle erreichbaren und verfügbaren Geräte mit ihren zugeordneten MAC- oder IP-Adressen anzuzeigen, aktivieren Sie das Optionskästchen "Alle erreichbaren Teilnehmer anzeigen".



Ist das erforderliche Netzwerkgerät in dieser Liste nicht enthalten, so wurde die Kommunikation mit diesem Gerät aus irgendeinem Grund unterbrochen. Das Gerät und das Netzwerk sind dann auf Hardware- und/oder Konfigurationsfehler zu überprüfen.

11.5.5 Ermitteln der Ethernet-Adresse (MAC-Adresse) der CPU

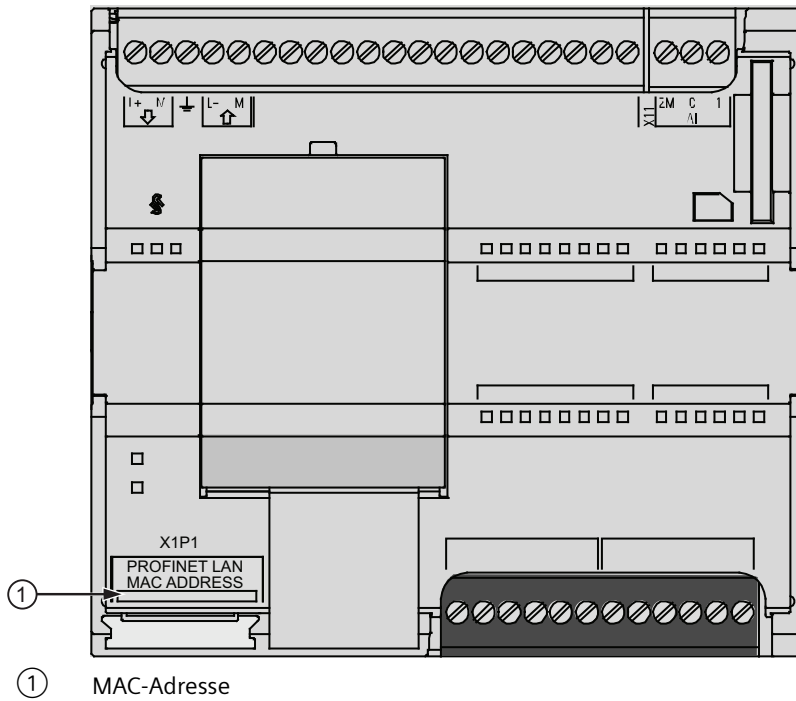
In PROFINET-Netzwerken ist die MAC-Adresse (Media-Access-Control-Adresse) eine Kennung, die den Netzwerkschnittstellen zur eindeutigen Kennzeichnung vom Hersteller zugewiesen wird. Die MAC-Adresse enthält üblicherweise die registrierte Identifikationsnummer des Herstellers.

Das Standardformat (IEEE 802.3) für den Druck von MAC-Adressen als Klartext besteht aus sechs Gruppen zu je zwei Hexadezimalziffern mit Bindestrichen (-) oder Doppelpunkten (:) als Trennzeichen in der Reihenfolge der Übertragung (zum Beispiel 01-23-45-67-89-ab oder 01:23:45:67:89:ab).

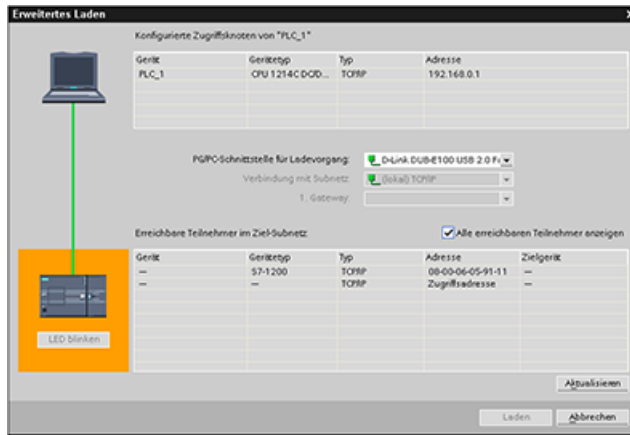
Hinweis

In jeder CPU wird im Werk eine dauerhafte, eindeutige MAC-Adresse eingestellt. Die MAC-Adresse einer CPU können Sie nicht ändern.

Die MAC-Adresse ist an der Vorderseite links unten auf der CPU aufgedruckt. Sie müssen die untere Abdeckklappe öffnen, um die MAC-Adresse lesen zu können.




Die CPU hat zunächst keine IP-Adresse, sondern nur eine im Werk eingestellte MAC-Adresse. In der PROFINET-Kommunikation ist es erforderlich, dass allen Geräten eine eindeutige IP-Adresse zugeordnet wird.



Über die Funktion "Laden in Gerät" der CPU und den Dialog "Erweitertes Laden" können Sie alle erreichbaren Teilnehmer im Netzwerk anzeigen und zusätzlich prüfen, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Dieser Dialog zeigt alle erreichbaren und verfügbaren Geräte mit ihren zugeordneten MAC- oder IP-Adressen an. MAC-Adressen sind besonders wichtig, um Geräte zu identifizieren, die keine eindeutige IP-Adresse haben.

11.5.6 NTP-Synchronisation (Network Time Protocol, NTP) konfigurieren

 WARNUNG
Gefahr unbefugter Zugriffe auf Ihre Netzwerke durch NTP-Synchronisation (Network Time Protocol)
Wenn ein Angreifer über die NTP-Synchronisation (Network Time Protocol) auf Ihre Netzwerke zugreifen kann, kann er Ihnen möglicherweise durch Veränderung der CPU-Systemzeit die Kontrolle über Ihren Prozess entziehen. Durch eine solche Übernahme der Prozesskontrolle kann es zu Tod, schweren Verletzungen oder Sachschäden kommen.
Die S7-1200 CPU deaktiviert die NTP-Client-Funktion standardmäßig. Wird die NTP-Funktion aktiviert, können nur die von Ihnen konfigurierten IP-Adressen als NTP-Server fungieren. Sie müssen die NTP-Funktion so einrichten, dass die CPU-Systemzeit von entfernten Servern korrigiert werden kann.
Die S7-1200 CPU unterstützt Uhrzeitalarme und Uhrzeitoperationen, die von einer korrekten CPU-Systemzeit abhängig sind. Wenn Sie NTP konfigurieren und die Uhrzeitsynchronisierung von einem Server akzeptieren, müssen Sie sicherstellen, dass es sich bei dem Server um eine vertrauenswürdige Quelle handelt. Ist dies nicht der Fall, kann es zu einer Sicherheitsverletzung kommen, bei der ein unbekannter Benutzer die CPU-Systemzeit verändern und Ihnen dadurch die Kontrolle über Ihren Prozess entziehen kann.
Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren Operational Guidelines für Industrial Security (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf) auf der Website "Service & Support" von Siemens.

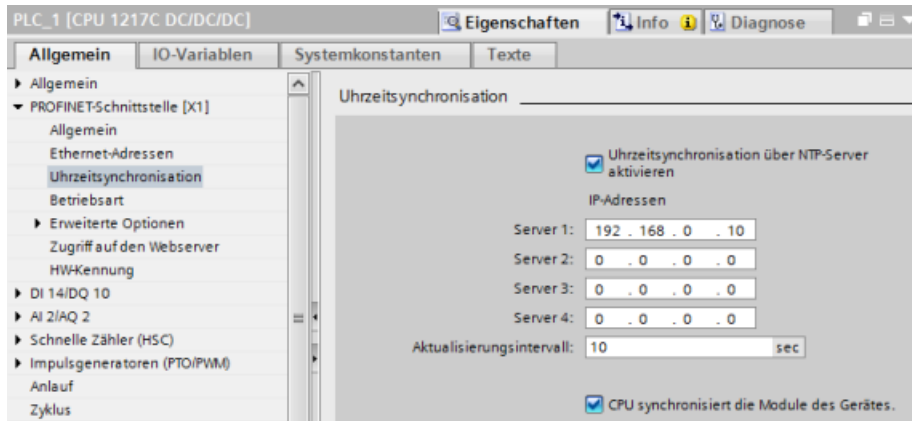
NTP (Network Time Protocol) ist weit verbreitet und dient der Synchronisierung der Uhren von Computersystemen mit Uhrzeitservern im Internet. Beim NTP-Verfahren sendet die CPU in regelmäßigen Zeitabständen Uhrzeitanfragen (im Client-Modus) an den NTP-Server im Subnetz (LAN). Anhand der Antworten der Server wird die zuverlässigste und genaueste Uhrzeit ermittelt und die Uhrzeit der Station synchronisiert.

Der Vorteil dieses Verfahrens ist, dass es Zeit für die subnetzübergreifende Synchronisierung gestattet.

Sie konfigurieren die IP-Adressen von bis zu vier NTP-Servern. Das Aktualisierungsintervall legt den Zeitraum zwischen den Uhrzeitabfragen fest (in Sekunden). Der Wert des Intervalls liegt zwischen 10 Sekunden und einem Tag.

Beim NTP-Verfahren wird im Allgemeinen durch die Server die UTC (Universal Time Coordinated) übertragen, dies entspricht GMT (Greenwich Mean Time).

Wählen Sie im Eigenschaftfenster der CPU Gerätekonfiguration den Eintrag "Uhrzeitsynchronisation". In STEP 7 wird der Dialog "Uhrzeitsynchronisation" angezeigt:



Hinweis

Die CPU empfängt beim Laden des Projekts alle IP-Adressen.

Tabelle 11-9 Parameter für die Uhrzeitsynchronisation

Parameter	Definition
Uhrzeitsynchronisation über NTP-Server aktivieren	Aktivieren Sie das Kontrollkästchen, um die Uhrzeitsynchronisation über den NTP-Server zu aktivieren.
Server 1	Zugewiesene IP-Adresse für Netzwerkzeit-Server 1
Server 2	Zugewiesene IP-Adresse für Netzwerkzeit-Server 2
Server 3	Zugewiesene IP-Adresse für Netzwerkzeit-Server 3
Server 4	Zugewiesene IP-Adresse für Netzwerkzeit-Server 4
Aktualisierungsintervall für die Uhrzeitsynchronisation	Intervallwert (s)
CPU synchronisiert die Module des Gerätes.	Aktivieren Sie das Kontrollkästchen, um die CP-Uhr mit der CPU-Uhr zu synchronisieren.

11.5.7 Anlaufzeit, Benennung und Adresszuweisung von PROFINET-Geräten

PROFINET IO kann die Anlaufzeit für Ihr System verlängern (konfigurierbarer Timeout-Wert). Ein größere Anzahl von Geräten und langsame Geräte wirken sich auf die Zeitdauer aus, die die Umschaltung in RUN benötigt.

Bei V4.0 und späteren Versionen sind maximal 16 PROFINET IO-Geräte im S7-1200 PROFINET-Netzwerk möglich.

Jede Station (bzw. IO-Gerät) startet beim Anlauf unabhängig und dies wirkt sich auf die Gesamtanlaufzeit der CPU aus. Wenn Sie den konfigurierbaren Timeout-Wert zu niedrig setzen, ist möglicherweise nicht genügend Gesamtanlaufzeit der CPU vorhanden, damit alle Stationen den Anlauf beenden. Tritt diese Situation ein, kommt es zu falschen Stationsfehlern.

Die "Zuweisungszeit der Parameter für dezentrale E/A" (Timeout) finden Sie in den CPU-Eigenschaften unter "Anlauf". Der Standardwert für den konfigurierbaren Timeout ist 60.000 ms (1 Minute). Diese Zeit lässt sich vom Anwender konfigurieren.

Namen und Adressen von PROFINET-Geräten in STEP 7

Alle PROFINET-Geräte **müssen** einen Gerätenamen und eine IP-Adresse haben. Sie legen die Gerätenamen und die IP-Adressen in STEP 7 fest. Die Gerätenamen werden über PROFINET DCP (Discovery and Configuration Protocol, Erkennungs- und Konfigurationsprotokoll) in die IO-Geräte geladen.

PROFINET-Adresszuweisung beim Anlauf des Systems

Die Steuerung überträgt die Namen der Geräte zum Netzwerk, und die Geräte reagieren mit ihren MAC-Adressen. Die Steuerung weist dem Gerät dann über das PROFINET DCP-Protokoll eine IP-Adresse zu:

- Wenn die MAC-Adresse eine konfigurierte IP-Adresse hat, führt die Station den Anlauf durch.
- Wenn die MAC-Adresse keine konfigurierte IP-Adresse hat, weist STEP 7 die im Projekt konfigurierte Adresse zu und die Station führt dann den Anlauf durch.
- Gibt es bei diesem Vorgang ein Problem, tritt ein Stationsfehler auf und es findet kein Anlauf statt. Diese Situation verursacht, dass der konfigurierbare Timeout-Wert überschritten wird.

11.5.8 Offene Benutzerkommunikation

11.5.8.1 Protokolle

Der integrierte PROFINET-Port der CPU unterstützt mehrere Kommunikationsstandards über ein Ethernet-Netzwerk:

- Transport Control Protocol (TCP)
- ISO on TCP (RFC 1006)
- User Datagram Protocol (UDP)

Tabelle 11-10 Protokolle und Kommunikationsanweisungen

Protokoll	Verwendungsbeispiele	Eintragen der Daten in den Empfangsbereich	Kommunikationsanweisungen	Adressierungsart
TCP	CPU-zu-CPU-Kommunikation Transport von Telegrammen	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) Portnummern zu
		Datenempfang mit angegebener Länge	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
ISO on TCP	CPU-zu-CPU-Kommunikation Fragmentierung und Zusammensetzung von Meldungen	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
		Protokollgesteuert	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	

Protokoll	Verwendungsbeispiele	Eintragen der Daten in den Empfangsbereich	Kommunikationsanweisungen	Adressierungsart
UDP	CPU-zu-CPU-Kommunikation Kommunikation im Anwenderprogramm	User Datagram Protocol	TUSEND und TURCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) Portnummern zu, es handelt sich jedoch nicht um eine dedizierte Verbindung
S7-Kommunikation	CPU-zu-CPU-Kommunikation Daten aus einer CPU lesen/in eine CPU schreiben	Datenübertragung und -empfang mit angegebener Länge	GET und PUT	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
PROFINET IO	Kommunikation zwischen CPU und PROFINET IO-Device	Datenübertragung und -empfang mit angegebener Länge	Integriert	Integriert

11.5.8.2 TCP und ISO on TCP

Transport Control Protocol (TCP) ist ein Standardprotokoll, beschrieben von RFC 793: Transmission Control Protocol. Wesentlicher Zweck des TCP ist die Bereitstellung einer zuverlässigen, sicheren Verbindung zwischen zwei Paaren von Prozessen. Dieses Protokoll hat die folgenden Merkmale:

- Effizientes Kommunikationsprotokoll, da es eng mit der Hardware verknüpft ist
- Geeignet für mittelgroße bis große Datenmengen (bis 8192 Byte)
- Bietet deutlich mehr Leistungsmerkmale für Anwendungen, insbesondere Wiederherstellung im Fehlerfall, Flusskontrolle und Zuverlässigkeit
- Ein verbindungsorientiertes Protokoll
- Lässt sich sehr flexibel mit Fremdsystemen einsetzen, die nur TCP unterstützen
- Routing-fähig
- Es gelten nur statische Datenlängen.
- Meldungen werden quittiert.
- Anwendungen werden über Portnummern adressiert.
- Die meisten Protokolle von Benutzeranwendungen wie TELNET und FTP nutzen TCP.
- Für die Datenverwaltung ist wegen der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich.

International Standards Organization (ISO) on Transport Control Protocol (TCP) (RFC 1006) (ISO on TCP) ist ein Verfahren, das es ermöglicht, ISO-Anwendungen ins TCP/IP-Netzwerk zu übernehmen. Dieses Protokoll hat die folgenden Merkmale:

- Eng mit der Hardware verknüpftes effizientes Kommunikationsprotokoll
- Geeignet für mittelgroße bis große Datenmengen (bis 8192 Byte)
- Im Gegensatz zu TCP sind die Meldungsfunktion und die Datenendeerkennung meldungsorientiert.

- Routing-fähig, Einsatz in WAN möglich
- Dynamische Datenlängen sind möglich.
- Für die Datenverwaltung ist wegen der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich.

Mit Transport Service Access Points (TSAPs) gestattet das TCP-Protokoll mehrere Verbindungen mit einer einzigen IP-Adresse (bis zu 64-K-Verbindungen). Bei RFC 1006 ermitteln TSAPs die eindeutige Zuordnung dieser Verbindungen der Kommunikationsendpunkte zu einer IP-Adresse.

11.5.8.3 Kommunikationsdienste und verwendete Portnummern

Die S7-1200 CPU unterstützt die in der nachstehenden Tabelle aufgeführten Protokolle. Bei jedem Protokoll weist die CPU die Adressparameter, die entsprechende Kommunikationsschicht sowie die Kommunikationsrolle und die Kommunikationsrichtung zu.

Diese Informationen ermöglichen die Anpassung der Sicherheitsmaßnahmen zum Schutz des Automatisierungssystem an die verwendeten Protokolle (z. B. Firewall). Nur die Ethernet- und PROFINET-Netzwerke haben Sicherheitsmaßnahmen. Da PROFIBUS keine Sicherheitsmaßnahmen bietet, enthält die Tabelle keine PROFIBUS-Protokolle.

Die nachstehende Tabelle zeigt die unterschiedlichen Schichten und Protokolle, die von der CPU verwendet werden:

Protokoll	Portnummer	(2) Sicherungsschicht (4) Transportschicht	Funktion	Beschreibung
PROFINET-Protokolle				
DCP (Discovery and Configuration Protocol, Erkennungs- und Konfigurationsprotokoll)	Nicht relevant	(2) Ethernet II und IEEE 802.1Q und Ethertype 0x8892 (PROFINET)	Erreichbare Teilnehmer PROFINET-Erkennung und -Konfiguration	PROFINET verwendet DCP, um Teilnehmer zu erkennen und Grundeinstellungen bereitzustellen. DCP verwendet die besondere Multicast-MAC-Adresse: xx-xx-xx-01-0E-CF, xx-xx-xx = Organizationally Unique Identifier (Herstellerkennung)
LLDP (Link Layer Discovery Protocol)	Nicht relevant	(2) Ethernet II und IEEE 802.1Q und Ethertype 0x88CC (PROFINET)	PROFINET Link Layer Discovery Protocol	PROFINET verwendet LLDP, um Beziehungen zwischen benachbarten PROFINET-Geräten zu erkennen und zu verwalten. LLDP verwendet die besondere Multicast-MAC-Adresse: 01-80-C2-00-00-0E

11.5.8.4 Ad-hoc-Modus

Typischerweise empfangen die Protokolle TCP und ISO-on-TCP Datenpakete mit fest angegebener Länge von 1 bis 8192 Byte. Die Kommunikationsanweisungen TRCV_C und TRCV jedoch bieten auch einen Ad-hoc-Kommunikationsmodus, in dem Datenpakete variabler Länge von 1 bis 1472 Byte empfangen werden können.

Hinweis

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

Um die Anweisung TRCV_C oder TRCV für den Ad-hoc-Modus zu konfigurieren, setzen Sie den Eingangsparameter ADHOC der Anweisung.

Wenn Sie die Anweisung TRCV_C oder TRCV nicht sehr häufig im Ad-hoc-Modus aufrufen, können Sie in einem Aufruf mehrere Pakete empfangen. Beispiel: Wenn Sie fünf 100-Byte-

Pakete in einem Aufruf empfangen möchten, liefert TCP diese fünf Pakete als ein 500-Byte-Paket, dagegen unterteilt ISO-on-TCP die Pakete in fünf 100-Byte-Pakete.

11.5.8.5 Verbindungs-IDs für Anweisungen für die offene Benutzerkommunikation

Wenn Sie die PROFINET-Anweisung TSEND_C, TRCV_C oder TCON in Ihr Anwenderprogramm einfügen, erstellt STEP 7 einen Instanz-DB für die Konfiguration des Kommunikationskanals (oder der Verbindung) zwischen den Geräten. Die Parameter der Verbindung konfigurieren Sie in den "Eigenschaften" (Seite 600) der Anweisung. Unter den Parametern ist auch die Verbindungs-ID der Verbindung.

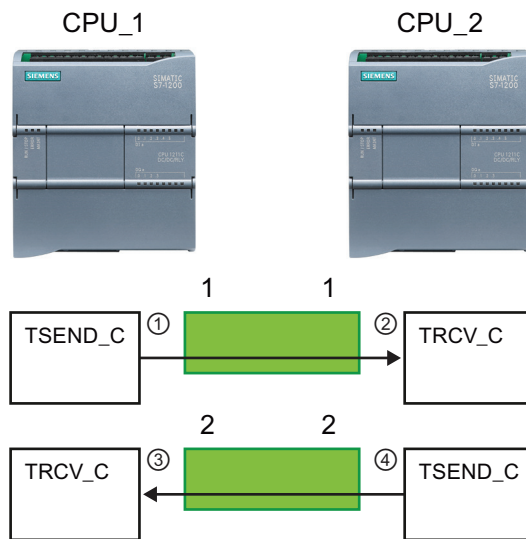
- Die Verbindungs-ID muss für die CPU eindeutig sein. Jede Verbindung, die Sie erstellen, benötigt einen anderen DB und eine andere Verbindungs-ID.
- Die lokale CPU und die Partner-CPU können dieselbe Verbindungs-ID für dieselbe Verbindung verwenden, doch die Verbindungs-IDs müssen nicht übereinstimmen. Die Verbindungs-ID ist nur für die PROFINET-Anweisungen innerhalb des Anwenderprogramms der jeweiligen CPU relevant.
- Für die Verbindungs-ID der CPU können Sie jede beliebige Nummer verwenden. Wenn Sie die Verbindungs-IDs jedoch sequenziell von "1" an vergeben, können Sie leichter die Anzahl der verwendeten Verbindungen für eine bestimmte CPU nachvollziehen.

Hinweis

Jede Anweisung TSEND_C, TRCV_C oder TCON in Ihrem Anwenderprogramm erstellt eine neue Verbindung. Es ist wichtig, dass Sie für jede Verbindung die korrekte Verbindungs-ID verwenden.

Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die zwei getrennte Verbindungen zum Senden und Empfangen von Daten nutzen.

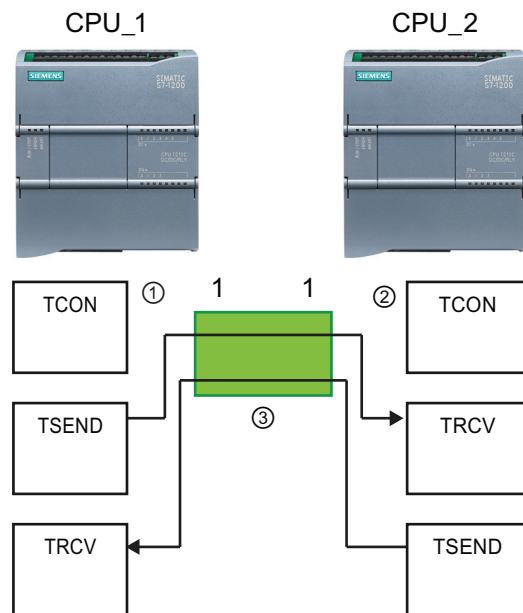
- Die Anweisung TSEND_C in CPU_1 bezieht sich auf die Anweisung TRCV_C in CPU_2 über die erste Verbindung ("Verbindungs-ID 1" bei beiden CPUs, CPU_1 und CPU_2).
- Die Anweisung TRCV_C in CPU_1 bezieht sich auf die Anweisung TSEND_C in CPU_2 über die zweite Verbindung ("Verbindungs-ID 2" bei beiden CPUs, CPU_1 und CPU_2).



- ① TSEND_C in CPU_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID zu (Verbindungs-ID 1 bei CPU_1).
- ② TRCV_C in CPU_2 erstellt die Verbindung für CPU_2 und weist die Verbindungs-ID zu (Verbindungs-ID 1 bei CPU_2).
- ③ TRCV_C in CPU_1 erstellt eine zweite Verbindung für CPU_1 und weist dieser Verbindung eine andere Verbindungs-ID zu (Verbindungs-ID 2 bei CPU_1).
- ④ TSEND_C in CPU_2 erstellt eine zweite Verbindung und weist dieser Verbindung eine andere Verbindungs-ID zu (Verbindungs-ID 2 bei CPU_2).

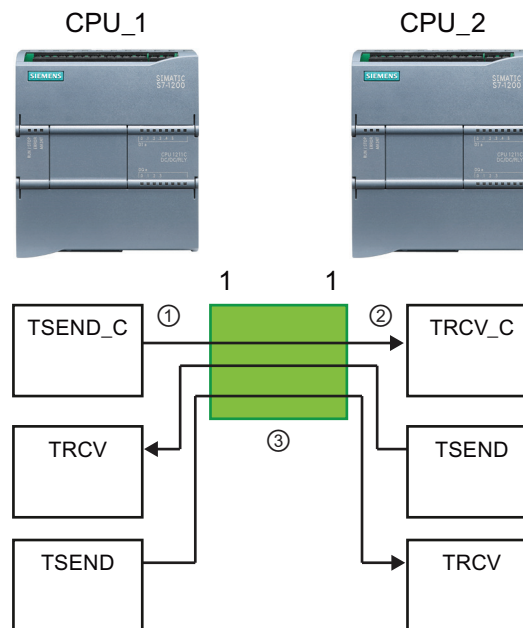
Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die nur eine Verbindung zum Senden und Empfangen von Daten nutzen.

- Jede CPU nutzt eine Anweisung TCON, um die Verbindung zwischen den beiden CPUs zu konfigurieren.
- Die Anweisung TSEND in CPU_1 bezieht sich auf die Anweisung TRCV in CPU_2 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU_1 konfiguriert wurde. Die Anweisung TRCV in CPU_2 bezieht sich auf die Anweisung TSEND in CPU_1 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU_2 konfiguriert wurde.
- Die Anweisung TSEND in CPU_2 bezieht sich auf die Anweisung TRCV in CPU_1 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU_2 konfiguriert wurde. Die Anweisung TRCV in CPU_1 bezieht sich auf die Anweisung TSEND in CPU_2 über die Verbindungs-ID ("Verbindungs-ID 1"), die von der Anweisung TCON in CPU_1 konfiguriert wurde.



- ① TCON in CPU_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU_1 zu (ID = 1).
- ② TCON in CPU_2 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU_2 zu (ID = 1).
- ③ TSEND und TRCV in CPU_1 nutzen die von TCON in CPU_1 erstellte Verbindungs-ID (ID = 1).
TSEND und TRCV in CPU_2 nutzen die von TCON in CPU_2 erstellte Verbindungs-ID (ID = 1).

Wie das folgende Beispiel zeigt, können Sie auch mit einzelnen Anweisungen TSEND und TRCV über eine von einer Anweisung TSEND_C oder TRCV_C erstellte Verbindung kommunizieren. Die Anweisungen TSEND und TRCV erstellen selbst keine neue Verbindung, deshalb müssen Sie den DB und die Verbindungs-ID nutzen, die von einer Anweisung TSEND_C, TRCV_C oder TCON erstellt wurden.



- ① TSEND_C in CPU_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID zu (ID = 1).
- ② TRCV_C in CPU_2 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU_2 zu (ID = 1).
- ③ TSEND und TRCV in CPU_1 nutzen die von TSEND_C in CPU_1 erstellte Verbindungs-ID (ID = 1).
TSEND und TRCV in CPU_2 nutzen die von TRCV_C in CPU_2 erstellte Verbindungs-ID (ID = 1).

11.5.8.6 Parameter für die PROFINET-Verbindung

Die Anweisungen TSEND_C, TRCV_C und TCON benötigen verbindungsbezogene Parameter, um eine Verbindung zum Partnergerät aufbauen zu können. Diese Parameter werden von der Struktur TCON_Param für die Protokolle TCP, ISO-on-TCP und UDP zugewiesen. Üblicherweise geben Sie diese Parameter in den "Eigenschaften" der Anweisung im Register "Konfiguration" (Seite 600) an. Kann auf das Register "Konfiguration" nicht zugegriffen werden, müssen Sie die Struktur TCON_Param in den Parametern der Anweisung angeben.

In V4.1 oder höher parametrisiert die Struktur TCON_IP_V4 das TCP-Protokoll und die Struktur TCON_IP_RFC parametrisiert das ISO-on-TCP-Protokoll.

Ab V4.3 ordnet die Struktur TCON_IP_V4_SEC weitere Parameter für das TCP-Protokoll zu. Um die sichere TCP-Kommunikation zwischen zwei S7-1200 CPUs aufzubauen, müssen Sie mit dem Systemdatentyp TCON_IP_V4_SEC oder in jeder CPU einen Datenbaustein erstellen, die Parametrierung durchführen und den Aufruf direkt an der Anweisung vornehmen. Die Anweisungen TCON, TSEND_C und TRCV_C unterstützen den Systemdatentyp TCON_IP_V4_SEC.

Ab V4.4 verwenden Sie die Struktur TCON_QDN und TCON_QDN_SEC, um die Kommunikationsverbindungen für TCP und UDP über den vollqualifizierten Domänennamen zu konfigurieren, und Sie verwenden die Struktur TCON_QDN_SEC, um die Kommunikationsverbindungen für TCP über den vollqualifizierten Domänennamen mit Secure Communication zu konfigurieren.

TCON_Param

Tabelle 11-11 Struktur der Verbindungsbeschreibung (TCON_Param)

Byte	Parameter und Datentyp		Beschreibung
0 bis 1	block_length	UInt	Länge: 64 Bytes (fest)
2 bis 3	id	CONN_OUC (Word)	Referenz auf diese Verbindung: Wertebereich: 1 (Standard) bis 4095. Geben Sie den Wert dieses Parameters für die Anweisung TSEND_C, TRCV_C oder TCON unter ID an.
4	connection_type	UInt	Verbindungstyp: <ul style="list-style-type: none"> • 17: TCP (Standard) • 18: ISO-on-TCP • 19: UDP
5	active_est	Bool	ID der Verbindungsart: <ul style="list-style-type: none"> • TCP und ISO-on-TCP: <ul style="list-style-type: none"> – FALSCH: Passive Verbindung – WAHR: Aktive Verbindung (Standard) • UDP: FALSCH
6	local_device_id	UInt	ID der lokalen PROFINET- oder Industrial Ethernet-Schnittstelle: 1 (Standard)
7	local_tsap_id_len	UInt	Länge des Parameters local_tsap_id in Bytes; mögliche Werte: <ul style="list-style-type: none"> • TCP: 0 (aktiv, Standard) oder 2 (passiv) • ISO-on-TCP: 2 bis 16 • UDP: 2
8	rem_subnet_id_len	UInt	Dieser Parameter wird nicht verwendet.

Byte	Parameter und Datentyp		Beschreibung
9	rem_staddr_len	USInt	Länge der Adresse des Partnerendpunkts in Bytes: <ul style="list-style-type: none"> • 0: nicht angegeben (Parameter rem_staddr ist irrelevant) • 4 (Standard): Gültige IP-Adresse im Parameter rem_staddr (nur bei TCP und ISO-on-TCP)
10	rem_tsap_id_len	USInt	Länge des Parameters rem_tsap_id in Bytes; mögliche Werte: <ul style="list-style-type: none"> • TCP: 0 (passiv) oder 2 (aktiv, Standard) • ISO-on-TCP: 2 bis 16 • UDP: 0
11	next_staddr_len	USInt	Dieser Parameter wird nicht verwendet.
12 bis 27	local_tsap_id	Array [1..16] of Byte	Komponente der lokalen Adresse der Verbindung: <ul style="list-style-type: none"> • TCP und ISO-on-TCP: lokale Port-Nr. (mögliche Werte: 1 bis 49151; empfohlene Werte: 2000...5000): <ul style="list-style-type: none"> – local_tsap_id[1] = High Byte der Portnummer in Hexadezimalnotation; – local_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotation; – local_tsap_id[3-16] = irrelevant • ISO-on-TCP: lokale TSAP-ID: <ul style="list-style-type: none"> – local_tsap_id[1] = B#16#E0; – local_tsap_id[2] = Baugruppenträger und Steckplatz der lokalen Endpunkte (Bits 0 bis 4: Steckplatznummer, Bits 5 bis 7: Nummer des Baugruppenträgers); – local_tsap_id[3-16] = TSAP-Erweiterung, optional • UDP: Dieser Parameter wird nicht verwendet. Hinweis: Stellen Sie sicher, dass jeder Wert von local_tsap_id innerhalb der CPU eindeutig ist.
28 bis 33	rem_subnet_id	Array [1..6] of USInt	Dieser Parameter wird nicht verwendet.
34 bis 39	rem_staddr	Array [1..6] of USInt	Nur bei TCP und ISO-on-TCP: IP-Adresse des Partnerendpunkts. (Nicht relevant bei passiven Verbindungen.) Beispiel: Die IP-Adresse 192.168.002.003 wird in den folgenden Elementen des Arrays gespeichert: <pre>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = irrelevant</pre>

Byte	Parameter und Datentyp		Beschreibung
40 bis 55	rem_tsap_id	Array [1..16] of Byte	Komponente der Partneradresse der Verbindung: <ul style="list-style-type: none"> TCP: Portnummer der Partner-CPU. Bereich: 1 bis 49151; empfohlene Werte: 2000 bis 5000): <ul style="list-style-type: none"> rem_tsap_id[1] = High Byte der Portnummer in Hexadezimalnotierung; rem_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotierung; rem_tsap_id[3-16] = irrelevant ISO-on-TCP: Partner-TSAP-ID: <ul style="list-style-type: none"> rem_tsap_id[1] = B#16#E0 rem_tsap_id[2] = Baugruppenträger und Steckplatz des Partnerendpunkts (Bits 0 bis 4: Steckplatznummer, Bits 5 bis 7: Nummer des Baugruppenträgers) rem_tsap_id[3-16] = TSAP-Erweiterung, optional UDP: Dieser Parameter wird nicht verwendet.
56 bis 61	next_staddr	Array [1..6] of Byte	Dieser Parameter wird nicht verwendet.
62 bis 63	spare	Word	Reserviert: W#16#0000

TCON_IP_V4

Tabelle 11-12 Struktur der Verbindungsbeschreibung (TCON_IP_V4): Zur Verwendung mit TCP

Byte	Parameter und Datentyp		Beschreibung
0 bis 1	Interfaceld	HW_ANY	HW-Kennung des IE-Schnittstellen-Submoduls
2 bis 3	ID	CONN_OUC (Word)	Verweis auf diese Verbindung: Wertebereich: 1 (Standard) bis 4095. Geben Sie den Wert dieses Parameters für die Anweisung TSEND_C, TRCV_C oder TCON unter ID an.
4	ConnectionType	Byte	Verbindungstyp: <ul style="list-style-type: none"> 11: TCP/IP (Standard) 17: TCP/IP (Dieser Verbindungstyp ist aus Gründen der Anbindung älterer Systeme vorhanden. Empfohlen wird die Verwendung von "11: TCP/IP (Standard)".) 19: UDP
5	ActiveEstablished	Bool	Aktiver/passiver Verbindungsaufbau: <ul style="list-style-type: none"> WAHR: Aktive Verbindung (Standard) FALSCH: Passive Verbindung
V4 IP-Adresse			
6	ADDR[1]	Byte	Oktett 1
7	ADDR[1]	Byte	Oktett 2
8	ADDR[1]	Byte	Oktett 3
9	ADDR[1]	Byte	Oktett 4
10 bis 11	RemotePort	UInt	Dezentrale UDP/TCP-Portnummer
12 bis 13	LocalPort	UInt	Lokale UDP/TCP-Portnummer

TCON_IP_V4_SEC

Tabelle 11-13 Struktur der Verbindungsbeschreibung (TCON_IP_V4_SEC): Zur Verwendung mit TCP

Byte	Parameter und Datentyp		Beschreibung
0 bis 15	ConnPara	TCON_IP_v4	SDT für die Verbindungsparameter Informationen über die Schnittstellenkennung (interface_id): <ul style="list-style-type: none"> • Wenn Sie für die Schnittstellenkennung den voreingestellten Wert 0 übernehmen, wertet das Betriebssystem der CPU die entfernte IP-Adresse und die lokal vorhandenen IP-Routen aus und gibt dann eine Industrial Ethernet-Schnittstelle der CPU zum Herstellen der sicheren OUC-Verbindung an. In diesem Fall werden die Diagnosedaten immer der ersten Industrial Ethernet-Schnittstelle der CPU zugeordnet. • Wenn Sie die Hardwarekennung einer Industrial Ethernet-Schnittstelle der CPU oder eines CP als Schnittstellenkennung angeben, wird die sichere OUC-Verbindung über die zugehörige Industrial Ethernet-Schnittstelle hergestellt.
16	ActivateSecure-Conn	Bool	Aktivierung von Secure Communication für diese Verbindung Wenn dieser Parameter den Wert FALSCH (Voreinstellung) hat, sind die nachfolgenden Sicherheitsparameter irrelevant, was bedeutet, dass die Verbindung nicht sicher ist. Sie können in diesem Fall eine nicht sichere TCP- oder UDP-Verbindung einrichten.
17	TLSServerReq-ClientCert	Bool	Nur für die Server-Seite: Anforderung eines Zertifikats X.509-V3 vom TLS-Client. FALSCH (Standard)
18 bis 19	ExtTlSCapabilities	Word	<ul style="list-style-type: none"> • Bit 0: Nur für die Client-Seite. Ein gesetztes Bit bedeutet, dass der Client den alternativen Namen des Zertifikatsubjekts (subjectAltName) im Zertifikat X.509-V3 des Servers validiert, um die Identität des Servers zu prüfen. Die Zertifikate werden geprüft, wenn die Verbindung hergestellt ist. 16#0 (Standard) • Bit 1 bis 15: Reserviert für künftige Upgrades
20 bis 23	TLSServerCertRef	UDInt	<ul style="list-style-type: none"> • Server-Seite: ID des eigenen Zertifikats X.509-V3 • Client-Seite: ID des Zertifikats X.509-V3 (üblicherweise ein CA-Zertifikat), das vom TLS-Client zur Validierung der Authentifizierung des TLS-Servers verwendet wird. Wenn dieser Parameter 0 ist, verwendet der TLS-Client alle derzeit in den Client-Zertifikatsspeicher geladenen (CA)-Zertifikate, um die Authentifizierung des Servers zu validieren. 0 (Standard)
24 bis 27	TLSClientCertRef	UDInt	<ul style="list-style-type: none"> • Client-Seite: ID des eigenen Zertifikats X.509-V3 • Server-Seite: ID des Zertifikats X.509-V3 (oder eine Gruppe von X.509-V3-Zertifikaten), die vom TLS-Server zur Validierung des TLS-Clients verwendet wird. Wenn dieser Parameter 0 ist, verwendet der TLS-Server alle derzeit in den Server-Zertifikatsspeicher geladenen (CA)-Zertifikate, um die Authentifizierung des Clients zu validieren. 0 (Standard)

Der Verbindungsparameter CONNECT der Instanz-DBs für die Anweisungen TCON, TSEND_C und TRCV_C enthält einen Verweis auf den verwendeten Datenbaustein.

Hinweis

Sie können nicht sichere TCP- oder UDP-Verbindungen über IPv4 herstellen.

Sie können auch den SDT TCON_IP_V4_SEC für eine nicht sichere TCP- oder UDP-Verbindung über IPv4 verwenden.

TCON_IP_RFC

Tabelle 11-14 Struktur der Verbindungsbeschreibung (TCON_IP_RFC): Zur Verwendung mit ISO-on-TCP

Byte	Parameter und Datentyp		Beschreibung
0 bis 1	InterfaceId	HW_ANY	HW-Kennung des IE-Schnittstellen-Submoduls
2 bis 3	ID	CONN_OUC (Word)	Verweis auf diese Verbindung: Wertebereich: 1 (Standard) bis 4095. Geben Sie den Wert dieses Parameters für die Anweisung TSEND_C, TRCV_C oder TCON unter ID an.
4	ConnectionType	Byte	Verbindungstyp: <ul style="list-style-type: none"> 12: ISO-on-TCP (Standard) 17: ISO-on-TCP (Dieser Verbindungstyp ist aus Gründen der Anbindung älterer Systeme vorhanden. Empfohlen wird die Verwendung von "12: ISO-on-TCP (Standard)".)
5	ActiveEstablished	Bool	Aktiver/passiver Verbindungsaufbau: <ul style="list-style-type: none"> WAHR: Aktive Verbindung (Standard) FALSCH: Passive Verbindung
6 bis 7	Reserve		Nicht verwendet
	V4 IP-Adresse		
8	ADDR[1]	Byte	Oktett 1
9	ADDR[1]	Byte	Oktett 2
10	ADDR[1]	Byte	Oktett 3
11	ADDR[1]	Byte	Oktett 4
	Auswahl dezentrale Übertragung		
12 bis 13	TSelLength	UInt	Länge von TSelector
14 bis 45	TSel	array [1..32] of Byte	Zeichenfeld für TSAP-Name
	Auswahl lokale Übertragung		
46 bis 47	TSelLength	UInt	Länge von TSelector
48 bis 79	TSel	array [1..32] of Byte	Zeichenfeld für TSAP-Name

TCON_QDN

Tabelle 11-15 Struktur der Verbindungsbeschreibung in Übereinstimmung mit TCON_QDN

Byte	Parameter und Datentyp		Beschreibung
0 bis 1	Interfaceld	HW_ANY	S7-1200 CPUs ab Firmware V4.4: <ul style="list-style-type: none"> • Interfaceld eines gesteckten CP: <ul style="list-style-type: none"> – Bei CPs von S7-1200 ab Firmware V3.2 – Bei CPs von ET 200SP ab Firmware V2 S7-1200 CPUs vor V4.4: <ul style="list-style-type: none"> • Parameter irrelevant
2 bis 3	ID	CONN_O UC	Verweis auf diese Verbindung (Wertebereich: 1 bis 4095). Geben Sie den Wert dieses Parameters für die Anweisung TCON unter ID an.
4	ConnectionType	BYTE	Verbindungstyp: <ul style="list-style-type: none"> • 11: TCP (11 dez. = 0x0B hex.) • 19: UDP (19 dez. = 0x13 hex.)
5	ActiveEstablished	BOOL	Kennung für die Art des Verbindungsaufbaus: <ul style="list-style-type: none"> • FALSCH: Passiver Verbindungsaufbau • WAHR: Aktiver Verbindungsaufbau
6 bis 261	RemoteQDN	Array aus STRING [1..254]	Vollqualifizierter Domänenname des Partner-Endpunkts, der auf "." enden muss. Beachten Sie, dass in einem SIMATIC-Netzwerk der Name einschließlich des abschließenden Punkts 254 Zeichen nicht überschreiten darf.
262 bis 263	RemotePort	UINT	Port-Adresse des entfernten Verbindungspartners
264 bis 265	LocalPort	UINT	Port-Adresse des lokalen Verbindungspartners

TCON_QDN_SEC

Tabelle 11-16 Struktur der Verbindungsbeschreibung gemäß TCON_QDN_SEC

Byte	Parameter und Datentyp		Beschreibung
0 bis 271	ConnPara	TCON_QDN	Verbindungsparameter
272	ActivateSecureConn	BOOL	Aktivierung von Secure Communication für diese Verbindung. Wenn dieser Parameter den Wert FALSCH hat, sind die nachfolgenden Sicherheitsparameter irrelevant. Sie können in diesem Fall eine nicht sichere TCP- oder UDP-Verbindung einrichten.
273	TLSReqClientCert	BOOL	Nur serverseitig: Anforderung eines Zertifikats X.509-V3 vom TLS-Client

Byte	Parameter und Datentyp		Beschreibung
274 bis 275	ExtTlSCapabilities	WORD	<ul style="list-style-type: none"> • Bit 0: Nur für die Client-Seite. Ein gesetztes Bit bedeutet, dass der Client den alternativen Namen des Zertifikatausstellers (Subject Alternate Name) im Zertifikat X.509-V3 des Servers validiert, um die Identität des Servers zu prüfen. Die Zertifikate werden geprüft, wenn die Verbindung hergestellt ist. • Bit 1 bis 15: reserviert für künftige Upgrades
276 bis 279	TLSserverCertRef	UDINT	<ul style="list-style-type: none"> • Serverseitig: ID des eigenen Zertifikats X.509-V3 • Clientseitig: ID des X.509-V3-Zertifikats (üblicherweise ein CA-Zertifikat), das vom TLS-Client verwendet wird, um die Authentifizierung des TLS-Servers zu authentifizieren. Wenn dieser Parameter "0" ist, verwendet der TLS-Client alle derzeit in den Client-Zertifikatspeicher geladenen (CA)-Zertifikate, um die Authentifizierung des Servers zu validieren.
280 bis 283	TLSclientCertRef	UDINT	<ul style="list-style-type: none"> • Clientseitig: ID des eigenen Zertifikats X.509-V3 • Serverseitig: ID des Zertifikats X.509-V3 (oder eine Gruppe von X.509-V3-Zertifikaten), die vom TLS-Server zur Validierung der TLS-Client-Authentifizierung verwendet wird. Wenn dieser Parameter 0 ist, verwendet der TLS-Server alle derzeit in den Server-Zertifikatspeicher geladenen (CA)-Zertifikate, um die Authentifizierung des Clients zu validieren.

Unterstützte TLS-Version

TLS steht für die Sicherheit der Transportschicht (Transport Layer Security) in der Anwendungsschicht der Datenkommunikation. TLS erhöht die Sicherheit und den Datenschutz in der Kommunikation zwischen der S7-1200 CPU und anderen Geräten. Die TLS-Version ist abhängig von der Firmware-Version der S7-1200 CPU und von der Version der CPU in der Gerätekonfiguration Ihres STEP 7-Projekts. Um die CPU-Version in Ihrem STEP 7-Projekt zu finden, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Gerätekonfiguration auf die CPU.
2. Sehen Sie sich im Inspektorfenster im Register "Eigenschaften" den Abschnitt "Allgemein" an.
3. Die Firmware-Version wird im Abschnitt zu den Kataloginformationen angezeigt.

Um die höchste Version von TLS zu verwenden, konfigurieren Sie für die CPU-Version in STEP 7 die tatsächliche Firmware-Version Ihrer CPU. Der Partner der Open User Communication kann dann die höchste unterstützte TLS-Version für maximale Sicherheit verwenden.

Tabelle 11-17 Unterstützte TLS-Version basierend auf der Firmware-Version von CPU und STEP 7-Projekt

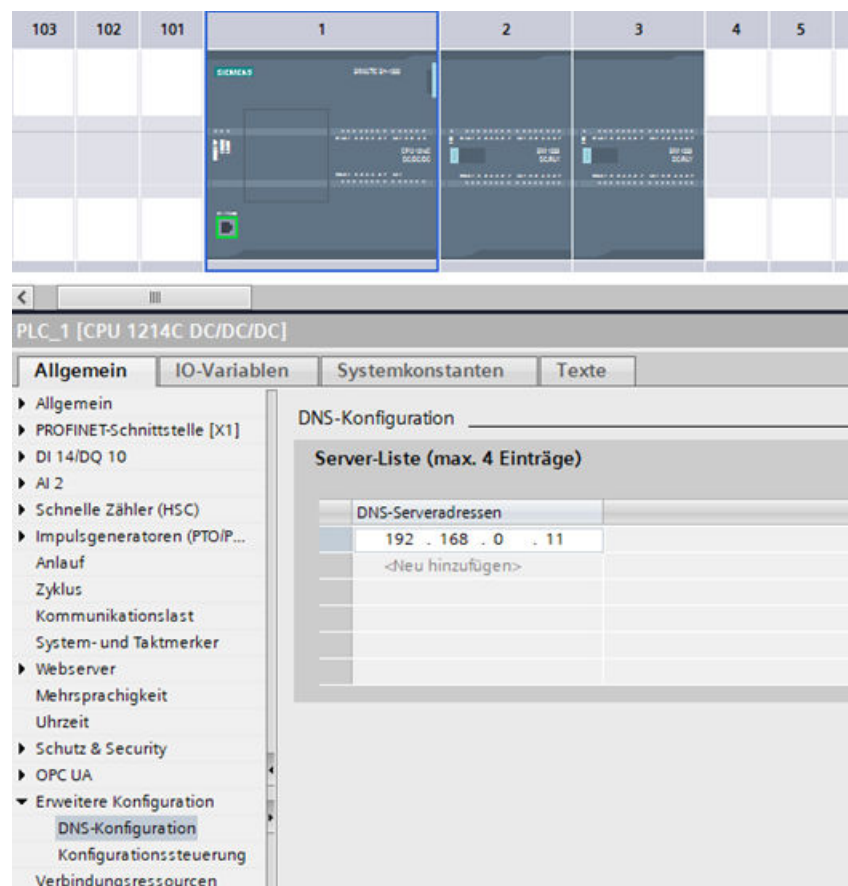
Im STEP 7-Projekt konfigurierte Firmware-Version:	Unterstützte TLS-Version
V4.5	TLS 1.2, TLS 1.3
V4.4	TLS 1.2
V4.3	TLS 1.2

11.5.8.7 Konfigurieren eines DNS

Für die sichere offene Benutzerkommunikation (OUC) müssen Sie ein Domain Name System (DNS) konfigurieren. In Ihrem Netzwerk muss es mindestens einen DNS-Server geben und Sie müssen mindestens einen DNS-Server für die S7-1200 CPU konfigurieren.

Einen DNS-Server konfigurieren Sie wie folgt:

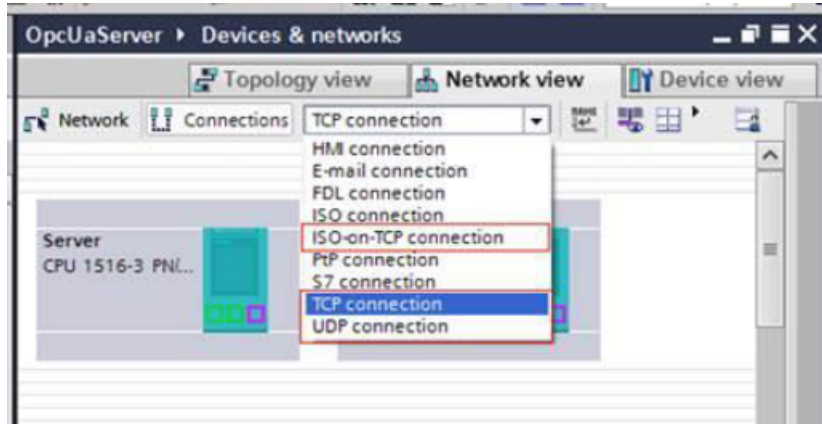
1. Navigieren Sie zur Geräteansicht für Ihre S7-1200 CPU.
2. Gehen Sie zur Seite "Eigenschaften" und öffnen Sie das Register "Allgemein".
3. Klicken Sie auf die DNS-Konfiguration, um die Konfigurationsseite anzuzeigen.
4. Klicken Sie in der Tabelle mit der Serverliste in der ersten Zeile unter den Adressen der DNS-Server auf <Neu hinzufügen> und geben Sie die IP-Adresse Ihres DNS-Servers ein.



11.5.8.8 OUC-Verbindung im TIA Portal V17 konfigurieren

Im TIA Portal V17 können Sie die folgenden Open-User-Communication-Verbindungen auswählen (wie unten gezeigt), um eine Verbindung zu oder von den S7-1200 bzw. S7-1500 CPUs zu ziehen.

- ISO-on-TCP-Verbindung
- TCP-Verbindung
- UDP-Verbindung



Wenn Sie zwischen Geräten eine Linie ziehen, wird eine Verbindung konfiguriert, um das Gerät zu übersetzen und ins Gerät zu laden. Diese Verbindungskonfiguration ermöglicht es der S7-1200 Firmware, eine Verbindung mit dem Partner herzustellen, wenn die CPU in RUN geht. Bei einer konfigurierten Verbindung braucht keine Anweisung TCON ausgeführt zu werden. Auch braucht für eine konfigurierte Verbindung keine Anweisung T_DISCON ausgeführt zu werden.

Um für diese Verbindungen eine Linie zu ziehen, müssen sich beide Netzwerkschnittstellen für die CPU (oder den CP) im selben Subnetz befinden. Das TIA Portal hindert Sie nicht daran, eine Verbindung zu Geräten in anderen Netzwerken zu ziehen. Das TIA Portal generiert beim Übersetzen oder Laden ins Gerät jedoch einen Fehler.

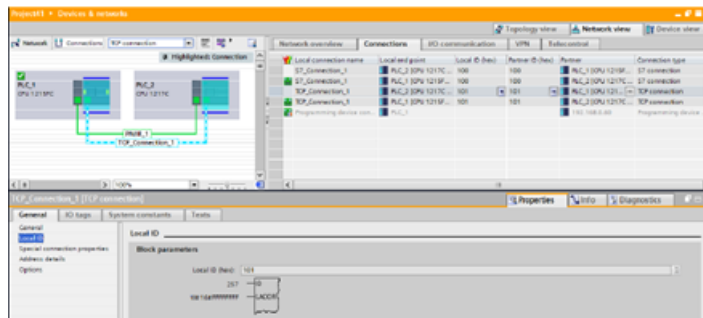
Sie können jetzt eine OUC-Verbindung zwischen den S7-1200 und S7-1500 CPUs ziehen, die Konfiguration laden und die Verbindung zwischen den CPUs automatisch herstellen (sofern die Verbindung physisch möglich ist).

Konfigurationsoptionen für konfigurierte OUC-Verbindungen

Sie konfigurieren die folgenden Eigenschaften der Verbindung wie nachfolgend gezeigt:

- Verbindungs-ID
- Name der Verbindung

- Welcher Partner führt den aktiven Verbindungsaufbau durch
- Portdetails im Menü "Eigenschaften" der Netzsicht durch Auswahl eines Verbindungsendes

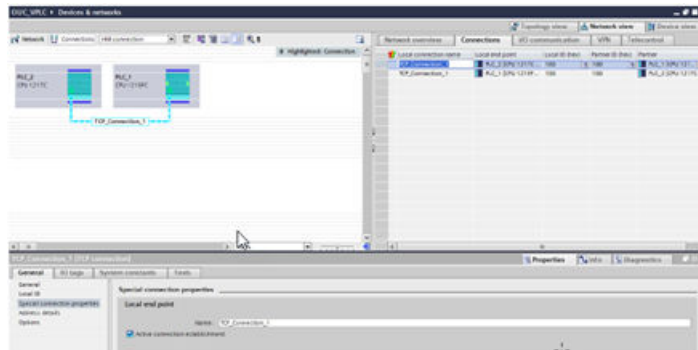


Wenn Sie im TIA Portal eine OUC-Verbindung ziehen, werden innerhalb des gültigen Bereichs für eine OUC-Verbindungs-ID eine "Lokale ID" und eine "Partner-ID" zugewiesen. Sie können den zugewiesenen Wert in der Verbindungstabelle oder in der lokalen ID ändern. Der für jede ID eingegebene Wert muss sich innerhalb des von den TBAustein-Anweisungen definierten Bereichs befinden (siehe TSEND).

Hinweis

Zugeordneter Bereich für die Verbindungs-ID

Der für die Verbindungs-ID zugeordnete Bereich befindet sich im selben Bereich wie der für eine S7-Verbindung zugeordnete Bereich. Wenn eine S7-Verbindungs-ID in eine Anweisung TSEND eingegeben wird, führt das zu einem Fehler seitens der Anweisung. Der Fehler lautet 16#80A1, da keine OUC-Verbindung hergestellt wird.



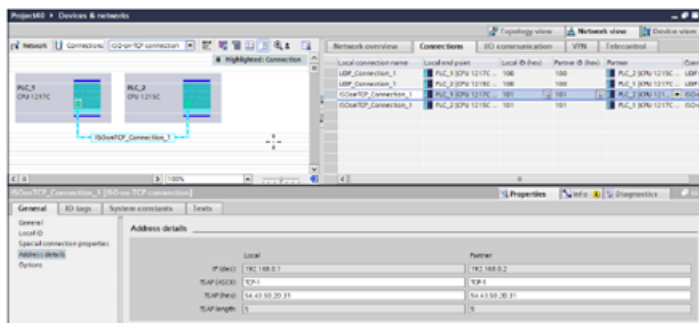
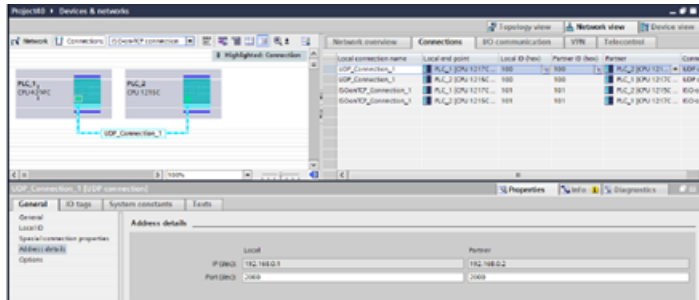
Im Menü "Spezielle Verbindungseigenschaften" definiert das Feld "Aktiver Verbindungsaufbau" das Gerät, das nach dem Laden der Konfiguration in die CPUs Verbindungsmeldungen aussendet. Ist das Kontrollkästchen "Aktiver Verbindungsaufbau" nicht aktiviert, wartet das geladene Gerät auf Verbindungsmeldungen von seinem Partner. Das TIA Portal aktualisiert den Partner automatisch, wenn das Kontrollkästchen "Aktiver Verbindungsaufbau" aktiviert wird. Es kann nur eine Seite der Verbindung als aktive Seite festgelegt werden.

Hinweis

UDP-Verbindungen

Bei UDP-Verbindungen ist die Option "Aktiver Verbindungsaufbau" nicht vorhanden.

Im Eigenschaftmenü "Adressdetails" wird die Konfiguration der Adressen angezeigt, die für die Kommunikation der Verbindung für TCP und ISO-on-TCP verwendet werden. Wenn Sie die Anweisungen TSEND und TURCV bei einer UDP-Verbindung verwenden, wird die Adresse für die Kommunikation von einem Parameter in den Anweisungen übersteuert. Außerdem können bei den Verbindungstypen Iso-on-TCP die TSAPs im Eigenschaftmenü "Adressdetails" wie in der zweiten Abbildung unten gezeigt geändert werden.



Funktionsweise von vorhandenen TBaustein-Anweisungen mit konfigurierten Verbindungen

Sobald eine Verbindung hergestellt ist, funktioniert eine konfigurierte Verbindung genauso wie eine programmierte Verbindung. Andere OUC-Anweisungen funktionieren genau wie bei Ausführung über programmierte Verbindungen. Nachfolgend finden Sie eine Beschreibung von OUC-Anweisungen, und diese arbeiten über konfigurierte Verbindungen.

OUC-Anweisung	Beschreibung mit konfigurierter Verbindung
TSEND_C	Ein SDT TCON_Configured muss an den Parameter Connect übergeben werden. Wenn eine konfigurierte Verbindung verwendet wird, muss im ID-Feld von TCON_Configured die konfigurierte Verbindungs-ID und im Feld "Verbindungstyp" der Wert 254 festgelegt werden. Bei Verwendung einer konfigurierten Verbindung muss der Verbindungsmechanismus in der Anweisung übersprungen werden, da die Verbindung bereits hergestellt ist.
TRCV_C	Ein SDT TCON_Configured muss an den Parameter Connect übergeben werden. Wenn eine konfigurierte Verbindung verwendet wird, muss im ID-Feld von TCON_Configured die konfigurierte Verbindungs-ID und im Feld "Verbindungstyp" der Wert 254 festgelegt werden. Bei Verwendung einer konfigurierten Verbindung muss der Verbindungsmechanismus in der Anweisung übersprungen werden, da die Verbindung bereits hergestellt ist.
TMAIL_C	Funktioniert nicht mit einer konfigurierten Verbindung
TCON	Meldet einen Fehler (0x8085), wenn die Verbindungs-ID für eine konfigurierte Verbindung angegeben wird, weil die Verbindung bereits offen ist.

OUC-Anweisung	Beschreibung mit konfigurierter Verbindung
TDISCON	Meldet einen Fehler (0x80A3)
TCONSettings ¹	Meldet einen Fehler (0x8085)
TSEND	Gleiche Funktionsweise
TRCV	Gleiche Funktionsweise
TUSEND	Gleiche Funktionsweise
TURCV	Gleiche Funktionsweise
T_RESET	Gleiche Funktionsweise Beim Verbinden wird die Verbindung getrennt und erneut eine Verbindung aufgebaut. Die gesendeten Bytes und empfangenen Bytes werden auf 0 zurückgesetzt.
T_DIAG	Gleiche Funktionsweise Im Feld "Art" der Varianten TDIAG_Status muss eine konfigurierte Verbindung angezeigt werden.
T_CONFIG	Gleiche Funktionsweise (nicht in direktem Zusammenhang mit der Kommunikation nach OUC-Protokoll)
MB_CLIENT	Ein SDT TCON_Configured muss an den Parameter Connect übergeben werden. Wenn eine konfigurierte Verbindung verwendet wird, muss im ID-Feld von TCON_Configured die konfigurierte Verbindungs-ID und im Feld "Verbindungstyp" der Wert 254 festgelegt werden. Bei Verwendung einer konfigurierten Verbindung muss der Verbindungsmechanismus in der Anweisung übersprungen werden, da die Verbindung bereits hergestellt ist.
MB_SERVER	Ein SDT TCON_Configured muss an den Parameter Connect übergeben werden. Wenn eine konfigurierte Verbindung verwendet wird, muss im ID-Feld von TCON_Configured die konfigurierte Verbindungs-ID und im Feld "Verbindungstyp" der Wert 254 festgelegt werden. Bei Verwendung einer konfigurierten Verbindung muss der Verbindungsmechanismus in der Anweisung übersprungen werden, da die Verbindung bereits hergestellt ist.

¹ TCONSettings gilt nicht für konfigurierte Verbindungen in V4.5. In V4.5 funktioniert bei TCONSettings die Schreibfunktion nicht mit konfigurierten Verbindungen, weil die Option zum normalen Herunterfahren bei konfigurierten Verbindungen nicht verwendet wird. Die Leseoperation gibt bei der Option zum normalen Herunterfahren stets FALSCH zurück.

11.5.8.9 Anweisungen TSEND_C und TRCV_C

Ab Version V4.1 der S7-1200 CPU zusammen mit STEP 7 V13 SP1 oder höher ist die Fähigkeit der Anweisungen TSEND_C und TRCV_C dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4 und TCON_IP_RFC verwendet werden können.

Ab Version V4.3 der S7-1200 CPU zusammen mit STEP 7 V15.1 oder höher ist die Fähigkeit der Anweisungen TSEND_C und TRCV_C dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC und TCON_IP_RFC verwendet werden können.

Ab Version V4.4 der S7-1200 CPU zusammen mit STEP 7 V16 oder höher ist die Fähigkeit der Anweisungen TSEND_C und TRCV_C dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN und TCON_QDN_SEC verwendet werden können.

Aus diesem Grund unterstützt die S7-1200 zwei Sätze TSEND_C- und TRCV_C-Anweisungen:

- Ältere TSEND_C- und TRCV_C-Anweisungen (Seite 649): Die Anweisungen TSEND_C und TRCV_C wurden vor der Version V4.1 der S7-1200 verwendet und funktionieren nur mit Verbindungsparametern mit Strukturen gemäß TCON_Param.
- TSEND_C- und TRCV_C-Anweisungen (Seite 636): Diese TSEND_C- und TRCV_C-Anweisungen stellen alle Funktionen der früheren Anweisungen bereit und bieten zusätzlich die Möglichkeit, Verbindungsparameter mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN, and TCON_QDN_SEC zu nutzen.

Version der Anweisungen TSEND_C und TRCV_C auswählen

Es gibt zwei Versionen der Anweisungen TSEND_C und TRCV_C in STEP 7:

- Die Versionen 2.5 und 3.1 waren in STEP 7 Basic/Professional V13 und früher verfügbar.
- Version 4.0 ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Verwenden Sie niemals Anweisungen unterschiedlicher Versionen zusammen in einem CPU-Programm.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.



Um die Version einer TSEND_C- und TRCV_C-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine TSEND_C- oder TRCV_C-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten TSEND_C- oder TRCV_C-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer TSEND_C- oder TRCV_C-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmeditor angezeigten Box. Wählen Sie in der Projektnavigation eine FB- oder FC-Instanz einer TSEND_C- oder TRCV_C Anweisung aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der TSEND_C- oder TRCV_C-Anweisung anzuzeigen.

TSEND_C und TRCV_C (Daten über Ethernet senden und empfangen)

Die Anweisung TSEND_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TSEND. Die Anweisung TRCV_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TRCV. (Weitere Informationen zu diesen Anweisungen finden Sie unter "TCON, TDISCON, TSEND UND TRCV (Seite 657)".)

Die Mindestdatengröße, die gesendet (TSEND_C) oder empfangen (TRCV_C) werden kann, ist ein Byte; die maximale Datengröße beträgt 8192 Byte. TSEND_C unterstützt nicht die Datenübertragung von Booleschen Adressen und TRCV_C empfängt Daten nicht in Booleschen

Adressen. Weitere Informationen zum Übertragen von Daten mit diesen Anweisungen finden Sie im Abschnitt zur Datenkonsistenz (Seite 187).

Hinweis

Initialisierung der Kommunikationsparameter

Nachdem Sie die Anweisung TSEND_C oder TRCV_C eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 600) die Kommunikationsparameter (Seite 624). Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 11-18 Anweisungen TSEND_C und TRCV_C

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>TSEND_C stellt eine TCP- oder ISO-on-TCP-Verbindung zu einem Partner her, sendet Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>
	<pre>"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_);</pre>	<p>TRCV_C stellt eine TCP- oder ISO on TCP-Verbindung zu einer CPU her, empfängt Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-19 Datentypen für die Parameter von TSEND_C und TRCV_C

Parameter und Datentyp	Datentyp	Beschreibung	
REQ (TSEND_C)	IN	Bool	Startet den Sendeauftrag bei einer steigenden Flanke
EN_R (TRCV_C)	IN	Bool	Empfang aktivieren

Parameter und Datentyp		Datentyp	Beschreibung
CONT	IN	Bool	<p>Steuert die Kommunikationsverbindung:</p> <ul style="list-style-type: none"> • 0: Kommunikationsverbindung nach dem Senden der Daten trennen. • 1: Aufbauen und Halten der Kommunikationsverbindung <p>Beim Senden von Daten (TSEND_C) (steigende Flanke am Parameter REQ) oder beim Empfangen von Daten (TRCV_C) (steigende Flanke am Parameter EN_R), muss der Parameter CONT den Wert TRUE haben, damit eine Verbindung aufgebaut oder gehalten werden kann.</p>
LEN	IN	UDInt	<p>Optionalen Parameter (versteckt)</p> <p>Maximale Bytezahl, die mit dem Auftrag gesendet (TSEND_C) oder empfangen (TRCV_C) werden soll. Wenn Sie am Parameter DATA rein symbolische Werte verwenden, muss der Parameter LEN den Wert 0 haben.</p>
ADHOC (TRCV_C)	IN	Bool	<p>Optionalen Parameter (versteckt)</p> <p>Ad-hoc-Modus-Anforderung für Verbindungstyp TCP.</p>
CONNECT	IN_OUT	Variant	<p>Pointer auf die Verbindungsbeschreibung:</p> <ul style="list-style-type: none"> • Bei TCP oder UDP verwenden Sie die Struktur TCON_IP_v4 oder TCON_QDN. Zur Beschreibung siehe: "Parameter für die PROFINET-Verbindung (Seite 624)". • Bei TCP mit sicherer Kommunikation verwenden Sie die Struktur TCON_IP_V4_SEC oder TCON_QDN_SEC. Zur Beschreibung siehe: "Parameter für die PROFINET-Verbindung (Seite 624)". • Bei ISO-on-TCP verwenden Sie die Struktur TCON_IP_RFC. Zur Beschreibung siehe: "Parameter für die PROFINET-Verbindung (Seite 624)". • Bei ISO-Verbindungen des CP 1543-1 / CP 1545-1 verwenden Sie die Struktur TCON_ISOnative. Zur Beschreibung siehe die Online-Hilfe im TIA Portal: "Struktur der Verbindungsbeschreibung nach TCON_ISOnative". • Bei Verbindungen mit SMS-Clients verwenden Sie den Systemdatentyp TCON_PHONE. Zur Beschreibung siehe die Online-Hilfe im TIA Portal: "Verbindungsparameter nach TCON_Phone". • Bei FDL-Verbindungen des CM 1542-5 verwenden Sie den Systemdatentyp TCON_FDL. Siehe hierzu die Online-Hilfe im TIA Portal: "Verbindungsparameter nach TCON_FDL".
DATA	IN_OUT	Variant	<p>Pointer auf den Sendebereich mit:</p> <ul style="list-style-type: none"> • Adresse und Länge der Daten, die gesendet (TSEND_C) werden sollen • Adresse und maximale Länge der empfangenen Daten (TRCV_C)

Parameter und Datentyp		Datentyp	Beschreibung
ADDR	IN_OUT	Variant	Optionaler Parameter (versteckt) Pointer auf die Adresse des Empfängers mit dem Verbindungstyp UDP. Die Adressinformationen werden in der Struktur TADDR_Param ### zugeordnet.
COM_RST	IN_OUT	Bool	Optionaler Parameter (versteckt) Startet die Anweisung neu: <ul style="list-style-type: none"> • 0: Irrelevant • 1: Startet die Anweisung vollständig neu. Die vorhandene Verbindung wird entweder beendet oder zurückgesetzt und gemäß CONT erneut aufgebaut. Der Parameter COM_RST wird nach Auswertung durch die Anweisung TSEND_C oder TRCV_C zurückgesetzt und darf deshalb nicht statisch geschaltet werden.
DONE	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Sendeauftrag wurde noch nicht gestartet oder wird noch ausgeführt. • 1: Sendeauftrag fehlerfrei ausgeführt. Dieser Zustand wird nur einen Zyklus lang angezeigt.
BUSY	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Sendeauftrag wurde noch nicht gestartet oder ist bereits beendet. • 1: Sendeauftrag noch nicht beendet. Es kann kein neuer Sendeauftrag gestartet werden.
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Kein Fehler • 1: Fehler bei Verbindungsaufbau, Datenübertragung oder Verbindungsbeendigung.
STATUS	OUT	Word	Status der Anweisung (siehe Beschreibung der Parameter ERROR und STATUS).
RCVD_LEN (TRCV_C)	OUT	Int	Tatsächliche empfangene Datenmenge (in Byte).

Hinweis

Die Anweisung TSEND_C erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

Hinweis

Die Standardeinstellung des Parameters LEN (LEN=0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Es ist empfehlenswert, dass die von der Anweisung TSEND_C gesendeten Daten die gleiche Größe haben wie der Parameter DATA der Anweisung TRCV_C.

Wenn Sie die Standardeinstellung des Parameters LEN verwenden und die Daten in kleineren Segmenten als die Größe des Parameters DATA gesendet werden sollen, gilt Folgendes. Wenn die Größe der von TSEND_C übertragenen Daten nicht der Größe des Parameters TRCV_C DATA entspricht, bleibt TRCV_C im Status "Belegt" (Statuscode: 7006), bis die Gesamtgröße der von TSEND_C übertragenen Daten der Größe des Parameters TRCV_C DATA entspricht.

Der Puffer des Parameters TRCV_C DATA zeigt die neu empfangenen Daten erst an, wenn die Datengröße gleich der Puffergröße des Parameters DATA ist.

Funktionsweise von TSEND_C

Die Anweisung TSEND_C wird asynchron ausgeführt und implementiert die folgenden Funktionen der Reihe nach:

1. Einrichten und Herstellen einer Kommunikationsverbindung:
TSEND_C richtet eine Kommunikationsverbindung ein und stellt diese Verbindung her, wenn am Parameter REQ eine steigende Flanke erkannt wird und noch keine Kommunikationsverbindung vorhanden ist. Nach dem Einrichten und Herstellen der Verbindung wird diese automatisch von der CPU gehalten und überwacht. Die am Parameter CONNECT angegebene Verbindungsbeschreibung dient zum Einrichten der Kommunikationsverbindung. Die folgenden Verbindungstypen sind verwendbar:
 - Struktur TCON_Param für die Protokolle TCP, ISO-on-TCP und UDP
 - Bei V4.1 und höher, TCP/UDP: Verbindungsbeschreibung über die Struktur TCON_IP_V4 am Parameter CONNECT
 - Bei V4.1 und höher, ISO-on-TCP: Verbindungsbeschreibung über die Struktur TCON_IP_RFC am Parameter CONNECT
 - Bei V4.3 und höher, TCP: Verbindungsbeschreibung über die Struktur TCON_IP_V4_SEC am Parameter CONNECT
 - Bei V4.4 und höher, TCP: Verbindungsbeschreibung über die Strukturen TCON_QDN und TCON_QDN_SEC am Parameter CONNECT

Wenn die CPU in den Betriebszustand STOP wechselt, wird eine vorhandene Verbindung beendet und die eingerichtete Verbindung entfernt. Um die Verbindung wieder einzurichten und herzustellen, müssen Sie TSEND_C erneut ausführen. Informationen zur Anzahl der möglichen Kommunikationsverbindungen finden Sie in den technischen Daten Ihrer CPU.

2. Senden von Daten über eine vorhandene Kommunikationsverbindung:
Der Sendeauftrag wird bei einer steigenden Flanke am Parameter REQ ausgeführt. Wie oben beschrieben, wird zunächst die Kommunikationsverbindung hergestellt. Den Sendebereich geben Sie über den Parameter DATA an. Dies umfasst die Adresse und die Länge der zu sendenden Daten. Verwenden Sie im Parameter DATA keinen Datenbereich mit dem Datentyp BOOL oder Array of BOOL. Mit dem Parameter LEN geben Sie die maximale Anzahl mit einem Auftrag gesendeter Bytes an. Wenn Sie am Parameter DATA einen symbolischen Namen verwenden, muss der Parameter LEN den Wert 0 haben.
Die zu sendenden Daten dürfen vor Fertigstellung des Sendeauftrags nicht bearbeitet werden.
3. Beenden der Kommunikationsverbindung:
Die Kommunikationsverbindung wird nach dem Senden der Daten beendet, wenn der Parameter CONT zum Zeitpunkt der steigenden Flanke am Parameter REQ den Wert 0 hatte. Ansonsten wird die Kommunikationsverbindung aufrechterhalten.

Bei erfolgreicher Ausführung des Sendeauftrags wird der Parameter DONE auf 1 gesetzt. Die Kommunikationsverbindung kann vorab beendet werden (siehe hierzu die Beschreibung oben über die Abhängigkeit vom Parameter CONT). Der Signalzustand 1 am Parameter DONE ist keine Bestätigung dafür, dass die gesendeten Daten bereits vom Kommunikationspartner gelesen wurden.

TSEND_C wird zurückgesetzt, wenn der Parameter COM_RST auf 1 gesetzt wird. Werden zu diesem Zeitpunkt Daten übertragen, kann es zu Datenverlust kommen.

Abhängig vom Parameter CONT sind die folgenden Szenarien möglich:

- CONT = "0":
Eine vorhandene Kommunikationsverbindung wird hergestellt.
- CONT = 1 und eine Kommunikationsverbindung wurde hergestellt:
Eine vorhandene Kommunikationsverbindung wird zurückgesetzt und erneut hergestellt.
- CONT = 1 und es wurde keine Kommunikationsverbindung hergestellt.
Es wird keine Kommunikationsverbindung hergestellt.

Der Parameter COM_RST wird nach Auswertung durch die Anweisung T_SEND zurückgesetzt. Um TSEND_C nach der Ausführung (DONE = 1) erneut zu aktivieren, rufen Sie die Anweisung einmal mit REQ = 0 auf.

Funktionsweise von TRCV_C

Die Anweisung TRCV_C wird asynchron ausgeführt und implementiert die folgenden Funktionen der Reihe nach:

1. Einrichten und Herstellen einer Kommunikationsverbindung:

TRCV_C richtet eine Kommunikationsverbindung ein und stellt diese Verbindung her, wenn der Parameter EN_R = 1 ist und keine Kommunikationsverbindung vorhanden ist. Nach dem Einrichten und Herstellen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.

Die am Parameter CONNECT angegebene Verbindungsbeschreibung dient zum Einrichten der Kommunikationsverbindung. Die folgenden Verbindungstypen sind verwendbar:

- Struktur TCON_Param für die Protokolle TCP, ISO-on-TCP und UDP
- Bei V4.1 und höher, TCP/UDP: Verbindungsbeschreibung über die Struktur TCON_IP_V4 am Parameter CONNECT
- Bei V4.1 und höher, ISO-on-TCP: Verbindungsbeschreibung über die Struktur TCON_IP_RFC am Parameter CONNECT
- Bei V4.3 und höher, TCP: Verbindungsbeschreibung über die Struktur TCON_IP_V4_SEC am Parameter CONNECT
- Bei V4.4 und höher, TCP: Verbindungsbeschreibung über die Strukturen TCON_QDN und TCON_QDN_SEC.

Wenn die CPU in den Betriebszustand STOP wechselt, wird eine vorhandene Verbindung beendet und die eingerichtete Verbindung entfernt. Um die Verbindung wieder einzurichten und herzustellen, müssen Sie TRCV_C erneut mit EN_R = 1 ausführen.

Wenn EN_R vor dem Herstellen der Kommunikationsverbindung auf 0 gesetzt wird, wird die Verbindung hergestellt und bleibt bestehen, auch wenn CONT = 0. Es werden jedoch keine Daten empfangen (DONE bleibt 0).

Informationen zur Anzahl der möglichen Kommunikationsverbindungen finden Sie in den technischen Daten Ihrer CPU.

2. Empfangen von Daten über eine vorhandene Kommunikationsverbindung:

Der Datenempfang ist aktiviert, wenn der Parameter EN_R auf den Wert 1 gesetzt ist. Wie oben beschrieben, wird zunächst die Kommunikationsverbindung hergestellt. Die empfangenen Daten werden in einen Empfangsbereich geschrieben. Sie geben die Länge des Empfangsbereichs abhängig von der verwendeten Protokollvariante entweder mit dem Parameter LEN an (wenn LEN <> 0) oder mit der Längenangabe des Parameters DATA (wenn LEN = 0). Wenn Sie am Parameter DATA rein symbolische Werte verwenden, muss der Parameter LEN den Wert 0 haben.

Wird EN_R auf "0" gesetzt, bevor erstmals Daten empfangen werden, bleibt die Kommunikationsverbindung auch dann erhalten, wenn CONT = 0. Es werden jedoch keine Daten empfangen (DONE bleibt "0").

3. Beenden der Kommunikationsverbindung:

Die Kommunikationsverbindung wird nach dem Datenempfang beendet, wenn der Parameter CONT beim Herstellen der Verbindung den Wert 0 hatte. Ansonsten wird die Kommunikationsverbindung aufrechterhalten.

Bei erfolgreicher Ausführung des Empfangsauftrags wird der Parameter DONE auf 1 gesetzt. Die Kommunikationsverbindung kann vorab beendet werden (siehe hierzu die Beschreibung oben über die Abhängigkeit vom Parameter CONT).

TRCV_C wird zurückgesetzt, wenn der Parameter COM_RST gesetzt wird. Werden bei erneuter Ausführung Daten empfangen, kann dies zu Datenverlust führen. Abhängig vom Parameter CONT sind die folgenden Szenarien möglich:

- CONT = "0":
Eine vorhandene Kommunikationsverbindung wird hergestellt.
- CONT = 1 und eine Kommunikationsverbindung wurde hergestellt:
Eine vorhandene Kommunikationsverbindung wird zurückgesetzt und erneut hergestellt.
- CONT = 1 und es wurde keine Kommunikationsverbindung hergestellt:
Es wird keine Kommunikationsverbindung hergestellt.

Der Parameter COM_RST wird nach Auswertung durch die Anweisung TRCV_ zurückgesetzt.

TRCV_C arbeitet mit den gleichen Empfangsmodi wie die Anweisung TRCV. Die folgende Tabelle zeigt, wie Daten in den Empfangsbereich geschrieben werden:

Protokollvariante	Verfügbarkeit der Daten im Empfangsbereich	Parameter Connection_type der Verbindungsbeschreibung	Parameter LEN	Parameter RCVD_LEN
TCP (Ad-hoc-Modus)	Die Daten sind sofort verfügbar.	B#16#11	Ausgewählt mit dem Eingang ADHOC der Anweisung TRCV_C	1 bis 1472
TCP (Datenempfang mit angegebener Länge)	Die Daten sind verfügbar, sobald die am Parameter LEN angegebene Datenlänge vollständig empfangen wurde.	B#16#11	1 bis 8192	Identisch mit dem Wert am Parameter LEN
ISO on TCP (protokollgesteuerte Datenübertragung)	Die Daten sind verfügbar, sobald die am Parameter LEN angegebene Datenlänge vollständig empfangen wurde.	B#16#12	1 bis 8192	Identisch mit dem Wert am Parameter LEN

Hinweis

Ad-hoc-Modus

Den "Ad-hoc-Modus" gibt es nur bei der Protokollvariante TCP. Um die Anweisung TRCV_C für den Ad-hoc-Modus zu konfigurieren, setzen Sie den Eingangsparameter ADHOC der Anweisung. Die Länge des Empfangsbereichs wird vom Pointer am Parameter DATA definiert. Die tatsächlich empfangene Datenlänge wird am Parameter RCVD_LEN ausgegeben. Es können maximal 1460 Bytes empfangen werden.

Hinweis

STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. In der S7-1200 können Sie die Anweisung TRCV_C für den Ad-hoc-Modus konfigurieren, indem Sie den Eingangsparameter ADHOC der Anweisung setzen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

Hinweis**TCP (Datenempfang mit angegebener Länge)**

Mit dem Wert des Parameters LEN geben Sie die Länge für den Datenempfang an. Die am Parameter DATA angegebenen Daten sind im Empfangsbereich verfügbar, sobald die am Parameter LEN angegebene Länge vollständig empfangen wurde.

Hinweis**ISO on TCP (protokollgesteuerte Datenübertragung)**

Bei der Protokollvariante ISO on TCP werden die Daten protokollgesteuert übertragen. Der Empfangsbereich wird von den Parametern LEN und DATA definiert.

Parameter BUSY, DONE und ERROR**Hinweis**

Durch die asynchrone Bearbeitung von TSEND_C müssen die Daten im Sendebereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt.

Für TSEND_C bedeutet der Status WAHR des Parameters DONE, dass die Daten erfolgreich gesendet wurden. Er bedeutet nicht, dass die Partner-CPU den Empfangspuffer tatsächlich liest.

Durch die asynchrone Bearbeitung von TRCV_C sind die Daten im Empfangsbereich nur konsistent, wenn der Parameter DONE = 1 ist.

Tabelle 11-20 Parameter BUSY, DONE und ERROR der Anweisungen TSEND_C und TRCV_C

BUSY	DONE	ERROR	Beschreibung
1	0	0	Der Sendeauftrag wird bearbeitet.
0	1	0	Der Sendeauftrag wurde erfolgreich ausgeführt.
0	0	1	Der Verbindungsaufbau oder der Sendeauftrag wurde mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
0	0	0	Es wurde kein neuer Sendeauftrag zugewiesen.

Den Status der Ausführung können Sie mit den Parametern BUSY, DONE, ERROR und STATUS prüfen. Der Parameter BUSY zeigt den Bearbeitungszustand an. Mit dem Parameter DONE können Sie prüfen, ob ein Sendeauftrag erfolgreich ausgeführt wurde. Der Parameter ERROR wird gesetzt, wenn während der Ausführung von TSEND_C oder TRCV_C Fehler auftraten. Die Fehlerinformationen werden am Parameter STATUS ausgegeben.

Parameter Error und Status

Tabelle 11-21 Bedingungscode von TSEND_C und TRCV_C für ERROR und STATUS

ERROR	STATUS * (W#16#...)	Beschreibung
0	0000	Sendeauftrag (TSEND_C) oder Empfangsauftrag (TRCV_C) fehlerfrei ausgeführt.
0	0001	Kommunikationsverbindung hergestellt.
0	0003	Kommunikationsverbindung geschlossen.
0	7000	Keine Ausführung eines Sendeauftrags aktiv; keine Kommunikationsverbindung hergestellt.
0	7001	<ul style="list-style-type: none"> • Ausführung des Sendeauftrags (TSEND_C) oder Empfangsauftrags (TRCV_C) starten. • Verbindung herstellen. • Auf Verbindungspartner warten.
0	7002	Auftragsausführung (REQ irrelevant)
0	7003	Die Anweisung beendet die Kommunikationsverbindung.
0	7004	Kommunikationsverbindung hergestellt und überwacht; kein Sendeauftrag (TSEND_C) oder Empfangsauftrag (TRCV_C) aktiv.
0	7005	TSEND_C: Datentransfer läuft.
0	7006	TRCV_C: Die Anweisung empfängt die Daten.
1	8085	<ul style="list-style-type: none"> • Der Parameter LEN ist größer als der größte zulässige Wert. • Die Anweisung hat den Wert am Parameter LEN oder DATA nach dem ersten Aufruf geändert.
1	8086	Der Parameter ID im Parameter CONNECT liegt außerhalb des zulässigen Bereichs.
1	8087	Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich.
1	8088	Der Wert am Parameter LEN entspricht nicht dem am Parameter DATA festgelegten Empfangsbereich.
1	8089	<ul style="list-style-type: none"> • Der Parameter CONNECT verweist nicht auf einen Datenbaustein. • Der Parameter CONNECT verweist nicht auf eine Verbindungsbeschreibung. • Die manuell erstellte Verbindungsbeschreibung hat eine falsche Struktur für den ausgewählten Verbindungstyp.
1	8091	Maximale Schachtelungstiefe überschritten.
1	809A	Der Parameter CONNECT zeigt auf ein Feld, das nicht der Länge der Verbindungsbeschreibung entspricht.
1	809B	Die Interfaceld in der Verbindungsbeschreibung entspricht nicht der CPU oder dem CP.
1	80A1	<ul style="list-style-type: none"> • Verbindung oder Port wird verwendet. • Kommunikationsfehler: <ul style="list-style-type: none"> – Die angegebene Verbindung ist noch nicht hergestellt. – Die angegebene Verbindung wird beendet. Übertragung über diese Verbindung ist nicht möglich. – Die Schnittstelle wird neu initialisiert.
1	80A2	Lokaler oder entfernter Port wird vom System benutzt. Weitere Informationen finden Sie unter "Anweisungen TCON und TDISCON" (Seite 657), "Bedingungscode von ERROR und STATUS".

ERROR	STATUS * (W#16#...)	Beschreibung
1	80A3	<ul style="list-style-type: none"> • Es wird versucht, eine bestehende Verbindung erneut herzustellen. • Es wird versucht, eine nicht vorhandene Verbindung zu beenden. • Die verschachtelte Anweisung T_DIAG meldet, dass die Anweisung die Verbindung geschlossen hat.
1	80A4	Die IP-Adresse des entfernten Endpunkts der Verbindung ist ungültig, d.h. sie entspricht der IP-Adresse des lokalen Partners.
1	80A7	Kommunikationsfehler: Sie haben die Anweisung mit COM_RST = 1 aufgerufen, bevor der Sendeauftrag abgeschlossen war.
1	80AA	Ein anderer Baustein stellt eine Verbindung mit derselben Verbindungs-ID her. Wiederholen Sie den Auftrag mit einer neuen steigende Flanke am Parameter REQ.
1	80B3	<ul style="list-style-type: none"> • Bei Verwendung der Protokollvariante UDP enthält der Parameter ADDR keine Daten. • Fehler in der Verbindungsbeschreibung • Eine andere Verbindungsbeschreibung verwendet bereits den lokalen Port.
1	80B4	Es wurden bei der passiven Herstellung einer Verbindung (ActiveEstablished = FALSE) mit der Protokollvariante ISO-on-TCP (ConnectionType = B#16#12) eine oder mehrere der folgenden Bedingungen verletzt: <ul style="list-style-type: none"> • local_tsap_id_len >= B#16#02 • local_tsap_id[1] = B#16#E0
1	80B5	Für Verbindungstyp 13 = UDP ist nur die passive Herstellung einer Verbindung zulässig.
1	80B6	Parametrierungsfehler im Parameter ConnectionType des Datenbausteins für die Verbindungsbeschreibung.
1	80B7	<ul style="list-style-type: none"> • Bei Systemdatentyp TCON_Param: Fehler in einem der folgenden Parameter des Datenbausteins für die Verbindungsbeschreibung: block_length, local_tsap_id_len, rem_subnet_id_len, rem_staddr_len, rem_tsap_id_len, next_staddr_len. • Bei Systemdatentypen TCON_IP_V4, TCON_IP_RFC, TCON_IP_V4_SEC, TCON_QDN, und TCON_QDN_SEC : Die Anweisung hat die IP-Adresse des Partner-Endpunkts auf 0.0.0.0 gesetzt.
1	80C3	<ul style="list-style-type: none"> • Alle Verbindungsressourcen sind belegt. • Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsgruppe bearbeitet.
1	80C4	<p>Temporärer Kommunikationsfehler:</p> <ul style="list-style-type: none"> • Die Anweisung kann die Verbindung derzeit nicht herstellen. • Die Anweisung kann die Verbindung nicht herstellen, weil die Firewalls auf dem Verbindungspfad nicht für die erforderlichen Ports geöffnet sind. • Die Schnittstelle empfängt neue Parameter oder die Anweisung stellt die Verbindung her. • Die Anweisung "TDISCON (Seite 657)" entfernt die konfigurierte Verbindung. • Ein Aufruf mit COM_RST = 1 beendet die verwendete Verbindung. • Temporär stehen am Verbindungspartner keine Empfangsressourcen zur Verfügung. Der Verbindungspartner ist nicht empfangsbereit.
1	80C5	<ul style="list-style-type: none"> • Verbindung vom Kommunikationspartner beendet. • Der entfernte Verbindungspartner hat den LSAP nicht freigegeben.

ERROR	STATUS * (W#16#...)	Beschreibung
1	80C6	Netzwerkfehler: <ul style="list-style-type: none"> • Das lokale Gerät kann den entfernten Partner nicht erreichen. • Physische Unterbrechung am PROFIBUS
1	8722	Fehler im Parameter CONNECT: Ungültiger Quellbereich (Bereich nicht im Datenbaustein deklariert).
1	873A	Fehler im Parameter CONNECT: Der Zugriff auf die Verbindungsbeschreibung ist nicht möglich (kein Zugriff auf den Datenbaustein).
1	877F	Fehler im Parameter CONNECT: Interner Fehler
1	8822	TSEND_C:Parameter DATA: Ungültiger Quellbereich, der Bereich ist im DB nicht vorhanden.
1	8824	TSEND_C:Parameter DATA: Bereichsfehler im Pointer VARIANT.
1	8832	TSEND_C:Parameter DATA: Die DB-Nummer ist zu groß.
1	883A	TSEND_C:Parameter CONNECT: Zugriff auf angegebene Verbindungsdaten nicht möglich (z. B. weil der DB nicht vorhanden ist).
1	887F	TSEND_C:Parameter DATA: Interner Fehler (z. B. ungültiger Verweis auf VARIANT)
1	893A	TSEND_C:Parameter DATA: Zugriff auf Sendebereich nicht möglich (z. B. weil der DB nicht vorhanden ist).
1	8922	TRCV_C:Parameter DATA: Ungültiger Zielbereich, der Bereich ist im DB nicht vorhanden.
1	8924	TRCV_C:Parameter DATA: Bereichsfehler im Pointer VARIANT.
1	8932	TRCV_C:Parameter DATA: Die DB-Nummer ist zu groß.
1	893A	TRCV_C:Parameter CONNECT: Zugriff auf angegebene Verbindungsdaten nicht möglich (z. B. weil der DB nicht vorhanden ist).
1	897F	TRCV_C:Parameter DATA: Interner Fehler (z. B. ungültiger Verweis auf VARIANT).
1	8A3A	TRCV_C:Parameter DATA: Kein Zugriff auf den Datenbereich (z. B. weil der Datenbaustein nicht vorhanden ist).

* Die Fehlercodes können als Ganzzahl- oder Hexadezimalwerte im Programmreditor angezeigt werden.

Hinweis**Fehlermeldungen der Anweisungen TCON, TSEND, TRCV und TDISCON**

Intern verwendet Anweisung TSEND_C die Anweisungen TCON, TSEND und TDISCON und die Anweisung TRCV_C verwendet die Anweisungen TCON, TRCV und TDISCON. Weitere Informationen zu den Fehlermeldungen dieser Anweisungen finden Sie unter TCON, TDISCON, TSEND UND TRCV (Seite 657).

Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND_C und TRCV_C sowie TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

Siehe auch

Verbindungs-IDs für Anweisungen für die offene Benutzerkommunikation (Seite 621)

11.5.8.10 Anweisungen TSEND_C und TRCV_C in älteren Systemen

Vor STEP 7 V13 SP1 und den S7-1200 V4.1-CPU's haben die Anweisungen TSEND_C und TRCV_C nur mit Verbindungsparametern mit Strukturen nach "TCON_Param" funktioniert. Für beide Anweisungsarten gelten die allgemeinen Konzepte. Informationen zur Programmierung finden sich in den Beschreibungen zu den einzelnen TSEND_C- und TRCV_C-Anweisungen.

Version der Anweisungen TSEND_C und TRCV_C auswählen

Es gibt zwei Versionen der Anweisungen TSEND_C und TRCV_C in STEP 7:

- Die Versionen 2.5 und 3.1 waren in STEP 7 Basic/Professional V13 und früher verfügbar.
- Version 4.0 ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Verwenden Sie niemals Anweisungen unterschiedlicher Versionen zusammen in einem CPU-Programm.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.

Open user communication		V3.1
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V2.1
TCON	Establish communication c...	V3.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V3.0
TRCV	Receive data via communic...	V3.0

Um die Version einer TSEND_C- und TRCV_C-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine TSEND_C- oder TRCV_C-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten TSEND_C- oder TRCV_C-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer TSEND_C- oder TRCV_C-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmeditor angezeigten Box. Wählen Sie in der Projektnavigation eine FB- oder FC-Instanz einer TSEND_C- oder TRCV_C-Anweisung aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der TSEND_C- oder TRCV_C-Anweisung anzuzeigen.

Alte Anweisungen TSEND_C und TRCV_C (Daten über Ethernet senden und empfangen)

Die alte Anweisung TSEND_C verbindet die Funktionen der alten Anweisungen TCON, TDISCON und TSEND. Die Anweisung TRCV_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TRCV. (Weitere Informationen zu diesen Anweisungen finden Sie unter "Alte Anweisungen TCON, TDISCON, TSEND und TRCV (TCP-Kommunikation) (Seite 674)".)

Die Mindestdatengröße, die gesendet (TSEND_C) oder empfangen (TRCV_C) werden kann, ist ein Byte; die maximale Datengröße beträgt 8192 Byte. TSEND_C unterstützt nicht die Datenübertragung von Booleschen Adressen und TRCV_C empfängt Daten nicht in Booleschen Adressen. Weitere Informationen zum Übertragen von Daten mit diesen Anweisungen finden Sie im Abschnitt zur Datenkonsistenz (Seite 187).

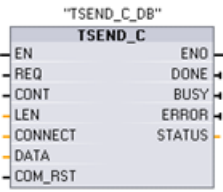
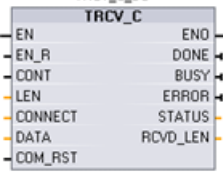
Hinweis

Initialisierung der Kommunikationsparameter

Nachdem Sie die Anweisung TSEND_C oder TRCV_C eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 600) die Kommunikationsparameter (Seite 624). Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 11-22 Anweisungen TSEND_C und TRCV_C

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>TSEND_C stellt eine TCP- oder ISO-on-TCP-Verbindung zu einem Partner her, sendet Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>
	<pre>"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_);</pre>	<p>TRCV_C stellt eine TCP- oder ISO on TCP-Verbindung zu einer CPU her, empfängt Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-23 Datentypen für die Parameter von TSEND_C und TRCV_C

Parameter und Datentyp		Datentyp	Beschreibung
REQ (TSEND_C)	IN	Bool	REQ = 1 startet den Sendeauftrag von TSEND_C bei einer steigenden Flanke über die im Parameter CONNECT beschriebene Verbindung. CONT = 1 ist ebenfalls erforderlich, um die Kommunikationsverbindung herzustellen und aufrechtzuerhalten.
EN_R (TRCV_C)	IN	Bool	Ist EN_R = 1, ist TRCV_C empfangsbereit. Der Empfangsauftrag wird verarbeitet. CONT = 1 ist ebenfalls erforderlich, um die Kommunikationsverbindung herzustellen und aufrechtzuerhalten.
CONT	IN	Bool	Steuert die Kommunikationsverbindung: <ul style="list-style-type: none"> 0: Trennen der Kommunikationsverbindung 1: Aufbauen und Halten der Kommunikationsverbindung Beim Senden von Daten (TSEND_C) (steigende Flanke am Parameter REQ) muss der Parameter CONT den Wert TRUE haben, damit eine Verbindung aufgebaut oder gehalten werden kann. Beim Empfangen von Daten (TRCV_C) (steigende Flanke am Parameter EN_R) muss der Parameter CONT den Wert TRUE haben, damit eine Verbindung aufgebaut oder gehalten werden kann.
LEN	IN	UInt	Maximale Bytezahl, die gesendet (TSEND_C) oder empfangen (TRCV_C) werden soll: <ul style="list-style-type: none"> Standard = 0: Der Parameter DATA legt die Länge der zu sendenden (TSEND_C) oder zu empfangenden (TRCV_C) Daten fest. Ad-hoc-Modus = 65535: Für den Datenempfang ist eine variable Länge eingestellt (TRCV_C).
CONNECT	IN_OUT	TCON_Param	Pointer auf die Verbindungsbeschreibung (Seite 624)
DATA	IN_OUT	Variant	<ul style="list-style-type: none"> Enthält Adresse und Länge der Daten, die gesendet (TSEND_C) werden sollen. Enthält Anfangsadresse und maximale Länge der empfangenen Daten (TRCV_C).
COM_RST	IN_OUT	Bool	Ermöglicht den Neustart der Anweisung: <ul style="list-style-type: none"> 0: Irrelevant 1: Kompletter Neustart des Funktionsbausteins, bestehende Verbindung wird beendet.
DONE	OUT	Bool	<ul style="list-style-type: none"> 0: Auftrag noch nicht gestartet oder läuft noch. 1: Auftrag fehlerfrei ausgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0: Auftrag beendet. 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Kein Fehler 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Fehler- und Statusparameter" unten.)
RCVD_LEN (TRCV_C)	OUT	Int	Tatsächliche empfangene Datenmenge in Byte

Hinweis

Die Anweisung TSEND_C erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

Hinweis

Die Standardeinstellung des Parameters LEN (LEN = 0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Es ist empfehlenswert, dass die von der Anweisung TSEND_C gesendeten Daten die gleiche Größe haben wie der Parameter DATA der Anweisung TRCV_C.

Wenn Sie die Standardeinstellung des Parameters LEN verwenden und die Daten in kleineren Segmenten als die Größe des Parameters DATA gesendet werden sollen, gilt Folgendes. Wenn die Größe der von TSEND_C übertragenen Daten nicht der Größe des Parameters DATA von TRCV_C entspricht, bleibt TRCV_C im Status "Belegt" (Statuscode: 7006), bis die Gesamtgröße der von TSEND_C übertragenen Daten der Größe des Parameters DATA von TRCV_C entspricht.

Der Puffer des Parameters DATA von TRCV_C zeigt die neu empfangenen Daten erst an, wenn die Datengröße gleich der Puffergröße des Parameters DATA ist.

Funktionsweise von TSEND_C

Die folgenden Funktionen beschreiben die Funktionsweise der Anweisung TSEND_C:

- Um eine Verbindung herzustellen, ist TSEND_C mit CONT = 1 auszuführen.
- Nach der erfolgreichen Herstellung der Verbindung setzt TSEND_C den Parameter DONE für einen Zyklus.
- Um die Verbindung zu beenden, führen Sie TSEND_C mit CONT = 0 aus. Die Verbindung wird dann sofort abgebrochen. Dies betrifft auch den Empfänger. Die Verbindung wird beendet und es können Daten im Empfangspuffer verlorengehen.
- Um Daten über eine bestehende Verbindung zu senden, muss TSEND_C bei einer steigenden Flanke an REQ ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND_C den Parameter DONE für einen Zyklus auf 1.
- Um eine Verbindung herzustellen und Daten zu senden, muss TSEND_C mit CONT = 1 und REQ = 1 ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND_C den Parameter DONE für einen Zyklus auf 1.

Funktionsweise von TRCV_C

Die folgenden Funktionen beschreiben die Funktionsweise der Anweisung TRCV_C:

- Um eine Verbindung herzustellen, ist TRCV_C mit CONT = 1 auszuführen.
- Um Daten zu empfangen, führen Sie TRCV_C mit dem Parameter EN_R = 1 aus. TRCV_C empfängt kontinuierlich Daten, wenn die Parameter EN_R = 1 und CONT = 1 sind.
- Um die Verbindung zu beenden, führen Sie TRCV_C mit CONT = 0 aus. Die Verbindung wird sofort beendet und es können Daten verlorengehen.

TRCV_C arbeitet mit den gleichen Empfangsmodi wie die Anweisung TRCV. Die folgende Tabelle zeigt, wie Daten in den Empfangsbereich geschrieben werden:

Tabelle 11-24 Eintragen der Daten in den Empfangsbereich

Protokollvariante	Eingabe der Daten im Empfangsbereich	Parameter "connection_type"	Wert des Parameters LEN	Wert des Parameters RCVD_LEN (Byte)
TCP	Ad-hoc-Modus	B#16#11	65535	1 bis 1472
TCP	Datenempfang mit angegebener Länge	B#16#11	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192
ISO on TCP	Ad-hoc-Modus	B#16#12	65535	1 bis 1472
ISO on TCP	Protokollgesteuert	B#16#12	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192

Hinweis

Ad-hoc-Modus

Den Ad-hoc-Modus gibt es nur bei den Protokollvarianten TCP und ISO on TCP. Den Ad-hoc-Modus stellen Sie ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen. Der Empfangsbereich ist identisch mit dem von Parameter DATA angegebenen Bereich. Die Länge der empfangenen Daten wird am Parameter RCVD_LEN ausgegeben.

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

Hinweis

STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. Bei der S7-1200 stellen Sie den Ad-hoc-Modus ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

Hinweis

Die Daten im Sendebereich müssen konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt.

Durch die asynchrone Bearbeitung von TSEND_C müssen die Daten im Sendebereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt.

Für TSEND_C bedeutet der Status WAHR des Parameters DONE, dass die Daten erfolgreich gesendet wurden. Er bedeutet nicht, dass die Partner-CPU den Empfangspuffer tatsächlich liest.

Durch die asynchrone Bearbeitung von TRCV_C sind die Daten im Empfangsbereich nur konsistent, wenn der Parameter DONE = 1 ist.

Tabelle 11-25 Parameter BUSY, DONE und ERROR der Anweisungen TSEND_C und TRCV_C

BUSY	DONE	ERROR	Beschreibung
WAHR	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSCH	WAHR	FALSCH	Auftrag erfolgreich ausgeführt.
FALSCH	FALSCH	WAHR	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Kein neuer Auftrag zugewiesen.

Parameter Error und Status

Tabelle 11-26 Bedingungscode von TSEND_C und TRCV_C für ERROR und STATUS

ERROR	STATUS	Beschreibung
0	0000	Auftrag fehlerfrei ausgeführt
0	7000	Keine laufende Auftragsbearbeitung
0	7001	Auftragsbearbeitung starten, Verbindung herstellen, auf Verbindungspartner warten
0	7002	Daten werden gesendet oder empfangen
0	7003	Verbindung wird beendet
0	7004	Verbindung hergestellt, wird überwacht, keine laufende Auftragsbearbeitung
1	8085	Der Parameter LEN ist größer als der größte zulässige Wert.
1	8086	Der Parameter CONNECT liegt außerhalb des zulässigen Bereichs.
1	8087	Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich.
1	8088	LEN-Parameter gilt nicht für den in DATA angegebenen Speicherbereich.
1	8089	Der Parameter CONNECT verweist nicht auf einen Datenbaustein.
1	8091	Maximale Schachtelungstiefe überschritten.
1	809A	Der Parameter CONNECT zeigt auf ein Feld, das nicht mit der Länge der Verbindungsbeschreibung übereinstimmt.
1	809B	local_device_id in der Verbindungsbeschreibung stimmt nicht mit der CPU überein.

ERROR	STATUS	Beschreibung
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> Die angegebene Verbindung ist noch nicht hergestellt Die angegebene Verbindung wird derzeit beendet; Datenübertragung über diese Verbindung nicht möglich Schnittstelle wird neu initialisiert
1	80A3	Es wird versucht, eine nicht vorhandene Verbindung zu beenden
1	80A4	Die IP-Adresse des entfernten Teilnehmers ist ungültig. Beispiel: Die IP-Adresse des entfernten Teilnehmers ist dieselbe wie die IP-Adresse des lokalen Teilnehmers.
1	80A5	Verbindungs-ID (Seite 621) wird bereits verwendet.
1	80A7	Kommunikationsfehler: Sie haben TDISCON aufgerufen, bevor TSEND_C beendet war.
1	80B2	Der Parameter CONNECT zeigt auf einen Datenbaustein, der mit dem Schlüsselwort UNLINKED erzeugt wurde.
1	80B3	Inkonsistente Parameter: <ul style="list-style-type: none"> Fehler in der Verbindungsbeschreibung Die lokale Schnittstelle (Parameter local_tsap_id) ist bereits in einer anderen Verbindungsbeschreibung vorhanden. Die ID in der Verbindungsbeschreibung stimmt nicht mit der als Parameter angegebenen ID überein.
1	80B4	Wenn Sie eine passive Verbindung über ISO on TCP (connection_type = B#16#12) aufbauen, werden Sie anhand des Fehlercodes 80B4 gewarnt, wenn der eingegebene TSAP keiner der folgenden Voraussetzungen für Adressen entspricht: <ul style="list-style-type: none"> Hat der lokale TSAP eine Länge von 2 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein. Hat der lokale TSAP eine Länge von 3 oder größer und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein und alle Bytes müssen gültige ASCII-Zeichen sein. Hat der lokale TSAP eine Länge von 3 oder größer und das erste Byte des ID-Werts des TSAP nicht den Wert E0 oder E1 (hexadezimal), müssen alle Bytes der TSAP-ID gültige ASCII-Zeichen sein. Gültige ASCII-Zeichen sind Bytewerte von 20 bis 7E (hexadezimal).
1	80B7	Der Datentyp und/oder die Länge der übertragenen Daten entsprechen nicht dem Bereich in der Partner-CPU, in den die Daten geschrieben werden sollen.
1	80C3	Alle Verbindungsressourcen sind belegt.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> Die Verbindung kann zur Zeit nicht hergestellt werden Die Schnittstelle empfängt neue Parameter Die konfigurierte Verbindung wird von TDISCON entfernt.
1	8722	Parameter CONNECT: Ungültiger Quellbereich: Der Bereich ist im DB nicht vorhanden.
1	873A	Parameter CONNECT: Der Zugriff auf die Verbindungsbeschreibung ist nicht möglich (z. B. DB nicht verfügbar)
1	877F	Parameter CONNECT: Interner Fehler, z. B. ungültiger Verweis auf ANY.
1	893A	Parameter enthält die Nummer eines DB, der nicht geladen ist.

Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND_C und TRCV_C sowie TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

11.5.8.11 Anweisungen TCON, TDISCON, TSEND und TRCV

Ab Version V4.1 der S7-1200 CPU zusammen mit STEP 7 V13 SP1 oder höher ist die Fähigkeit der Anweisung TCON dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4 und TCON_IP_RFC verwendet werden können. Bei der S7-1200 CPU ist ebenso die Fähigkeit der Anweisungen TSEND und TRCV dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4 und TCON_IP_RFC verwendet werden können.

Ab Version V4.3 der S7-1200 CPU zusammen mit STEP 7 V15.1 oder höher ist die Fähigkeit der Anweisung TCON dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC und TCON_IP_RFC verwendet werden können.

Ab Version V4.4 der S7-1200 CPU zusammen mit STEP 7 V13 V16 oder höher ist die Fähigkeit der Anweisung TCON dahingehend erweitert, dass Verbindungsparameter mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN und TCON_QDN_SEC verwendet werden können.

Aus diesem Grund unterstützt die S7-1200 zwei Sätze TCON-, TDISCON-, TSEND- und TRCV-Anweisungen:

- Ältere TCON-, TDISCON-, TSEND- und TRCV-Anweisungen (Seite 674): Die Anweisungen TCON, TDISCON, TSEND und TRCV wurden vor der Version V4.1 der S7-1200 verwendet und funktionieren nur mit Verbindungsparametern mit Strukturen gemäß TCON_Param.
- Anweisungen TCON, TDISCON, TSEND und TRCV (Seite 657): Diese TCON-, TDISCON-, TSEND- und TRCV-Anweisungen stellen alle Funktionen der früheren Anweisungen bereit und bieten zusätzlich die Möglichkeit zur Nutzung von Verbindungsparametern mit Strukturen nach TCON_IP_V4, TCON_IP_V4_SEC, TCON_IP_RFC, TCON_QDN und TCON_QDN_SEC.

Version der Anweisungen TCON, TDISCON, TSEND und TRCV auswählen

Es gibt zwei Versionen der Anweisungen TCON, TDISCON, TSEND oder TRCV in STEP 7:

- Die Versionen 2.5 und 3.1 waren in STEP 7 Basic/Professional V13 und früher verfügbar.
- Version 4.0 ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Verwenden Sie niemals Anweisungen unterschiedlicher Versionen zusammen in einem CPU-Programm.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.

Open user communication		V4.0
TSEND_C	Daten über Ethernet (TCP) senden	V2.5
TRCV_C	Daten über Ethernet (TCP) empfan...	V3.1
TMAIL_C	E-Mail Übertragen	V4.0
Weitere		V4.0
TCON	Kommunikationsverbindung aufb...	V4.0
TDISCON	Kommunikationsverbindung abb...	V2.1
TSEND	Daten über Kommunikationsverbi...	V4.0
TRCV	Daten über Kommunikationsverbi...	V4.0

Um die Version einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine TCON-, TDISCON-, TSEND- oder TRCV-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten TCON-, TDISCON-, TSEND- oder TRCV-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmeditor angezeigten Box. Wählen Sie in der Projektnavigation eine FB- oder FC-Instanz einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der TCON-, TDISCON-, TSEND- oder TRCV-Anweisung anzuzeigen.

Anweisungen TCON, TDISCON, TSEND und TRCV (TCP-Kommunikation)

Ethernet-Kommunikation mit den Protokollen TCP und ISO on TCP

Hinweis

Anweisungen TSEND_C und TRCV_C

Um die Programmierung der PROFINET/Ethernet-Kommunikation zu vereinfachen, verbinden die Anweisungen TSEND_C und TRCV_C die Funktionalität der Anweisungen TCON, TDISCON, TSEND und TRCV:

- TSEND_C verbindet die Anweisungen TCON, TDISCON und TSEND.
- TRCV_C verbindet die Anweisungen TCON, TDISCON und TRCV.

Die folgenden Anweisungen steuern den Kommunikationsprozess:

- TCON baut die TCP/IP-Verbindung zwischen dem Client- und dem Server-PC (CPU) auf.
- TSEND und TRCV senden und empfangen Daten.
- TDISCON trennt die Verbindung.

Die Mindestdatengröße, die gesendet (TSEND) oder empfangen (TRCV) werden kann, ist ein Byte; die maximale Datengröße beträgt 8192 Byte. TSEND unterstützt nicht die Datenübertragung von Booleschen Adressen und TRCV empfängt Daten nicht in Booleschen Adressen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 187).

TCON, TDISCON, TSEND und TRCV laufen asynchron ab, so dass die Bearbeitung eines Auftrags mehrere Anweisungsausführungen umfasst. Beispielsweise starten Sie einen Auftrag zum Einrichten und Herstellen einer Verbindung durch die Ausführung von TCON mit Parameter REQ = 1. Dann überwachen Sie den Stand der Auftragsausführung, indem TCON wiederholt ausgeführt wird, und prüfen, ob der Parameter DONE den Abschluss des Auftrags bestätigt.

Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE und ERROR. Den aktuellen Status eines Auftrags können Sie anhand der Tabelle feststellen:

Tabelle 11-27 Interaktionen zwischen den Parametern BUSY, DONE und ERROR

BUSY	DONE	ERROR	Beschreibung
1	0	0	Auftrag wird bearbeitet.
0	1	0	Auftrag erfolgreich ausgeführt.
0	0	1	Auftrag mit Fehler beendet. Die Fehlerursache wird am Parameter STATUS ausgegeben.
0	0	0	Kein neuer Auftrag zugewiesen.

TCON und TDISCON

Hinweis

Initialisierung der Kommunikationsparameter

Nachdem Sie die Anweisung TCON eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 600) die Kommunikationsparameter (Seite 624). Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den Instanz-DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 11-28 Anweisungen TCON und TDISCON

KOP/FUP		Beschreibung
	<pre>"TCON_DB" (req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:=_struct_inout_);</pre>	TCP und ISO on TCP: TCON initiiert eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.
	<pre>"TDISCON_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	TCP und ISO on TCP: TDISCON beendet eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-29 Datentypen für die Parameter von TCON und TDISCON

Parameter	Deklaration	Datentyp	Beschreibung
REQ	IN	Bool	Startet den Auftrag für den Aufbau der in der ID angegebenen Verbindung bei einer steigenden Flanke.
ID	IN	CONN_OUC (Word)	Verweis auf die zugewiesene Verbindung. Wertebereich: W#16#0001 bis W#16#0FFF
CONNECT (TCON)	IN_OUT	VARIANT	Pointer auf die Verbindungsbeschreibung: <ul style="list-style-type: none"> Bei TCP oder UDP verwenden Sie die Struktur TCON_IP_v4 oder TCON_QDN. Zur Beschreibung siehe: "Parameter für die PROFINET-Verbindung (Seite 624)". Bei TCP mit sicherer Kommunikation verwenden Sie die Struktur TCON_IP_V4_SEC oder TCON_QDN_SEC. Zur Beschreibung siehe: "Parameter für die PROFINET- (Seite 624)Verbindung (Seite 624)". Bei ISO-on-TCP verwenden Sie die Struktur TCON_IP_RFC. Zur Beschreibung siehe: "Parameter für die PROFINET-Verbindung (Seite 624)". Bei ISO-Verbindungen des CP 1543-1 / CP 1545-1 verwenden Sie die Struktur TCON_ISOnative. Zur Beschreibung siehe: Online-Hilfe im TIA Portal: "Struktur der Verbindungsbeschreibung nach TCON_ISOnative". Bei Verbindungen mit SMS-Clients verwenden Sie den Systemdatentyp TCON_PHONE. Zur Beschreibung siehe die Online-Hilfe im TIA Portal: "Verbindungsparameter nach TCON_Phone". Bei FDL-Verbindungen des CM 1542-5 verwenden Sie den Systemdatentyp TCON_FDL. Siehe hierzu die Online-Hilfe im TIA Portal: "Verbindungsparameter nach TCON_FDL".
DONE	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Auftrag wurde noch nicht gestartet oder wird noch ausgeführt. 1: Auftrag fehlerfrei ausgeführt.
BUSY	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Auftrag wurde noch nicht gestartet oder ist bereits abgeschlossen. 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.
ERROR	OUT	Bool	Statusparameter ERROR: <ul style="list-style-type: none"> 0: Kein Fehler 1: Fehler aufgetreten
STATUS	OUT	Word	Zustand der Anweisung

Beide Kommunikationsteilnehmer führen die Anweisung TCON aus, um die Kommunikationsverbindung einzurichten und herzustellen. Mit Parametern geben Sie die aktiven und passiven Endpunkte der Kommunikation an. Nach dem Einrichten und Aufbauen der Verbindung wird sie automatisch von der CPU gehalten und überwacht.

Wird die Verbindung durch eine Leitungsstörung oder vom entfernten Teilnehmer abgebrochen, so versucht der aktive Teilnehmer, sie wieder aufzubauen. Sie müssen TCON nicht erneut ausführen.

Wird die Anweisung TDISCON ausgeführt oder ist die CPU in den Betriebszustand STOP gegangen, so wird die bestehende Verbindung beendet und die eingerichtete Verbindung gelöscht. Um die Verbindung einzurichten und wiederherzustellen, müssen Sie TCON erneut ausführen.

Tabelle 11-30 Bedingungs-codes von ERROR und STATUS für TCON und TDISCON

ERROR	STATUS * (W#16#...)	Erklärung
0	0000	Verbindung erfolgreich aufgebaut.
0	7000	Keine laufende Auftragsbearbeitung
0	7001	Auftragsbearbeitung starten; Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON).
0	7002	Die Anweisung stellt eine Verbindung her (REQ irrelevant); Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON).
1	8085	TCON: Verwendungs-ID wird verwendet.
1	8086	TCON: Der Parameter ID liegt außerhalb des gültigen Bereichs.
1	8087	TCON: Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich
1	8089	TCON: Der Parameter CONNECT verweist nicht auf eine Verbindungsbeschreibung oder die Verbindungsbeschreibung wurde manuell erstellt.
1	809A	TCON: Die Anweisung unterstützt nicht die Struktur am Parameter CONNECT, oder die Länge ist ungültig.
1	809B	TCON: <ul style="list-style-type: none"> Das Element Interfaceld in der Verbindungsbeschreibung entspricht nicht der CPU oder dem CP, oder es ist "0". Das Element Interfaceld in der Struktur TCON_XXX verweist nicht auf eine Hardwarekennung einer CPU- oder CM/CP-Schnittstelle.
1	80A1	TCON: Bei TCP/UDP: Verbindung oder Port wird verwendet.
1	80A2	TCON: Das System verwendet den lokalen oder entfernten Port. Weitere Informationen finden Sie unter "Gemeinsame Parameter für Anweisungen" (Seite 727), "Eingeschränkte TSAPs und Portnummern für die passive ISO- und TCP-Kommunikation".
1	80A3	TCON: Eine vom Anwenderprogramm erstellte Verbindung (TCON) verwendet den Wert am Parameter ID. Die Verbindung verwendet die gleiche ID und die gleichen Verbindungseinstellungen am Parameter CONNECT.
1	80A4	TCON: Die IP-Adresse des entfernten Endpunkts der Verbindung ist ungültig oder entspricht der IP-Adresse des lokalen Partners.
1	80A7	TCON: Kommunikationsfehler: Sie haben "TDISCON" ausgeführt, bevor "TCON" beendet war.
1	80B3	Inkonsistente Parametrierung
1	80B4	TCON: Nur mit TCON_IP_RFC: Eine der folgenden Situationen ist aufgetreten: <ul style="list-style-type: none"> Die Anweisung hat die lokale Transportauswahl nicht zugeordnet. Das erste Byte enthält nicht den Wert 0x0E. Die lokale Transportauswahl beginnt mit "SIMATIC".
1	80B5	TCON: Die Anweisung gestattet nur die passive Herstellung einer Verbindung für Verbindungstyp 13 = UDP (Parameter ActiveEstablished der Struktur TCON_XXX hat den Wert TRUE).

ERROR	STATUS * (W#16#...)	Erklärung
1	80B6	TCON: Parametrierungsfehler im Parameter ConnectionType des Datenbausteins für die Verbindungsbeschreibung: <ul style="list-style-type: none"> Nur gültig für TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: 0x11, 0x0B und 0x13 Nur gültig für TCON_IP_RFC: 0x0C und 0x12
1	80B7	TCON: Bei TCON_IP_V4, TCON_IP_V4_SEC, TCON_QDN, TCON_QDN_SEC: <ul style="list-style-type: none"> TCP (aktiver Verbindungsaufbau) Entfernter Port ist "0". TCP (passiver Verbindungsaufbau): Lokaler Port ist "0". UDP: Lokaler Port ist "0". Die Anweisung hat die IP-Adresse des Partner-Endpunkts auf 0.0.0.0 gesetzt. TCON: Bei TCON_IP_RFC: <ul style="list-style-type: none"> Die Anweisung hat die lokale (local_selector) oder entfernte (remote_selector) Transportauswahl mit einer Länge von mehr als 32 Bytes zugeordnet. Für TSELlength der Transportauswahl (lokal oder entfernt) hat die Anweisung eine Länge über 32 zugeordnet. Längenfehler in der IP-Adresse des spezifischen Verbindungspartners Die Anweisung hat die IP-Adresse des Partner-Endpunkts auf 0.0.0.0 gesetzt.
1	80B8	TCON: Der Parameter ID in der lokalen Verbindungsbeschreibung (Struktur am Parameter CONNECT) und der Parameter ID der Anweisung sind verschieden.
1	80C3	TCON: Alle Verbindungsressourcen sind belegt.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> Die Anweisung kann die Verbindung derzeit nicht herstellen (TCON). Die Anweisung kann die Verbindung nicht herstellen, weil die Firewalls auf dem Verbindungspfad nicht für die erforderlichen Ports geöffnet sind (TCON). Die Schnittstelle empfängt neue Parameter (TCON und TDISCON). Die Anweisung "TDISCON" entfernt die konfigurierte Verbindung (TCON).
1	80C5	TCON: Der entfernte Partner hat eine der folgenden Aktionen vorgenommen: <ul style="list-style-type: none"> Sich geweigert, die Verbindung herzustellen Die Verbindung beendet Aktiv die Verbindung beendet
1	80C6	TCON: Die Anweisung kann den entfernten Partner nicht erreichen (Netzwerkfehler).
1	80C7	TCON: Zeitüberschreitung während der Ausführung
1	80C8	TCON: Eine vom Anwenderprogramm erstellte Verbindung (TCON) verwendet den Wert am Parameter ID. Die Verbindung verwendet die gleiche ID, jedoch abweichende Verbindungseinstellungen am Parameter CONNECT.
1	80C9	TCON: Bestätigung des entfernten Partners fehlgeschlagen. Der entfernte Partner, der die Verbindung aufbauen will, stimmt nicht mit dem Partner überein, der in der Struktur von Parameter CONNECT festgelegt ist.
1	80CE	TCON: Die IP-Adresse der lokalen Schnittstelle lautet 0.0.0.0.
1	80E0	TCON: Die Anweisung hat eine ungeeignete oder schlechte Meldung empfangen.


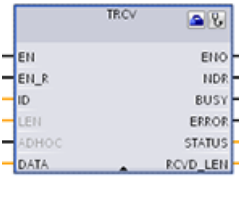
ERROR	STATUS * (W#16#...)	Erklärung
1	80E1	TCON: Fehler beim Handshake. Mögliche Ursachen: <ul style="list-style-type: none"> • Abbruch durch den Anwender • Sicherheit ist nicht hoch genug. • Die Anweisung unterstützt die erneute Negotiation nicht. • Die Anweisung unterstützt die Version SSL/TLS nicht. • Validierung des Hostnamens fehlgeschlagen.
1	80E2	Zertifikat nicht unterstützt / Zertifikat ungültig / kein Zertifikat Mögliche Ursache: Die CPU hat bei dem betroffenen Modul die Uhrzeit nicht eingestellt oder das Modul nicht synchronisiert. Beispiel: Die Standardeinstellung für das Datum des Moduls ist 1/1/2012, und die CPU hat das Datum während der Inbetriebnahme nicht eingestellt. Der Gültigkeitszeitraum des Zertifikats beginnt am 20. August 2016 und endet am 20. August 2024. In diesem Fall liegt das Datum des Moduls außerhalb des Gültigkeitszeitraums des Zertifikats, somit ist das Zertifikat für das Modul ungültig.
1	80E3	Zertifikat verworfen.
1	80E4	Keine gültige Zertifizierungsstelle gefunden.
1	80E5	Zertifikat abgelaufen.
1	80E6	Integritätsfehler im Transport Layer Security Protocol
1	80E7	Nicht unterstützte Erweiterung im Zertifikat X.509-V3
1	80E9	Die Anweisung unterstützt einen TLS-Server nicht ohne Serverzertifikat.
1	80EA	Die Anweisung unterstützt das Protokoll DTLS (UDP) nicht.
1	80EB	Ein Client kann kein Client-Zertifikat anfordern.
1	80EC	Der Server kann die Validierung nicht basierend auf dem subjectAlternateName durchführen (das können nur Clients).
1	80ED	TLSServerCertRef_m-ID ungültig
* Die Fehlercodes können im Programmeditor als Ganzzahl- oder Hexadezimalwerte angezeigt werden.		

TSEND und TRCV

Hinweis

Wenn Sie in der offenen Benutzerkommunikation über PROFINET eine Anweisung TSEND ohne die entsprechende Anweisung TRCV auf dem Remote-Gerät ausführen, kann es passieren, dass die Anweisung TSEND auf unbestimmte Zeit im Zustand "Busy" bleibt und darauf wartet, dass die Anweisung TRCV die Daten empfängt. In diesem Zustand ist der Ausgang "Busy" der Anweisung TSEND gesetzt und der Ausgang "Status" hat den Wert "0x7002". Diese Bedingung kann auftreten, wenn Sie mehr als 4096 Byte an Daten übertragen. Das Problem wird bei der nächsten Ausführung der Anweisung TRCV behoben.

Tabelle 11-31 Anweisungen TSEND und TRCV

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	TCP und ISO on TCP: TSEND sendet über eine Kommunikationsverbindung Daten von der CPU zu einer Partnerstation.
	<pre>"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_);</pre>	TCP und ISO on TCP: TRCV empfängt über eine Kommunikationsverbindung Daten von einer Partnerstation auf der CPU.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-32 Datentypen für die Parameter von TSEND und TRCV

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
EN_R	IN	Bool
ID	IN	CONN_OUC (Word)
LEN	IN	UDInt
ADHOC	IN	Bool
DATA	IN_OUT	Variant

Detailed description of the table content:

- REQ**: TSEND: Startet den Sendeauftrag bei einer steigenden Flanke. Die Daten werden aus dem mit DATA und LEN bezeichneten Bereich übertragen.
- EN_R**: TRCV: Aktiviert die CPU für den Empfang, bei EN_R = 1 ist TRCV empfangsbereit. Der Empfangsauftrag wird bearbeitet.
- ID**: Dies ist ein Verweis auf die zugehörige Verbindung. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen. Wertebereich: W#16#0001 bis W#16#0FFF
- LEN**: Maximale Bytezahl, die gesendet (TSEND) oder empfangen (TRCV) werden soll:
 - Standard = 0: Der Parameter DATA legt die Länge der zu sendenden (TSEND) oder zu empfangenden (TRCV) Daten fest.
 - Ad-hoc-Modus = 65535: Für den Datenempfang ist eine variable Länge eingestellt (TRCV).
- ADHOC**: TRCV: Optionaler Parameter (versteckt) Ad-hoc-Modus-Anforderung für Verbindungstyp TCP.
- DATA**: Pointer auf den Datenbereich für Senden (TSEND) oder Empfangen (TRCV). Der Datenbereich enthält die Adresse und Länge. Die Adresse bezieht sich auf einen der Speicherbereiche E, A oder M oder auf einen DB.

Parameter und Datentyp		Datentyp	Beschreibung
DONE	OUT	Bool	TSEND: <ul style="list-style-type: none"> • 0: Auftrag noch nicht gestartet oder läuft noch. • 1: Auftrag fehlerfrei ausgeführt.
NDR	OUT	Bool	TRCV: <ul style="list-style-type: none"> • NDR = 0: Auftrag noch nicht gestartet oder läuft noch. • NDR = 1: Der Auftrag wurde erfolgreich durchgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1: Der Auftrag ist noch nicht beendet. Es kann kein neuer Auftrag gestartet werden. • BUSY = 0: Auftrag beendet.
ERROR	OUT	Bool	ERROR = 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für Error und Status" unten.)
RCVD_LEN	OUT	UDInt	TRCV: Tatsächliche empfangene Datenmenge in Byte

Hinweis

Die Anweisung TSEND erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

Funktionsweise von TRCV

Die Anweisung TRCV schreibt die empfangenen Daten in einen Empfangsbereich, der von den folgenden zwei Variablen angegeben wird:

- Zeigt auf den Anfang des Bereichs
- Länge des Bereichs oder der Wert, der am Eingang LEN bereitgestellt wird, sofern er nicht 0 ist

Hinweis

Die Standardeinstellung des Parameters LEN (LEN = 0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Es ist empfehlenswert, dass die von der Anweisung TSEND gesendeten Daten die gleiche Größe haben wie der Parameter DATA der Anweisung TRCV.

Wenn Sie die Standardeinstellung des Parameters LEN verwenden und die Daten in kleineren Segmenten als die Größe des Parameters DATA gesendet werden sollen, gilt Folgendes. Es ist empfehlenswert, das Bit EN_R auf 1 zu halten, bis die entsprechende Anweisung TSEND die entsprechende Menge an Daten überträgt, um den Parameter DATA von TRCV zu füllen. Wenn die Größe der von TSEND übertragenen Daten nicht der Größe des Parameters DATA von TRCV entspricht, bleibt TRCV im Status "Belegt" (Statuscode: 7002), solange das Bit EN_R = 1 ist, bis die Gesamtgröße der von TSEND übertragenen Daten der Größe des Parameters DATA von TRCV entspricht. Wenn das Bit EN_R von TRCV Impulse empfängt, muss es die gleiche Anzahl von Impulsen erhalten, wie die Anweisung TSEND ausgeführt wird, um Daten zu empfangen.

Der Puffer des Parameters DATA von TRCV zeigt die neu empfangenen Daten erst an, wenn die Datengröße gleich der Puffergröße des Parameters DATA ist.

Sobald alle Auftragsdaten empfangen sind, werden sie von TRCV in den Empfangsbereich übertragen und NDR wird auf 1 gesetzt.

Tabelle 11-33 Eintragen der Daten
in den Empfangsbereich

Protokollvariante	Eingabe der Daten im Empfangsbereich	Parameter "connection_type"	Wert des Parameters LEN	Wert des Parameters RCVD_LEN (Byte)
TCP	Ad-hoc-Modus	B#16#11	Ausgewählt mit dem Eingang ADHOC der Anweisung TRCV	1 bis 1472
TCP	Datenempfang mit angegebener Länge	B#16#11	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192
ISO on TCP	Ad-hoc-Modus	B#16#12	65535	1 bis 1472
ISO on TCP	protokollgesteuert	B#16#12	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192

Hinweis

Ad-hoc-Modus

Den Ad-hoc-Modus gibt es nur bei den Protokollvarianten TCP und ISO on TCP. Um die Anweisung TRCV für den Ad-hoc-Modus zu konfigurieren, setzen Sie den Eingangsparameter ADHOC der Anweisung. Der Empfangsbereich ist identisch mit dem von Parameter DATA angegebenen Bereich. Die Länge der empfangenen Daten wird am Parameter RCVD_LEN ausgegeben. Unmittelbar nach dem Empfang eines Datenblocks schreibt TRCV die Daten in den Empfangsbereich und setzt NDR auf 1.

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

Hinweis

STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. In S7-1200 können Sie die Anweisung TRCV für den Ad-hoc-Modus konfigurieren, indem Sie den Eingangsparameter ADHOC der Anweisung setzen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

Tabelle 11-34 Bedingungs-codes von ERROR und STATUS für TSEND und TRCV

ERROR	STATUS	Beschreibung
0	0000	<ul style="list-style-type: none"> • Sendeauftrag fehlerfrei ausgeführt (TSEND) • Neue Daten angenommen: Die aktuelle Länge der empfangenen Daten wird in RCVD_LEN (TRCV) gezeigt.
0	7000	<ul style="list-style-type: none"> • Keine laufende Auftragsbearbeitung (TSEND) • Baustein nicht empfangsbereit (TRCV)
0	7001	<ul style="list-style-type: none"> • Auftragsbearbeitung starten, Daten werden gesendet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND). • Baustein empfangsbereit, Empfangsauftrag wurde aktiviert (TRCV).
0	7002	<ul style="list-style-type: none"> • Folgeaufruf (REQ irrelevant), Auftrag wird bearbeitet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND). • Folgeaufruf, Empfangsauftrag wird bearbeitet: Während dieser Bearbeitung werden die Daten in den Empfangsbereich geschrieben. Deshalb kann ein Fehler zu inkonsistenten Daten im Empfangsbereich führen (TRCV).
1	8085	<ul style="list-style-type: none"> • Der Parameter LEN ist größer als der größte zulässige Wert (TSEND) und (TRCV). • Der Parameter LEN oder DATA hat sich seit der ersten Ausführung der Anweisung geändert (TRCV).
1	8086	Der Parameter ID ist nicht im zulässigen Adressbereich.
1	8088	Der Parameter LEN ist größer als der in DATA festgelegte Speicherbereich.

ERROR	STATUS	Beschreibung
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> • Die angegebene Verbindung ist noch nicht hergestellt (TSEND und TRCV). • Die angegebene Verbindung wird beendet. Ein Sende- oder Empfangsauftrag ist über diese Verbindung nicht möglich (TSEND und TRCV). • Die Schnittstelle wird neu initialisiert (TSEND). • Die Schnittstelle empfängt neue Parameter (TRCV).
1	80C3	Interner Mangel an Verbindungsressourcen (Seite 621): Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsklasse bearbeitet.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> • Die Verbindung zum Kommunikationspartner kann zur Zeit nicht hergestellt werden. • Die Schnittstelle empfängt neue Parametrierungen oder die Verbindung wird gerade aufgebaut.

Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND_C, TRCV_C, TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

11.5.8.12 TCONSettings

Ab V4.5 und TIA Portal V17 können Sie mit der Anweisung "TCONSettings" die folgenden Funktionen ausführen:

- Eine Verbindungs-ID für eine neue OUC-Verbindung anfordern
- Eine Verbindungs-ID für eine neue OUC-Verbindung anfordern und gleichzeitig für diese Verbindung eine Eigenschaft angeben
- Eine Eigenschaft einer vorbereiteten oder einer vorhandenen OUC-Verbindung lesen
- Eine Eigenschaft einer vorbereiteten oder einer vorhandenen OUC-Verbindung angeben

Sie können mit der Anweisung "TCONSettings" die folgenden Verbindungseigenschaften lesen oder angeben:

- Wie eine TCP-Verbindung beendet wird

Die Anweisung "TCONSettings" ist eine asynchrone Anweisung. Die Verarbeitung erstreckt sich über mehrere Aufrufe. Die Verarbeitung beginnt mit einer steigenden Flanke am Parameter "REQ".

Die Parameter "Busy" und "Done" geben den Auftragsstatus an.

Tritt während der Ausführung ein Fehler auf, wird dies von den Parametern "Error" und "Status" angezeigt.

Tabelle 11-35 Datentypen für die Parameter von TCONSettings

Parameter und Typ		Datentyp	Beschreibung
REQ	Eingang	Bool	Steuerparameter REQUEST Aktiviert den Auftrag bei einer steigenden Flanke.
MODE	Eingang	USInt	Verwendet den Parameter "Mode", um die Informationen auszuwählen, die Sie aus Ihrer CPU auslesen möchten: <ul style="list-style-type: none"> • 0: Eine Verbindungs-ID für eine neue OUC-Verbindung anfordern und bei Bedarf eine Eigenschaft für diese Verbindung angeben (wenn ein gültiger Wert einer Eigenschaft im Parameter OPTION vorhanden ist) • 1: Lesen einer Eigenschaft der OUC-Verbindung, auf die mittels ID verwiesen wird • 2: Angeben einer Eigenschaft der OUC-Verbindung, auf die mittels ID verwiesen wird • 3 bis 255: Reserviert
DONE	Ausgang	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Auftrag wurde noch nicht gestartet oder wird noch ausgeführt. • 1: Auftrag fehlerfrei ausgeführt. Dieser Zustand wird nur einen Aufruf lang angezeigt.
BUSY	Ausgang	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Auftrag wurde noch nicht gestartet oder ist bereits abgeschlossen. • 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag mit dieser Instanz gestartet werden.
ERROR	Ausgang	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> • 0: Kein Fehler. • 1: Fehler bei der Bearbeitung aufgetreten. STATUS enthält ausführliche Informationen über die Fehlerart. Dieser Zustand wird nur einen Aufruf lang angezeigt.
STATUS	Ausgang	Wort	Rückgabewert oder Fehlerinformationen der Anweisung "TCONSettings".
ID	InOut	CONN_OUC	Verweis auf die Verbindung: Hinweis: Bei MODE = 0 ist ID ein Ausgangsparameter, bei MODE = 1 und MODE = 2 ist ID ein Eingangsparameter.
OPTION	InOut	Variante	Zeiger auf die Beschreibung der zu lesenden oder anzugebenden Verbindungseigenschaft: <ul style="list-style-type: none"> • TCON_TCPTermination: Wie die TCP-Verbindung beendet werden soll.

Parameter BUSY, DONE und ERROR

Den Status des Auftrags können Sie mit den Parametern BUSY, DONE, ERROR und STATUS prüfen. Der Parameter BUSY zeigt den Bearbeitungszustand an. Mit dem Parameter DONE können Sie prüfen, ob ein Auftrag erfolgreich ausgeführt wurde. Der Parameter ERROR wird gesetzt, wenn während der Ausführung von "TCONSettings" Fehler auftreten. Die Fehlerinformationen werden am Parameter STATUS ausgegeben.

Die folgende Tabelle zeigt die Beziehung zwischen den Parametern BUSY, DONE und ERROR:

BUSY	DONE	ERROR	Beschreibung
1	0	0	Auftrag wird bearbeitet.
0	1	0	Der Auftrag wurde erfolgreich ausgeführt.
0	0	1	Auftrag mit Fehler beendet. Die Fehlerursache wird im Parameter STATUS ausgegeben.
0	0	0	Es wurde kein neuer Auftrag zugewiesen.

Tabelle 11-36 Bedingungscode von TCONSettings für Status

STATUS (W#16#...)	Bedeutung
0000	Die Ausführung von "TCONSettings" wurde fehlerfrei beendet.
7000	Keine laufende Auftragsbearbeitung.
7001	Start der Auftragsausführung
7002	Zwischenzeitlicher Aufruf (REQ irrelevant):
8086	Die ID liegt außerhalb des zulässigen Bereichs.
8087	Maximale Anzahl OUC-Verbindungen erreicht; keine weitere Verbindung möglich.
8089	OPTION zeigt nicht auf einen gültigen Datentyp oder OPTION ist leer.
8090	OPTION zeigt auf eine Verbindungseigenschaft, die für die von ID angegebene Verbindung nicht geändert werden darf.
8091	Ungültiger Wert für MODE
8092	Ein Wert im von OPTION referenzierten Datenbaustein ist nicht zulässig
8093	Wenn MODE den Wert 0 hat, muss ID auch den Wert 0 haben.
809A	OPTION zeigt auf einen Datentyp, der für "TCONSettings" nicht zulässig ist.
80A3	ID zeigt auf einen nicht vorhandenen Kommunikationsendpunkt.
80B1	Der Parameter OPTION wurde geändert, bevor die Ausführung von "TCONSettings" beendet war. OPTION darf während der Ausführung von "TCONSettings" nicht geändert werden.
80C3	Die maximale Anzahl gleichzeitig aktiver Aufträge ist bereits erreicht.

Maximale Anzahl gleichzeitig aktiver Aufträge

Die maximale Anzahl gleichzeitig aktiver Aufträge ist mit der maximalen Anzahl von OUC-Verbindungen Ihrer CPU identisch.

Siehe auch

Neue Funktionen (Seite 35)

Verbindungsressource reservieren

Rufen Sie TCONSettings mit MODE = 0 auf. Weisen Sie die relevanten Parameter wie folgt zu:

- Geben Sie am Parameter ID den Wert NULL ein.
- Wenn Sie für die zugehörige Verbindung keine Eigenschaft angeben möchten, lassen Sie den Parameter OPTION leer.
Wenn Sie für die zugehörige Verbindung eine Eigenschaft angeben möchten, weisen Sie dem Parameter OPTION (Seite 667) einen gültigen Wert zu.

Sobald der Parameter DONE von TCONSettings nach WAHR wechselt, ist am Parameter ID eine Verbindungs-ID für eine neue OUC-Verbindung verfügbar. Wenn Sie im Parameter OPTION eine Eigenschaft angegeben haben, verwendet die Verbindung diese Eigenschaft für die Verbindung. Die Anweisung TCONSettings nutzt eine OUC-Verbindungsressource für die ID und erstellt die entsprechenden Diagnoseobjekte. Die Anweisung TCONSettings hat die Verbindung vorbereitet, doch externe Kommunikationspartner wissen noch nichts über die Verbindung.

Sie haben keine Details für die Verbindung angegeben, weder den Verbindungspartner, noch das Protokoll, noch die Schnittstelle oder den DB mit der Verbindungsbeschreibung.

Hinweis

Verbindung herstellen

TCONSettings stellt keine Verbindung her.

Die zugehörige Verbindung herstellen

Wenn Sie die entsprechende Verbindung nach der erfolgreichen Ausführung von "TCONSettings" herstellen möchten, gehen Sie wie folgt vor:

1. Speichern Sie die von "TCONSettings" vorgegebene Verbindungs-ID.
2. Rufen Sie die Anweisung "TCON" mit genau dieser ID auf.

Die Anzahl der verfügbaren OUC-Verbindungen ändert sich nicht, weil die Anweisung TCONSettings die Verbindung bereits konsumiert hat.

Verbindungs-ID und die entsprechende Verbindungsressource aktivieren

Wenn Sie die von "TCONSettings" angegebene Verbindungs-ID und die entsprechende Verbindungsressource erneut aktivieren möchten, rufen Sie die Anweisung "TDISCON" mit genau dieser ID auf.

CPU geht in den Betriebszustand STOP

Wenn die CPU in den Betriebszustand STOP geht, werden alle von "TCONSettings" angegebenen Verbindungs-IDs und die entsprechenden Verbindungsressourcen wieder aktiviert.

Eine Eigenschaft einer vorbereiteten oder einer vorhandenen Verbindung lesen

Rufen Sie "TCONSettings" mit MODE = 1 auf. Weisen Sie die relevanten Parameter wie folgt zu:

- Geben Sie am Parameter ID die Referenz zur gewünschten Verbindung an.
- Geben Sie am Parameter OPTION die Verbindungseigenschaft an, die Sie lesen möchten.

Nachdem der Parameter DONE den Wert WAHR angenommen hat, sind die aktuellen Werte der gewünschten Eigenschaft im von OPTION angegebenen Datenbereich verfügbar.

Eine Eigenschaft einer vorbereiteten oder einer vorhandenen Verbindung angeben

Rufen Sie "TCONSettings" mit MODE = 2 auf. Weisen Sie die relevanten Parameter wie folgt zu:

- Geben Sie am Parameter ID die Referenz zu der Verbindung an, der Sie eine Eigenschaft zuweisen möchten.
- Geben Sie am Parameter OPTION die Verbindungseigenschaft an, die Sie vorgeben möchten.

Nachdem der Parameter DONE den Wert WAHR angenommen hat, wurde der Verbindung die gewünschte Eigenschaft zugewiesen.

Von OUC- und Modbus-Anweisungen erstellte Verbindungen

Die Anweisungen der OUC-Bibliothek, die auf "_C" enden, und die Anweisungen der MODBUS-TCP-Bibliothek stellen Verbindungen her, indem intern die Anweisung "TCON" aufgerufen wird. Sie können solche Verbindungen mit "TCONSettings" genauso ändern wie Verbindungen, die Sie durch explizites Aufrufen von "TCON" erstellt haben.

Verbindungseigenschaften, die generell gelesen oder angegeben werden können

Die folgenden Verbindungseigenschaften können mit der Anweisung "TCONSettings" gelesen oder angegeben werden:

- Wie eine TCP-Verbindung beendet wird

Beziehung zwischen dem Protokoll oder der Schnittstelle und den tatsächlich lesbaren oder angebbaren Verbindungseigenschaften

Nicht jedes Protokoll und nicht jede Schnittstelle kann alle oben genannten Verbindungseigenschaften lesen oder angeben. In der folgenden Tabelle wird gezeigt, welche Verbindungseigenschaften für die einzelnen Protokolle bzw. Schnittstellen möglich sind.

Protokoll /Schnittstelle	Verbindung beenden
Programmierte Verbindung:	Ja
Konfigurierte Verbindung:	Nein ¹⁾
TCP	Ja
UDP	Nein ³⁾
ISO on TCP	Nein
CPU-Schnittstelle	Ja
CP-Schnittstelle	Nein

Protokoll /Schnittstelle	Verbindung beenden
Virtuelle CP-Schnittstelle	Ja
¹⁾ Sie können "TDISCON" nicht für eine konfigurierte Verbindung aufrufen. Deshalb gibt es keine Möglichkeit, die Verbindung ordnungsgemäß zu beenden. ²⁾ UDP ist auf Protokollebene verbindungslos, deshalb ist keine Beendigung notwendig.	

Konflikte beim Angeben von Verbindungseigenschaften

Jede vordefinierbare Verbindungseigenschaft ist nur für spezifische Protokolle oder Schnittstellen zulässig. Deshalb kann es zwischen Ihrer Angabe einer Verbindungseigenschaft und dem gewünschten Protokoll bzw. der gewünschten Schnittstelle zu Konflikten kommen. In diesem Fall gibt "TCONSettings" am Parameter STATUS den Wert W#16#8090 zurück.

Wie lässt sich eine TCP-Verbindung beenden?

Sie können eine bestehende TCP-Verbindung auf eine von zwei Arten beenden:

- Mit TCP-Reset (Standard)
Die Verbindung wird geschlossen, nachdem das Telegramm mit dem in der Kopfzeile gesetzten RST-Bit gesendet wurde. Die zugehörigen Ressourcen werden sofort gelöscht und aktiviert. Restliche Daten werden weder gesendet noch an das Anwenderprogramm übertragen.

Hinweis

Beenden einer TCP-Verbindung bei S7-1500 CPUs mit Firmware-Version < V2.9 und S7-1200 CPUs mit Firmware-Version < V4.5

Bei S7-1500 CPUs mit Firmware-Version < V2.9 und S7-1200 CPUs mit Firmware-Version < V4.5 wird eine TCP-Verbindung immer mit TCP-Reset beendet.

- Mit TCP-Finish:
Wenn Sie TCP-Finish zum Beenden einer Verbindung festgelegt haben und dann die Anweisung "TDISCON" aufrufen, wird die Verbindung aus Sicht des Anwenders geschossen, nachdem "TDISCON" mit DONE = WAHR beendet wurde, d.h. die Verbindung ist wieder verfügbar. In den unteren Schichten im TCP/IP-Stack des Moduls jedoch sind die Ressourcen noch eine Zeit lang zugewiesen, genauso wie die zur Verbindung gehörenden Diagnoseobjekte.
Wenn Sie viele Verbindungen mit TCP-Finish entfernen und Verbindungen reservieren (mit "TCONSettings") oder herstellen (mit "TCON"), bevor die Zeit zum Aktivieren der Ressourcen abgelaufen ist, kann das zu einem Ressourcenengpass führen.

Bedingungen für TCP-Finish

Die folgenden Bedingungen müssen erfüllt sein, damit Sie eine Verbindung mit TCP-Finish ordnungsgemäß beenden können:

- Das verwendete Protokoll ist TCP.
- Die zugehörige Schnittstelle befindet sich an der CPU.
- Der Grund für die Beendigung der Verbindung ist der Aufruf der Anweisung "TDISCON".

Hinweis**TCP-Verbindung beim Übergang in STOP beenden**

Während eines Wechsels in STOP wird eine TCP-Verbindung immer mit einem TCP-Reset beendet.

SDT für Verbindungsbeendigung: TCON_TCPTermination

Der SDT für die Beendigung der Verbindung hat die folgende Struktur:

Parameter	Datentyp	Startwert	Beschreibung
GracefulShut-down	Bool	FALSCH	<ul style="list-style-type: none"> FALSCH: Verwenden Sie TCP-Reset, um die Verbindung zu beenden. WAHR: Verwenden Sie TCP-Finish, um die Verbindung zu beenden.

11.5.8.13 Ältere Anweisungen TCON, TDISCON, TSEND und TRCV

Vor STEP 7 V13 SP1 und den S7-1200 V4.1 CPUs haben die Anweisungen TCON, TDISCON, TSEND und TRCV nur mit Verbindungsparametern mit Strukturen nach "TCON_Param" funktioniert. Für beide Anweisungsarten gelten die allgemeinen Konzepte. Informationen zur Programmierung finden sich in den Beschreibungen zu den einzelnen TCON-, TDISCON-, TSEND- und TRCV-Anweisungen.

Version der Anweisungen TCON, TDISCON, TSEND und TRCV auswählen

Es gibt zwei Versionen der Anweisungen TCON, TDISCON, TSEND oder TRCV in STEP 7:

- Die Versionen 2.5 und 3.1 waren in STEP 7 Basic/Professional V13 und früher verfügbar.
- Version 4.0 ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Verwenden Sie niemals Anweisungen unterschiedlicher Versionen zusammen in einem CPU-Programm.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.

Open user communication		V3.1
TSEND_C	Send data via Ethernet (TCP)	V2.5
TRCV_C	Receive data via Ethernet (T...	V3.1
TMAIL_C	Send e-mail	V4.0
Others		V2.1
TCON	Establish communication c...	V3.0
TDISCON	Terminate communication ...	V2.1
TSEND	Send data via communicati...	V3.0
TRCV	Receive data via communic...	V3.0

Um die Version einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine TCON-, TDISCON-, TSEND- oder TRCV-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten TCON-, TDISCON-, TSEND- oder TRCV-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmeditor angezeigten Box. Wählen Sie in der Projektnavigation eine FB- oder FC-Instanz einer TCON-, TDISCON-, TSEND- oder TRCV-Anweisung aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der TCON-, TDISCON-, TSEND- oder TRCV-Anweisung anzuzeigen.

Alte Anweisungen TCON, TDISCON, TSEND und TRCV (TCP-Kommunikation)

Ethernet-Kommunikation mit den Protokollen TCP und ISO on TCP

Hinweis

Anweisungen TSEND_C und TRCV_C

Um die Programmierung der PROFINET/Ethernet-Kommunikation zu vereinfachen, verbinden die Anweisungen TSEND_C und TRCV_C die Funktionalität der Anweisungen TCON, TDISCON, TSEND und TRCV:

- TSEND_C verbindet die Anweisungen TCON, TDISCON und TSEND.
 - TRCV_C verbindet die Anweisungen TCON, TDISCON und TRCV.
-

Die folgenden Anweisungen steuern den Kommunikationsprozess:

- TCON baut die TCP/IP-Verbindung zwischen dem Client- und dem Server-PC (CPU) auf.
- TSEND und TRCV senden und empfangen Daten.
- TDISCON trennt die Verbindung.

Die Mindestdatengröße, die gesendet (TSEND) oder empfangen (TRCV) werden kann, ist ein Byte; die maximale Datengröße beträgt 8192 Byte. TSEND unterstützt nicht die Datenübertragung von Booleschen Adressen und TRCV empfängt Daten nicht in Booleschen Adressen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 187).

TCON, TDISCON, TSEND und TRCV laufen asynchron ab, so dass die Bearbeitung eines Auftrags mehrere Anweisungsausführungen umfasst. Beispielsweise starten Sie einen Auftrag zum Einrichten und Herstellen einer Verbindung durch die Ausführung von TCON mit Parameter REQ = 1. Dann überwachen Sie den Stand der Auftragsausführung, indem TCON wiederholt ausgeführt wird, und prüfen, ob der Parameter DONE den Abschluss des Auftrags bestätigt.

Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE und ERROR. Den aktuellen Status eines Auftrags können Sie anhand der Tabelle feststellen:

Tabelle 11-37 Interaktionen zwischen den Parametern BUSY, DONE und ERROR

BUSY	DONE	ERROR	Beschreibung
WAHR	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSCH	WAHR	FALSCH	Auftrag erfolgreich ausgeführt.
FALSCH	FALSCH	WAHR	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Kein neuer Auftrag zugewiesen.

TCON und TDISCON

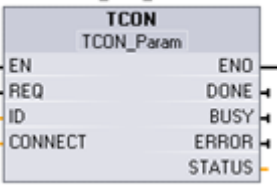
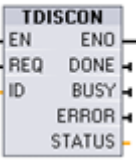
Hinweis

Initialisierung der Kommunikationsparameter

Nachdem Sie die Anweisung TCON eingefügt haben, konfigurieren Sie in den "Eigenschaften" der Anweisung (Seite 600) die Kommunikationsparameter (Seite 624). Wenn Sie die Parameter für die Kommunikationspartner im Inspektorfenster eingeben, gibt STEP 7 die entsprechenden Daten in den Instanz-DB der Anweisung ein.

Wenn Sie einen Multiinstanz-DB verwenden möchten, müssen Sie den DB manuell auf beiden CPUs konfigurieren.

Tabelle 11-38 Anweisungen TCON und TDISCON

KOP/FUP		Beschreibung
	<pre>"TCON_DB" (req:=_bool_in_, ID:=_undef_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, connect:=_struct_inout_);</pre>	TCP und ISO on TCP: TCON initiiert eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.
	<pre>"TDISCON_DB" (req:=_bool_in_, ID:=_word_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	TCP und ISO on TCP: TDISCON beendet eine Kommunikationsverbindung von der CPU zu einem Kommunikationsteilnehmer.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-39 Datentypen für die Parameter von TCON und TDISCON

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Steuerparameter REQ startet den Auftrag durch Herstellung der von ID angegebenen Verbindung. Der Auftrag beginnt mit einer steigenden Flanke.
ID	IN	CONN_OUC (Word)	Verweis auf die Verbindung, die mit dem entfernten Teilnehmer oder zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems hergestellt (TCON) oder beendet (TDISCON) werden soll. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen. Wertebereich: W#16#0001 bis W#16#0FFF
CONNECT (TCON)	IN_OUT	TCON_Param	Pointer auf die Verbindungsbeschreibung (Seite 624)
DONE	OUT	Bool	<ul style="list-style-type: none"> 0: Auftrag noch nicht gestartet oder läuft noch. 1: Auftrag fehlerfrei ausgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0: Auftrag beendet. 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Kein Fehler 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für ERROR und STATUS" unten.)

Beide Kommunikationsteilnehmer führen die Anweisung TCON aus, um die Kommunikationsverbindung einzurichten und herzustellen. Mit Parametern geben Sie die aktiven und passiven Endpunkte der Kommunikation an. Nach dem Einrichten und Aufbauen der Verbindung wird sie automatisch von der CPU gehalten und überwacht.

Wird die Verbindung durch eine Leitungsstörung oder vom entfernten Teilnehmer abgebrochen, so versucht der aktive Teilnehmer, sie wieder aufzubauen. Sie müssen TCON nicht erneut ausführen.

Wird die Anweisung TDISCON ausgeführt oder ist die CPU in den Betriebszustand STOP gegangen, so wird die bestehende Verbindung beendet und die eingerichtete Verbindung gelöscht. Um die Verbindung einzurichten und wiederherzustellen, müssen Sie TCON erneut ausführen.

Tabelle 11-40 Bedingungscode von ERROR und STATUS für TCON und TDISCON

ERROR	STATUS	Beschreibung
0	0000	Verbindung wurde erfolgreich aufgebaut.
0	7000	Keine laufende Auftragsbearbeitung
0	7001	Auftragsbearbeitung starten, Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON)
0	7002	Folgeaufruf (REQ irrelevant), Verbindung herstellen (TCON) oder Verbindung beenden (TDISCON)
1	8086	Parameter ID ist außerhalb des zulässigen Adressbereichs.

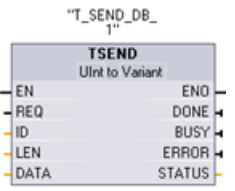
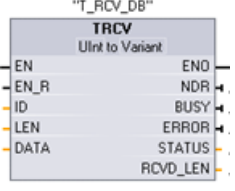
ERROR	STATUS	Beschreibung
1	8087	TCON: Maximale Anzahl Verbindungen erreicht; keine weitere Verbindung möglich.
1	809B	TCON: local_device_id in der Verbindungsbeschreibung stimmt nicht mit der CPU überein.
1	80A1	TCON: Verbindung oder Port ist vom Anwender belegt.
1	80A2	TCON: Lokaler oder entfernter Port ist vom System belegt.
1	80A3	Es wird versucht, eine vorhandene Verbindung erneut aufzubauen (TCON) oder eine nicht vorhandene Verbindung zu beenden (TDISCON).
1	80A4	TCON: IP-Adresse des entfernten Verbindungspunkts ist ungültig, sie entspricht der lokalen IP-Adresse.
1	80A5	TCON: Verbindungs-ID (Seite 621) wird bereits verwendet.
1	80A7	TCON: Kommunikationsfehler: Sie haben TDISCON ausgeführt, bevor TCON beendet war. TDISCON muss zunächst die von der ID angegebene Verbindung vollständig beenden.
1	80B2	TCON: Der Parameter CONNECT zeigt auf einen Datenbaustein, der mit dem Attribut "Nur im Ladespeicher ablegen" generiert wurde.
1	80B4	TCON: Wenn Sie eine passive Verbindung über ISO on TCP (connection_type = B#16#12) aufbauen, werden Sie anhand des Fehlercodes 80B4 gewarnt, wenn der eingegebene TSAP keiner der folgenden Voraussetzungen für Adressen entspricht: <ul style="list-style-type: none"> • Hat der lokale TSAP eine Länge von 2 und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein. • Hat der lokale TSAP eine Länge von 3 oder größer und einen ID-Wert von E0 oder E1 (hexadezimal) für das erste Byte, muss das zweite Byte entweder 00 oder 01 sein und alle Bytes müssen gültige ASCII-Zeichen sein. • Hat der lokale TSAP eine Länge von 3 oder größer und das erste Byte des ID-Werts des TSAP nicht den Wert E0 oder E1 (hexadezimal), müssen alle Bytes der TSAP-ID gültige ASCII-Zeichen sein. Gültige ASCII-Zeichen sind Bytewerte von 20 bis 7E (hexadezimal).
1	80B5	TCON: Verbindungstyp "13 = UDP" gestattet nur passiven Verbindungsaufbau.
1	80B6	TCON: Parametrierungsfehler im Parameter CONNECTION_TYPE des SDT TCON_Param.
1	80B7	TCON: Fehler in einem der folgenden Parameter des Datenbausteins für die Verbindungsbeschreibung: <ul style="list-style-type: none"> • block_length • local_tsap_id_len • rem_subnet_id_len • rem_staddr_len • rem_tsap_id_len • next_staddr_len Hinweis: Beim Betrieb von TCON im passiven TCP-Modus muss LOCAL_TSAP_ID_LEN "2" und REM_TSAP_ID_LEN muss "0" sein.
1	80B8	TCON: Parameter der lokalen Verbindungsbeschreibung und Parameter-ID sind verschieden.
1	80C3	TCON: Alle Verbindungsressourcen sind belegt.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> • Die Verbindung kann zur Zeit nicht hergestellt werden (TCON). • Die konfigurierte Verbindung wird von TDISCON (TCON) entfernt. • Die angegebene Verbindung wird derzeit aufgebaut (TDISCON). • Die Schnittstelle empfängt neue Parameter (TCON und TDISCON).

TSEND und TRCV

Hinweis

Wenn Sie in der offenen Benutzerkommunikation über PROFINET eine Anweisung TSEND ohne die entsprechende Anweisung TRCV auf dem Remote-Gerät ausführen, kann es passieren, dass die Anweisung TSEND auf unbestimmte Zeit im Zustand "Busy" bleibt und darauf wartet, dass die Anweisung TRCV die Daten empfängt. In diesem Zustand ist der Ausgang "Busy" der Anweisung TSEND gesetzt und der Ausgang "Status" hat den Wert "0x7002". Diese Bedingung kann auftreten, wenn Sie mehr als 4096 Byte an Daten übertragen. Das Problem wird bei der nächsten Ausführung der Anweisung TRCV behoben.

Tabelle 11-41 Anweisungen TSEND und TRCV

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	TCP und ISO on TCP: TSEND sendet über eine Kommunikationsverbindung Daten von der CPU zu einer Partnerstation.
	<pre>"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_);</pre>	TCP und ISO on TCP: TRCV empfängt über eine Kommunikationsverbindung Daten von einer Partnerstation auf der CPU.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 11-42 Datentypen für die Parameter von TSEND und TRCV

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool TSEND: Startet den Sendeauftrag bei einer steigenden Flanke. Die Daten werden aus dem mit DATA und LEN bezeichneten Bereich übertragen.
EN_R	IN	Bool TRCV: Aktiviert die CPU für den Empfang, bei EN_R = 1 ist TRCV empfangsbereit. Der Empfangsauftrag wird bearbeitet.
ID	IN	CONN_OUC (Word) Dies ist ein Verweis auf die zugehörige Verbindung. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen. Wertebereich: W#16#0001 bis W#16#0FFF

Parameter und Datentyp		Datentyp	Beschreibung
LEN	IN	UInt	Maximale Bytezahl, die gesendet (TSEND) oder empfangen (TRCV) werden soll: <ul style="list-style-type: none"> • Standard = 0: Der Parameter DATA legt die Länge der zu sendenden (TSEND) oder zu empfangenden (TRCV) Daten fest. • Ad-hoc-Modus = 65535: Für den Datenempfang ist eine variable Länge eingestellt (TRCV).
DATA	IN_OUT	Variante	Pointer auf den Datenbereich für Senden (TSEND) oder Empfangen (TRCV). Der Datenbereich enthält die Adresse und Länge. Die Adresse bezieht sich auf einen der Speicherbereiche E, A oder M oder auf einen DB.
DONE	OUT	Bool	TSEND: <ul style="list-style-type: none"> • 0: Auftrag noch nicht gestartet oder läuft noch. • 1: Auftrag fehlerfrei ausgeführt.
NDR	OUT	Bool	TRCV: <ul style="list-style-type: none"> • NDR = 0: Auftrag noch nicht gestartet oder läuft noch. • NDR= 1: Der Auftrag wurde erfolgreich durchgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • BUSY = 1: Der Auftrag ist noch nicht beendet. Es kann kein neuer Auftrag gestartet werden. • BUSY = 0: Auftrag beendet.
ERROR	OUT	Bool	ERROR = 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für Error und Status" unten.)
RCVD_LEN	OUT	Int	TRCV: Tatsächliche empfangene Datenmenge in Byte

Hinweis

Die Anweisung TSEND erfordert einen Wechsel von 0 nach 1 am Eingangsparameter REQ, um einen Sendeauftrag zu starten. Der Parameter BUSY wird dann während der Verarbeitung auf 1 gesetzt. Die Fertigstellung des Sendeauftrags wird kenntlich gemacht, indem einer der Parameter DONE oder ERROR einen Zyklus lang auf 1 gesetzt wird. Während dieses Zeitraums wird ein Wechsel von 0 nach 1 am Eingangsparameter REQ ignoriert.

Funktionsweise von TRCV

Die Anweisung TRCV schreibt die empfangenen Daten in einen Empfangsbereich, der von den folgenden zwei Variablen angegeben wird:

- Zeigt auf den Anfang des Bereichs
- Länge des Bereichs oder der Wert, der am Eingang LEN bereitgestellt wird, sofern er nicht 0 ist

Hinweis

Die Standardeinstellung des Parameters LEN (LEN = 0) ermittelt anhand des Parameters DATA die Länge der zu übertragenden Daten. Es ist empfehlenswert, dass die von der Anweisung TSEND gesendeten Daten die gleiche Größe haben wie der Parameter DATA der Anweisung TRCV.

Wenn Sie die Standardeinstellung des Parameters LEN verwenden und die Daten in kleineren Segmenten als die Größe des Parameters DATA gesendet werden sollen, gilt Folgendes. Es ist empfehlenswert, das Bit EN_R auf 1 zu halten, bis die entsprechende Anweisung TSEND die entsprechende Menge an Daten überträgt, um den Parameter DATA von TRCV zu füllen. Wenn die Größe der von TSEND übertragenen Daten nicht der Größe des Parameters DATA von TRCV entspricht, bleibt TRCV im Status "Belegt" (Statuscode: 7002), solange das Bit EN_R = 1 ist, bis die Gesamtgröße der von TSEND übertragenen Daten der Größe des Parameters DATA von TRCV entspricht. Wenn das Bit EN_R von TRCV Impulse empfängt, muss es die gleiche Anzahl von Impulsen erhalten, wie die Anweisung TSEND ausgeführt wird, um Daten zu empfangen.

Der Puffer des Parameters DATA von TRCV zeigt die neu empfangenen Daten erst an, wenn die Datengröße gleich der Puffergröße des Parameters DATA ist.

Sobald alle Auftragsdaten empfangen sind, werden sie von TRCV in den Empfangsbereich übertragen und NDR wird auf 1 gesetzt.

Tabelle 11-43 Eintragen der Daten in den Empfangsbereich

Protokollvariante	Eingabe der Daten im Empfangsbereich	Parameter "connection_type"	Wert des Parameters LEN	Wert des Parameters RCVD_LEN (Byte)
TCP	Ad-hoc-Modus	B#16#11	65535	1 bis 1472
TCP	Datenempfang mit angegebener Länge	B#16#11	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192
ISO on TCP	Ad-hoc-Modus	B#16#12	65535	1 bis 1472
ISO on TCP	protokollgesteuert	B#16#12	0 (empfohlen) oder 1 bis 8192, außer 65535	1 bis 8192

Hinweis**Ad-hoc-Modus**

Den Ad-hoc-Modus gibt es nur bei den Protokollvarianten TCP und ISO on TCP. Den Ad-hoc-Modus stellen Sie ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen. Der Empfangsbereich ist identisch mit dem von Parameter DATA angegebenen Bereich. Die Länge der empfangenen Daten wird an den Parameter RCVD_LEN ausgegeben. Unmittelbar nach dem Empfang eines Datenblocks schreibt TRCV die Daten in den Empfangsbereich und setzt NDR auf 1.

Wenn Sie die Daten in einem "optimierten" DB (nur symbolisch adressierbar) speichern, können Sie Daten nur in Arrays der Datentypen Byte, Char, USInt und SInt empfangen.

Hinweis**STEP 7-Projekte der S7-300/400 mit Ad-hoc-Modus in die S7-1200 importieren**

In STEP 7-Projekten der S7-300/400 wählen Sie den Ad-hoc-Modus aus, indem Sie dem Parameter LEN den Wert 0 zuweisen. Bei der S7-1200 stellen Sie den Ad-hoc-Modus ein, indem Sie dem Parameter LEN den Wert 65535 zuweisen.

Wenn Sie ein STEP 7-Projekt der S7-300/400, das den Ad-hoc-Modus umfasst, in die S7-1200 importieren, müssen Sie den Parameter LEN in 65535 ändern.

Bedingungscode von TSEND und TRCV Error und Status

ERROR	STATUS	Beschreibung
0	0000	<ul style="list-style-type: none"> • Sendeauftrag fehlerfrei ausgeführt (TSEND) • Neue Daten angenommen: Die aktuelle Länge der empfangenen Daten wird in RCVD_LEN (TRCV) gezeigt.
0	7000	<ul style="list-style-type: none"> • Keine laufende Auftragsbearbeitung (TSEND) • Baustein nicht empfangsbereit (TRCV)
0	7001	<ul style="list-style-type: none"> • Auftragsbearbeitung starten, Daten werden gesendet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND). • Baustein empfangsbereit, Empfangsauftrag wurde aktiviert (TRCV).
0	7002	<ul style="list-style-type: none"> • Folgeaufruf (REQ irrelevant), Auftrag wird bearbeitet: Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu (TSEND). • Folgeaufruf, Empfangsauftrag wird bearbeitet: Während dieser Bearbeitung werden die Daten in den Empfangsbereich geschrieben. Deshalb kann ein Fehler zu inkonsistenten Daten im Empfangsbereich führen (TRCV).
1	8085	<ul style="list-style-type: none"> • Der Parameter LEN ist größer als der größte zulässige Wert (TSEND) und (TRCV). • Der Parameter LEN oder DATA hat sich seit der ersten Ausführung der Anweisung geändert (TRCV).
1	8086	Der Parameter ID ist nicht im zulässigen Adressbereich.
1	8088	Der Parameter LEN ist größer als der in DATA festgelegte Speicherbereich.

ERROR	STATUS	Beschreibung
1	80A1	Kommunikationsfehler: <ul style="list-style-type: none"> Die angegebene Verbindung ist noch nicht hergestellt (TSEND und TRCV). Die angegebene Verbindung wird beendet. Ein Sende- oder Empfangsauftrag ist über diese Verbindung nicht möglich (TSEND und TRCV). Die Schnittstelle wird neu initialisiert (TSEND). Die Schnittstelle empfängt neue Parameter (TRCV).
1	80C3	Interner Mangel an Ressourcen: Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsklasse bearbeitet.
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> Die Verbindung zum Kommunikationspartner kann zur Zeit nicht hergestellt werden. Die Schnittstelle empfängt neue Parametrierungen oder die Verbindung wird gerade aufgebaut.

Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TSEND_C, TRCV_C, TSEND und TRCV unterstützen alle die Ethernet-Protokolle TCP und ISO on TCP.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

11.5.8.14 Anweisung T_RESET (bestehende Verbindung beenden und neu aufbauen)

Mit der Anweisung "T_RESET" wird eine bestehende Verbindung beendet und anschließend neu aufgebaut:

Tabelle 11-44 Anweisung T_RESET

KOP/FUP	SCL	Beschreibung
	<pre>"T_RESET_DB" (req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, status=>_word_out_);</pre>	<p>Mit der Anweisung "T_RESET" wird eine bestehende Verbindung beendet und anschließend neu aufgebaut.</p>

Die lokalen Endpunkte der Verbindung bleiben erhalten. Sie werden automatisch erzeugt:

- wenn eine Verbindung konfiguriert und in die CPU geladen wurde
- wenn eine Verbindung vom Anwenderprogramm erzeugt wurde, zum Beispiel durch Aufruf der Anweisung "TCON (Seite 657)".

Die Anweisung "T_RESET" kann für alle Verbindungstypen ausgeführt werden, unabhängig davon, ob die lokale Schnittstelle der CPU oder die Schnittstelle eines CM/CP für die Verbindung verwendet wurde. Eine Ausnahme bilden Verbindungen für die Datenübertragung im Ad-hoc-

Modus mit TCP, da auf diese Verbindungen nicht mit einer Verbindungs-ID verwiesen werden kann.

Wenn die Anweisung "T_RESET" mit dem Parameter REQ aufgerufen wurde, wird die mit dem Parameter ID angegebene Verbindung beendet und, falls erforderlich, der Sende- und Empfangspuffer für die Daten gelöscht. Durch den Abbau der Verbindung wird auch eine laufende Datenübertragung abgebrochen. Daher kann es zu Datenverlust kommen, wenn gerade eine Datenübertragung stattfindet. Die als aktiver Verbindungspartner eingerichtete CPU versucht dann automatisch, die unterbrochene Kommunikationsverbindung wieder aufzubauen. Es ist deshalb nicht nötig, die Anweisung "TCON (Seite 657)" aufzurufen, um die Kommunikationsverbindung wiederherzustellen.

Die Ausgangsparameter DONE, BUSY und STATUS geben den Auftragsstatus an.

Datentypen für die Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung T_RESET:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L, T, C oder konstant	Der Steuerparameter REQUEST startet den Auftrag zur Beendigung der durch ID angegebenen Verbindung. Der Auftrag startet bei einer steigenden Flanke.
ID	Input	CONN_OUC (WORD)	L, D oder konstant	Verweis auf die Verbindung mit dem passiven Partner, die beendet werden soll. ID muss mit der entsprechenden Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen. Wertebereich: W#16#0001 bis W#16#0FFF
DONE	Output	BOOL	E, A, M, D, L	Statusparameter DONE <ul style="list-style-type: none"> 0: Auftrag noch nicht gestartet oder wird noch ausgeführt. 1: Auftrag fehlerfrei ausgeführt.
BUSY	Output	BOOL	E, A, M, D, L	Statusparameter BUSY <ul style="list-style-type: none"> 0: Auftrag beendet. 1: Auftrag noch nicht beendet.
ERROR	Output	BOOL	E, A, M, D, L	Statusparameter ERROR <ul style="list-style-type: none"> 0: Kein Fehler. 1: Fehler bei der Bearbeitung aufgetreten. Parameter STATUS enthält ausführliche Informationen über die Fehlerart
STATUS	Output	WORD	E, A, M, D, L	Statusparameter STATUS Fehlerinformation (siehe Tabelle "STATUS-Parameter").

Parameter STATUS

Fehlerbit	STATUS* (W#16#...)	Beschreibung
0	0000	Kein Fehler.
0	0001	Verbindung nicht aufgebaut.
0	7001	Verbindungsabbau gestartet.
0	7002	Verbindung wird beendet.
1	8081	Unbekannte Verbindung an Parameter ID angegeben.

11.5.8.15 Anweisung T_DIAG (Status einer Verbindung prüfen und Informationen lesen)

Die Anweisung T_DIAG prüft den Status einer Verbindung und liest weitere Informationen auf dem lokalen Endpunkt dieser Verbindung:

Tabelle 11-45 Anweisung T_DIAG

KOP/FUP	SCL	Beschreibung
	<pre>"T_DIAG_DB" (req:=_bool_in_, id:=_word_in_, done=>_bool_out_, error=>_bool_out_, status=>_dword_out_);</pre>	<p>Mit der Anweisung T_DIAG prüfen Sie den Status einer Verbindung und lesen weitere Informationen auf dem lokalen Endpunkt dieser Verbindung.</p>

Die Anweisung T_DIAG funktioniert wie folgt:

- Die Verbindung wird vom Parameter ID angegeben. Sie können beide im Verbindungseditor konfigurierte Verbindungsendpunkte und programmierte Verbindungsendpunkte (z. B. mit der Anweisung TCON) lesen. Temporäre Verbindungsendpunkte (z. B. bei der Verbindung mit einer Engineering Station erstellte Endpunkte) können nicht diagnostiziert werden, weil bei diesem Vorgang keine Verbindungs-ID erzeugt wird.
- Die gelesenen Verbindungsinformationen werden in einer vom Parameter RESULT angegebenen Struktur gespeichert.
- Der Ausgangsparameter STATUS gibt an, ob das Lesen der Verbindungsinformationen möglich war. Die Verbindungsinformationen in der Struktur am Parameter RESULT sind nur gültig, wenn die Anweisung T_DIAG mit STATUS = W#16#0000 und ERROR = FALSE abgeschlossen wurde. Tritt ein Fehler auf, können die Verbindungsinformationen nicht ausgewertet werden.

Mögliche Verbindungsinformationen

Die Struktur TDiag_Status kann zum Lesen der Verbindungsinformationen am Parameter RESULT verwendet werden. Die Struktur TDiag_Status enthält nur die wichtigsten Informationen über einen Verbindungsendpunkt (z. B. das verwendete Protokoll, den Verbindungszustand und die Anzahl der gesendeten oder empfangenen Datenbytes).

Die Struktur und Parameter der TDiag_Status-Struktur werden nachstehend beschrieben (siehe Tabelle zur Struktur TDIAG_Status).

Datentypen für die Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung T_DIAG:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L, T, C oder Konstante	Startet die Anweisung zum Prüfen der im Parameter ID angegebenen Verbindung, wenn eine steigende Flanke vorliegt.
ID	Input	CONN_OUC (WORD)	L, D oder Konstante	Verweis auf die zugewiesene Verbindung. Wertebereich: W#16#0001 bis W#16#0FFF
RESULT	InOut	VARIANT	D	Pointer auf die Struktur, in der die Verbindungs- informationen gespeichert werden. Die Struktur TDiag_Status kann am Parameter RESULT verwendet werden (eine Beschreibung finden Sie in der Tabelle zur Struktur TDIAG_Status).
DONE	Output	BOOL	E, A, M, D, L	Statusparameter: <ul style="list-style-type: none"> 0: Anweisung noch nicht gestartet oder noch in Bearbeitung. 1: Anweisung fehlerfrei ausgeführt.
BUSY	Output	BOOL	E, A, M, D, L	Statusparameter: <ul style="list-style-type: none"> 0: Anweisung noch nicht gestartet oder bereits beendet. 1: Anweisung noch nicht abgeschlossen. Es kann kein neuer Auftrag gestartet werden.
ERROR	Output	BOOL	E, A, M, D, L	Statusparameter: <ul style="list-style-type: none"> 0: Kein Fehler. 1: Fehler aufgetreten.
STATUS	Output	WORD	E, A, M, D, L	Zustand der Anweisung

Parameter BUSY, DONE und ERROR

Den Status der Ausführung von T_DIAG können Sie mit den Parametern BUSY, DONE, ERROR und STATUS prüfen. Der Parameter BUSY zeigt den Bearbeitungszustand an. Mit dem Parameter DONE prüfen Sie, ob eine Anweisung erfolgreich ausgeführt wurde. Der Parameter ERROR wird gesetzt, wenn während der Ausführung von T_DIAG Fehler auftreten.

Die folgende Tabelle zeigt die Beziehungen zwischen den Parametern BUSY, DONE und ERROR:

BUSY	DONE	ERROR	Beschreibung
1	-	-	Die Anweisung wird bearbeitet.
0	1	0	Die Anweisung wurde erfolgreich ausgeführt. Die Daten in der von RESULT angegebenen Struktur sind nur gültig, wenn dies der Fall ist.
0	0	1	Anweisung mit Fehler beendet. Die Fehlerursache wird am Parameter STATUS ausgegeben.
0	0	0	Es wurde keine neue Anweisung zugewiesen.

Parameter STATUS

Die folgende Tabelle zeigt die Bedeutung der Werte am Parameter STATUS:

Fehlerbit	STATUS* (W#16#...)	Beschreibung
0	0000	Die Anweisung T_DIAG wurde erfolgreich ausgeführt. Die Daten in der am Parameter RESULT angegebenen Struktur können ausgewertet werden.
0	7000	Keine Anweisungsbearbeitung aktiv.
0	7001	Anweisungsbearbeitung gestartet.
0	7002	Verbindungsinformationen werden gelesen (Parameter REQ ist irrelevant).
1	8086	Der Wert am Parameter ID liegt außerhalb des gültigen Bereichs (W#16#0001 bis W#16#0FFF).
1	8089	Der Parameter RESULT zeigt auf einen ungültigen Datentyp (nur Strukturen TDIAG_Status und TDIAG_StatusExt).
1	80A3	Der Parameter ID verweist auf einen Verbindungsendpunkt, der nicht vorhanden ist. Bei programmierten Verbindungen kann dieser Fehler auch nach Aufruf der Anweisung TDISCON auftreten.
1	80C4	Interner Fehler. Der Zugriff auf den Verbindungsendpunkt ist kurzzeitig nicht verfügbar.

Struktur TDIAG_Status

Die folgende Tabelle enthält Einzelheiten zur Struktur TDIAG_Status. Der Wert jedes Elements ist nur gültig, wenn die Anweisung fehlerfrei ausgeführt wurde. Tritt ein Fehler auf, ändert sich der Inhalt der Parameter nicht:

Name	Datentyp	Beschreibung
Die folgenden Parameter befinden sich in der Struktur TDIAG_Status:		
InterfacelD	HW_ANY	Schnittstellen-ID (LADDR) der CPU oder des CM/CP.
ID	CONN_OUC	ID der diagnostizierten Verbindung. Nach einem erfolgreichen Aufruf ist der Wert dieses Elements mit der Parameter-ID der Anweisung T_DIAG identisch.

Name	Datentyp	Beschreibung
ConnectionType	BYTE	<p>Für die Verbindung verwendeter Protokolltyp:</p> <ul style="list-style-type: none"> • 0x01: Nicht verwendet. • ... • 0x0B: TCP-Protokoll (IP_v4) • 0x0C: ISO-on-TCP-Protokoll (RFC1006) • 0x0D: TCP-Protokoll (DNS) • 0x0E: Wählprotokoll • 0x0F: WDC-Protokoll • 0x10: SMTP-Protokoll • 0x11: TCP-Protokoll • 0x12: TCP- und ISO-on-TCP-Protokoll (RFC1006) • 0x13: UDP-Protokoll • 0x14: Reserviert • 0x15: PROFIBUS-Buszugriffsprotokoll (FDL) • 0x16: Transportprotokoll nach ISO 8073 (ISO nativ) • ... • 0x20: SMTP- oder SMTPS-Protokoll - basierend auf IPv4 • 0x21: SMTP- oder SMTPS-Protokoll - basierend auf IPv6 • 0x22: SMTP- oder SMTPS-Protokoll - basierend auf FQDN (Fully Qualified Domain Name) • ... • 0x70: S7-Verbindung • Andere: Reserviert
ActiveEstablished	BOOL	<ul style="list-style-type: none"> • FALSE: Lokal, der passive Verbindungsendpunkt • TRUE: Lokal, der aktive Verbindungsendpunkt
State	BYTE	<p>Aktueller Status des Verbindungsendpunkts</p> <ul style="list-style-type: none"> • 0x00: Nicht verwendet. • 0x01: Verbindung beendet. Temporärer Status, beispielsweise nach Aufruf der Anweisung T_RESET. Das System versucht dann automatisch, die Verbindung wiederherzustellen. • 0x02: Der aktive Verbindungsendpunkt versucht, eine Verbindung zum dezentralen Kommunikationspartner herzustellen. • 0x03: Der passive Verbindungsendpunkt wartet auf die Herstellung der Verbindung zum dezentralen Kommunikationspartner. • 0x04: Verbindung hergestellt. • 0x05: Die Verbindung wird beendet. Dies geschieht möglicherweise, weil die Anweisung "T_RESET" oder "T_DISCON" aufgerufen wurde. Andere mögliche Gründe sind Protokollfehler oder Leitungspausen. • 0x06..0xFF: Nicht verwendet.

Name	Datentyp	Beschreibung
Kind	BYTE	Modus des Verbindungsendpunkts: <ul style="list-style-type: none"> • 0x00: Nicht verwendet. • 0x01: Konfigurierte statische Verbindung, die konfiguriert und in die CPU geladen wurde. • 0x02: Konfigurierte dynamische Verbindung, die konfiguriert und in die CPU geladen wurde (wird derzeit nicht unterstützt). • 0x03: Programmierte Verbindung, die im Anwenderprogramm mit der Anweisung "TCON" erzeugt wurde. Ein Aufruf der Anweisung TDISCON oder ein Wechsel in den Betriebszustand STOP der CPU hat den Verbindungsendpunkt zerstört. • 0x04: Temporäre, dynamische Verbindung, die beispielsweise von der Engineering Station (ES) oder der Operator Station (OS) aufgebaut wurde. (Dieser Verbindungstyp kann derzeit nicht diagnostiziert werden, weil es keine ID gibt). • 0x05..0xFF: Nicht verwendet.
SentBytes	UDINT	Anzahl der gesendeten Datenbytes.
ReceivedBytes	UDINT	Anzahl der empfangenen Datenbytes.

11.5.8.16 Anweisung TMAIL_C (Email über die Ethernet-Schnittstelle der CPU senden)

Überblick

Mit der Anweisung "TMAIL_C" können Sie eine E-Mail über die Ethernet-Schnittstelle der S7-1200 CPU senden.

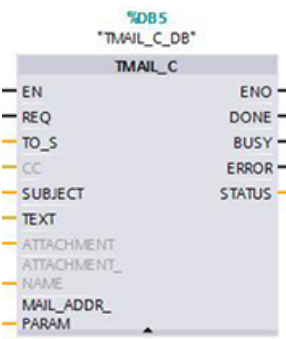
Die Anweisung TMAIL_C hat zwei Funktionalitäten:

- E-Mail über die CPU-Schnittstelle
- E-Mail über eine CP-Schnittstelle

Zur Verwendung der Anweisung TMAIL_C müssen die folgenden Voraussetzungen erfüllt sein:

- Sie haben die Hardware konfiguriert
- Die Netzwerkinfrastruktur gestattet eine Kommunikationsverbindung zum Mailserver

Tabelle 11-46 Anweisung TMAIL_C

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit der Anweisung TMAIL_C können Sie eine E-Mail über die Ethernet-Schnittstelle der S7-1200 CPU senden.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Den Inhalt der E-Mail und die Verbindungsdaten legen Sie mit den folgenden Parametern fest:

- Empfängeradressen mit den Parametern TO_S und CC
- Inhalt der E-Mail mit den Parametern SUBJECT und TEXT
- Optionalen Anhang mit VARIANT-Pointern an den Parametern ATTACHMENT und ATTACHMENT_NAME
- Verbindungsdaten, Adressierung und Authentifizierung für den Mailserver am Parameter MAIL_ADDR_PARAM (Seite 693)

Ab TMAIL_C Version V6.0 oder höher und S7-1200 CPU-Firmware ab Version V4.4 können Sie die Anweisung TMAIL_C zum Senden einer E-Mail mit sicherer Kommunikation über einen integrierten Ethernet-Port Ihrer S7-1200 CPU verwenden. Sie definieren die für den Sendevorgang erforderlichen Daten am Parameter MAIL_ADDR_PARAM (Seite 693) mit dem SDT TMail_V4_SEC oder TMail_QDN_SEC.

Sie können eine SMS nicht direkt mit der Anweisung TMAIL_C senden. Ob die E-Mail vom Mailserver als SMS weitergeleitet werden kann oder nicht, hängt von Ihrem Telekommunikationsanbieter ab.

Funktionsweise der Anweisung

Das Senden der E-Mail wird gestartet, wenn die Flanke für den Parameter REQ von "0" auf "1" wechselt.

Die Anweisung TMAIL_C meldet den Auftragsstatus anhand der Ausgangsparameter "BUSY", "DONE", "ERROR" und "STATUS".

Die Anweisung TMAIL_C funktioniert asynchron, das heißt, ihre Ausführung erstreckt sich über mehrere Aufrufe. Beim Aufrufen der Anweisung "TMAIL_C" muss eine Instanz angegeben werden.

In den folgenden Fällen wird die Verbindung zum Mailserver abgebrochen:

- Wenn die CPU in den Betriebszustand STOP geht, während TMAIL_C aktiv ist.
- Wenn Kommunikationsprobleme auf dem Industrial Ethernet-Bus auftreten.
In diesem Fall wird das Senden der E-Mail unterbrochen und die E-Mail erreicht den Empfänger nicht.

Die Verbindung wird auch beendet, wenn die Anweisung erfolgreich ausgeführt und die E-Mail gesendet wurde.

ACHTUNG

Anwenderprogramme ändern

Sie können die Teile Ihres Anwenderprogramms, die direkt die Aufrufe von TMAIL_C betreffen, nur ändern, wenn:

- sich die CPU in STOP befindet
- gerade keine E-Mail gesendet wird (d. h. REQ = 0 und BUSY = 0).

Das betrifft insbesondere das Löschen und Ersetzen von Programmbausteinen, die Aufrufe von TMAIL_C oder der Instanz von TMAIL_C enthalten.

Wird diese Einschränkung nicht beachtet, können Verbindungsressourcen blockiert werden. Das Automatisierungssystem kann gegenüber den TCP/IP-Kommunikationsfunktionen über Industrial Ethernet in einen undefinierten Zustand wechseln.

Nach Übertragung der Änderungen ist ein Neustart der CPU mittels Warm- oder Kaltstart erforderlich.

Datenkonsistenz

Die Parameter TO_S, CC, SUBJECT, TEXT, ATTACHMENT und MAIL_ADDR_PARAM werden von der Anweisung TMAIL_C während ihrer Ausführung benutzt, sodass die Parameter erst geändert werden können, nachdem der Auftrag abgeschlossen ist (BUSY = 0).

SMTP-Authentifizierung

Authentifizierung bezeichnet ein Verfahren zur Identitätsprüfung, zum Beispiel mit einer Passwortabfrage.

Wenn Sie die Schnittstelle der S7-1200 CPU verwenden, unterstützt die Anweisung TMAIL_C das SMTP-Authentifizierungsverfahren AUTH-LOGIN, das für die meisten Mailserver erforderlich ist.

Informationen zum Authentifizierungsverfahren Ihres Mailserver finden Sie im Handbuch zu Ihrem Mailserver oder auf der Website Ihres Internet-Diensteanbieters.

- Bevor Sie das Authentifizierungsverfahren AUTH-LOGIN nutzen können, benötigt die Anweisung TMAIL_C den Benutzernamen, mit dem sie sich beim Mailserver anmelden soll. Dieser Benutzername ist der gleiche, mit dem Sie einen Mail-Account auf Ihrem Mailserver einrichten. Er wird mit dem Parameter UserName an die Struktur im Parameter MAIL_ADDR_PARAM übertragen.
Wird im Parameter MAIL_ADDR_PARAM kein Benutzername eingegeben, wird das Authentifizierungsverfahren AUTH-LOGIN nicht verwendet. Die E-Mail wird dann ohne Authentifizierung gesendet.
- Für die Anmeldung benötigt die Anweisung TMAIL_C ferner das zugehörige Passwort. Dies ist das Passwort, das Sie beim Einrichten Ihres Mail-Accounts angegeben haben. Er wird mit dem Parameter PassWord an die Struktur im Parameter MAIL_ADDR_PARAM übertragen.

Datentypen für die Parameter

Die folgende Tabelle zeigt die Parameter der Anweisung TMAIL_C:

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
REQ	Input	BOOL	E, A, M, D, L, T, C oder konstant	Steuerparameter REQUEST: Aktiviert das Senden einer E-Mail bei einer steigenden Flanke.
TO_S (Seite 702)	Input	STRING	D	Empfängeradressen STRING mit einer maximalen Länge von 180 Zeichen (Byte). Das E-Mail-Adressformat entnehmen Sie bitte dem Beispiel in der Parameterbeschreibung.
CC (Seite 702)	Input	STRING	D	CC-Empfängeradressen (optional) STRING mit einer maximalen Länge von 180 Zeichen (Byte). Gleiches E-Mail-Adressformat wie für Parameter TO_S. Wenn Sie hier eine leere Zeichenkette zuweisen, sendet die Anweisung die E-Mail nicht an einen CC-Empfänger.
SUBJECT	Input	STRING	D	Betreff der E-Mail STRING mit einer maximalen Länge von 180 Zeichen (Byte).
TEXT	Input	STRING	D	Text der E-Mail (optional) STRING mit einer maximalen Länge von 180 Zeichen (Byte). Wenn Sie an diesem Parameter eine leere Zeichenkette zuweisen, sendet die Anweisung die E-Mail ohne Text.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
ATTACHMENT	Input	VARIANT	D	E-Mail-Anhang (optional) Verweis auf ein Zeichen/Byte/Wort/ Doppelwort/Zeichenketten-Feld (Array- OfChar, ArrayOfByte, ArrayOfWord, Ar- rayOfDWord, or String) mit einer maxi- malen Länge von 64 Byte. Hinweis: Wenn Sie am Parameter AT- TACHMENT keinen Wert oder eine leere Zeichenkette zuweisen, sendet die An- weisung die E-Mail ohne Anhang.
ATTACH- MENT_NAME	Input	VARIANT	D	Name des E-Mail-Anhangs (optional) Verweis auf eine Zeichenkette mit einer maximalen Länge von 50 Zeichen (Byte), die den Dateinamen des An- hangs darstellt. Wird diesem Parameter eine leere Zeichenkette zugewiesen, wird der Mail-Anhang mit dem Dateina- men "attachment.bin" gesendet. Mit dem Parameter AttachmentName können Sie den Namen des Anhangs zu- weisen, der angezeigt wird, wenn der Kommunikationspartner die E-Mail empfängt. Der SDT TMail_FileReference verwendet automatisch den Parameter FileName für den Parameter Attach- mentName. Wenn Sie den SDT TMail_FileReference verwenden, ist der Parameter Attach- mentName nicht relevant. Lassen Sie ihn leer. Wenn Sie bei Verwendung des SDT TMail_FileReference den Parameter At- tachmentName eingeben, erzeugt die Anweisung TMAIL_C einen Fehler. Wei- tere Informationen finden Sie unter "Fehlercodes, Parameter SFB_STATUS des Instanz-DB".
MAIL_ADDR_P ARAM (Sei- te 693)	Input	VARIANT	D	Verbindungsparameter und Adresse des E-Mail-servers Die Verbindungsparameter legen Sie mit dem SDT mit der Struktur TMail_V4, TMail_FQDN, TMail_V4_SEC oder TMail_QDN_SEC fest (siehe Parameter- beschreibung).
DONE (Sei- te 702)	Output	BOOL	E, A, M, D, L	Statusparameter <ul style="list-style-type: none"> • DONE = 0: Auftrag noch nicht gest- artet oder wird noch ausgeführt. • DONE = 1: Auftrag wurde fehlerfrei ausgeführt.

Parameter	Deklaration	Datentyp	Speicherbereich	Beschreibung
BUSY (Seite 702)	Output	BOOL	E, A, M, D, L	Statusparameter <ul style="list-style-type: none"> BUSY = 0: Die Verarbeitung von TMAIL_C wurde angehalten. BUSY = 1: Die Übertragung der E-Mail ist noch nicht beendet.
ERROR (Seite 702)	Output	BOOL	E, A, M, D, L	Statusparameter <ul style="list-style-type: none"> ERROR = 0: Kein Fehler aufgetreten. ERROR = 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS (Seite 706)	Output	WORD	E, A, M, D, L	Statusparameter Rückgabewert oder Fehlerinformationen der Anweisung TMAIL_C (siehe Parameterbeschreibung).

Hinweis**Optionale Parameter**

Die Anweisung sendet die optionalen Parameter CC, TEXT und ATTACHMENT nur, wenn die entsprechenden Parameter eine Zeichenkette mit einer Länge > 0 enthalten.

Parameter MAIL_ADDR_PARAM

Am Parameter MAIL_ADDR_PARAM definieren Sie die Verbindung zum Senden der E-Mail und speichern die Adresse des E-Mailservers und Anmeldedetails.

Der Systemdatentyp (SDT), den Sie am Parameter MAIL_ADDR_PARAM verwenden, ist von dem Format abhängig, in dem der E-Mailserver adressiert werden muss:

SDT	Beschreibung	Schnittstellenunterstützung
TMail_V4	Adressierung über IP-Adresse (IPv4)	CPU und CP
TMail_V6	Adressierung über IP-Adresse (IPv6)	CP
TMail_FQDN	Adressierung über vollqualifizierten Domänennamen (FQDN)	CP
TMail_V4_SEC	Sichere Adressierung über IP-Adresse (IPv4)	CPU und CP
TMail_V6_SEC	Sichere Adressierung über IP-Adresse (IPv6)	CP
TMail_QDN_SEC	Sichere Adressierung über vollqualifizierten Domänennamen (FQDN)	CPU und CP

TMail_V4: Adressierung des Mailservers über IP-Adresse (IPv4)

Parameter	Datentyp	Beschreibung
TMail_v4	Struct	
Interfaceld	LADDR	Hardware-Kennung der Ethernet-Schnittstelle
ID	CONN_OUC	Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#20 als Verbindungstyp für IPv4.
ActiveEstablished	BOOL	Aktiver/passiver Verbindungsaufbau. Die CPU ist immer der SMTP-Client.
CertIndex	BYTE	= 0: SMTP wird verwendet (S imple M ail T ransfer P rotocol). SMTP muss verwendet werden, wenn die E-Mail über die Schnittstelle einer S7-1200 CPU gesendet wird. ≠0: SMTPS wird verwendet, um die Verbindung abzusichern (CP-Schnittstelle)
WatchDogTime	TIME	Ausführungszeitüberwachung. Mit diesem Parameter kann die maximale Ausführungszeit für das Senden festgelegt werden. Dieser Wert bestimmt, wie lange die Anweisung TMAIL_C in der Ausführung bleiben kann, bevor eine Zeitüberschreitung erreicht und die Ausführung der Anweisung TMAIL_C gestoppt wird. Ab Version 6.0 der Anweisung TMAIL_C ist jetzt ein WatchDogTime-Wert 0 zulässig, womit der Anweisung TMAIL_C signalisiert wird, die Zeit während der Ausführung zu deaktivieren. Sie können für WatchDogTime weiterhin einen Wert ungleich 0 eingeben, um die Ausführung der Anweisung TMAIL_C deterministischer zu machen. Hinweis: Bei langsamen Verbindungen kann der Verbindungsaufbau länger dauern (ca. eine Minute). Berücksichtigen Sie bei Angabe des Parameters WatchDogTime einen Zeitraum für den Verbindungsaufbau.
MailServerAddress	IP_v4	IP-Adresse des Mailservers. IPv4 im folgenden Format: XXX.XXX.XXX.XXX (dezimal). Beispiel: 192.142.131.237
UserName	STRING[254]	Name für die Anmeldung am Mailserver
PassWord	STRING[254]	Passwort für den Mailserver
From	EMAIL_ADDR	Adresse des Absenders der E-Mail, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Lokaler Teil der Absenderadresse einschließlich @-Zeichen. Beispiel: "myname@".
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (Abkürzung FQDN) des Mailservers. Beispiel: "mymailserver.com".

Wenn Sie die Schnittstelle der S7-1200 CPU mit Firmwareversion V4.3 verwenden, müssen Sie den SDT TMail_V4 verwenden. In diesem Fall können Sie die E-Mail nur mit SMTP senden. Die E-Mails sind nicht gesichert.

TMail_V6: Adressierung des Mailservers über IP-Adresse (IPv6)

Parameter	Datentyp	Beschreibung
TMail_V6	Struct	
Interfaceld	LADDR	Hardwarekennung der Schnittstelle
ID	CONN_OUC	Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#21 als Verbindungstyp für IPv6.
ActiveEstablished	BOOL	Statusbit. Wird auf "1" gesetzt, sobald die Verbindung hergestellt ist.
CertIndex	BYTE	<p>=0: SMTP wird verwendet (Simple Mail Transfer Protocol). SMTP muss verwendet werden, wenn die E-Mail über die Schnittstelle einer S7-1500 CPU gesendet wird.</p> <p>≠0: SMTPS wird verwendet, um die Verbindung abzusichern, bevor sie hergestellt ist (mit CPs/CMs). Sie verwenden den Parameter CertIndex, um das zu verwendende Zertifikat anzugeben (siehe "Projektnavigation" > "Globale Sicherheitseinstellungen" > "Zertifikatsmanager").</p>
WatchDogTime	TIME	<p>Zeitüberwachung während der Ausführung. Legen Sie mit diesem Parameter die maximale Ausführungszeit für den Sendevorgang fest.</p> <p>Hinweis: Wenn die Verbindung langsam ist, kann der Verbindungsaufbau länger dauern (ca. eine Minute). Berücksichtigen Sie bei Angabe des Parameters WATCH_DOG_TIME einen Zeitraum für den Verbindungsaufbau.</p> <p>Die Verbindung wird beendet, sobald die angegebene Zeit abgelaufen ist.</p>
MailServerAddress	IP_V6	<p>IP-Adresse des Mailservers (IPv6) im folgenden Format: XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX.XXXX (hexadezimal).</p> <p>Die Adresse wird in 8 Blöcke mit jeweils 2 Byte unterteilt (16 Byte insgesamt).</p> <p>Beispiel: 2001:db8:1f11:08d3:290:27ff:0370:2093</p>
UserName	STRING[254]	Anmeldename für den Mailserver
PassWord	STRING[254]	Passwort für den Mailserver
From	EMAIL_ADDR	E-Mail-Adresse des Absenders, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@my-mailserver.com".
LocalPartPlusAt-Sign	STRING[64]	Lokaler Bestandteil der Absenderadresse, einschließlich @-Zeichen. Beispiel: "myname@"
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abgekürzt FQDN) des Mailservers. Beispiel: "mymailserver.com"

TMail_FQDN : Adressierung des Mailservers überFQDN

Parameter	Datentyp	Beschreibung
TMail_FQDN	Struct	
Interfaceld	LADDR	Hardware-Kennung der Ethernet-Schnittstelle
ID	CONN_OUC	Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#22 als Verbindungstyp für FQDN.
ActiveEstablished	BOOL	Aktiver/passiver Verbindungsaufbau. Ein Kommunikationsprozessor (CP) ist immer der SMTP-Client.
CertIndex	BYTE	= 0: SMTP wird verwendet (S imple M ail T ransfer P rotocol). ≠0: SMTPS wird verwendet, um die Verbindung abzusichern, bevor sie hergestellt ist (mit CP-Schnittstelle). CertIndex gibt das zu verwendende Zertifikat an.
WatchDogTime	TIME	Ausführungszeitüberwachung. Mit diesem Parameter kann die maximale Ausführungszeit für das Senden festgelegt werden. Hinweis: Bei langsamen Verbindungen kann der Verbindungsaufbau länger dauern (ca. eine Minute). Berücksichtigen Sie bei Angabe des Parameters WatchDogTime einen Zeitraum für den Verbindungsaufbau.
MailServerAddress	STRING[254]	FQDN (F ully Q ualified D omain N ame, vollqualifizierter Domänenname) des Mailservers. Der Mailserver wird über den FQDN adressiert. Beispiel: "www.mymailserver.com."
UserName	STRING[254]	Name für die Anmeldung am Mailserver
PassWord	STRING[254]	Passwort für den Mailserver
From	Struct	Adresse des Absenders der E-Mail, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Lokaler Bestandteil der Absenderadresse, einschließlich @-Zeichen. Beispiel: "myname@".
FullQualifiedDomainName	STRING[254]	Vollqualifizierter Domänenname (Abkürzung FQDN) des Mailservers. Beispiel: "mymailserver.com".

TMail_V4_SEC: Adressierung des Mailservers über IP-Adresse (IPv4)

Parameter	Datentyp	Beschreibung
TMail_V4_SEC	Struct	
Interfaceld	LADDR	Hardware-Kennung des Wertebereichs der Ethernet-Schnittstelle: <ul style="list-style-type: none"> • 0 (neu): Das Betriebssystem wählt selbst einen geeigneten integrierten Port aus. • Hardware-Kennung des zu verwendenden integrierten Ports.
ID	CONN_OUC	Verweis auf die Verbindung: Wertebereich: <ul style="list-style-type: none"> • 0 (neu): Das Betriebssystem wählt eine freie Verbindungs-ID aus dem internen Bereich aus. • 1 bis 4095: Die zu verwendende Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#20 als Verbindungstyp für IPv4.
ActiveEstablishment	BOOL	Aktiver/passiver Verbindungsaufbau. Die CPU ist immer der SMTP-Client.
WatchDogTime	TIME	Zeitüberwachung während der Ausführung. Legen Sie mit diesem Parameter die maximale Ausführungszeit für den Sendevorgang fest. Dieser Wert bestimmt, wie lange die Anweisung TMAIL_C in der Ausführungsphase bleiben kann, bevor eine Zeitüberschreitung erreicht und die Ausführung der Anweisung TMAIL_C gestoppt wird. Ab Version 6.0 der Anweisung TMAIL_C ist jetzt ein WatchDogTime-Wert 0 zulässig, womit der Anweisung TMAIL_C signalisiert wird, die Zeit während der Ausführung zu deaktivieren. Sie können für WatchDogTime weiterhin einen Wert ungleich 0 eingeben, um die Ausführung der Anweisung TMAIL_C deterministischer zu machen. Hinweis: Der Verbindungsaufbau kann länger dauern (ca. eine Minute), wenn die Verbindung langsam ist. Berücksichtigen Sie bei Angabe des Parameters WatchDogTime einen Zeitraum für den Verbindungsaufbau.
MailServerAddress	IP_V4	IP-Adresse des Mailservers intrinsisch im IPv4-Format: XXX.XXX.XXX.XXX (dezimal) Beispiel: 192.142.131.237
UserName	STRING[254]	Benutzername Sie müssen für den Zugriff auf Ihr E-Mail-Postfach Ihren "Benutzernamen" verwenden, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
PassWord	STRING[254]	Benutzerpasswort Sie müssen für den Zugriff auf Ihr E-Mail-Postfach Ihr "Passwort" verwenden, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
From	EMAIL_ADDR	Adresse des Absenders der E-Mail, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@mymailserver.com"

Parameter	Datentyp	Beschreibung
LocalPartPlusAtSign	STRING[64]	Lokaler Bestandteil der Absenderadresse, einschließlich @-Zeichen. Beispiel: "myname@"
FullQualifiedDomainName	STRING[254]	Vollqualifizierter Domänenname (Abkürzung FQDN) des Mail-servers. Beispiel: "mymailserver.com"
RemotePort	UINT	TCP-Port des Mailservers
ActivateSecureConn	BOOL	0: SMTP-Verbindung (nicht gesichert). In diesem Fall sind die folgenden Parameter irrelevant. 1: Sichere SMTP-Verbindung
ExtTLSCapabilities	BYTE	Derzeit nicht verwendet
TLSServerCertRef	UDINT	Verweis auf das Zertifikat X.509 V3 (CA) des Mailservers, mit dem der TLS-Client die Authentifizierung des TLS-Servers validiert

TMail_V6_SEC: Adressierung des Mailservers über die IP-Adresse im IPv6-Format

Parameter	Datentyp	Beschreibung
TMail_V6_SEC	Struct	
Interfaceld	LADDR	Hardware-Kennung der Ethernet-Schnittstelle
ID	CONN_OUC	Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#21 als Verbindungstyp für IPv6.
ActiveEstablishment	BOOL	Aktiver/passiver Verbindungsaufbau. Da der CP stets der SMTP-Client ist, muss dieser Parameter auf "1" gesetzt werden.
WatchDogTime	TIME	Zeitüberwachung während der Ausführung. Legen Sie mit diesem Parameter die maximale Ausführungszeit für den Sendevorgang fest. Hinweis: Wenn die Verbindung langsam ist, kann der Verbindungsaufbau länger dauern (ca. eine Minute). Berücksichtigen Sie bei Angabe des Parameters WatchDogTime einen Zeitraum für den Verbindungsaufbau. Die Verbindung wird beendet, sobald die angegebene Zeit abgelaufen ist.
MailServerAddress	IP_V6	IP-Adresse des Mailservers im IPv6-Format: XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX (hexadezimal). Die Adresse wird in 8 Blöcke mit jeweils 2 Byte unterteilt (16 Byte insgesamt). Beispiel: 2001:db8:1f11:08d3:290:27ff:0370:2093
UserName	STRING[254]	Benutzername. Ist für den Zugriff auf das E-Mail-Postfach erforderlich, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
PassWord	STRING[254]	Benutzerpasswort. Ist für den Zugriff auf das E-Mail-Postfach erforderlich, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
From	EMAIL_ADDR	E-Mail-Adresse des Absenders, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@mymailserver.com".
LocalPartPlusAtSign	STRING[64]	Lokaler Bestandteil der Absenderadresse, einschließlich @-Zeichen. Beispiel: "myname@"
FullQualifiedDomainName	STRING[254]	Fully Qualified Domain Name (abgekürzt FQDN) des Mailservers. Beispiel: "mymailserver.com"

Parameter	Datentyp	Beschreibung
RemotePort	UINT	TCP-TCP-Port des Mailservers
ActivateSecureConn	BOOL	0: SMTP-Verbindung (nicht gesichert). In diesem Fall sind die folgenden Parameter irrelevant. 1: Sichere SMTP-Verbindung
ExtTLSCapabilities	BYTE	Derzeit nicht verwendet
TLSServerCertRef	UDINT	Verweis auf das Zertifikat X.509 V3 (CA) des Mailservers, mit dem der TLS-Client die Authentifizierung des TLS-Servers validiert.

TMail_QDN_SEC: Adressierung des Mailservers überFQDN

Parameter	Datentyp	Beschreibung
TMail_QDN_SEC	Struct	
Interfaceld	LADDR	Hardware-Kennung des Wertebereichs der Ethernet-Schnittstelle: <ul style="list-style-type: none"> • 0 (neu): Das Betriebssystem wählt selbst einen geeigneten integrierten Port aus. • Hardware-Kennung des zu verwendenden integrierten Ports.
ID	CONN_OUC	Verweis auf die Verbindung: Wertebereich: <ul style="list-style-type: none"> • 0 (neu): Das Betriebssystem wählt eine freie Verbindungs-ID aus dem internen Bereich aus. • 1 bis 4095: Die zu verwendende Verbindungs-ID
ConnectionType	BYTE	Verbindungstyp. Wählen Sie 16#22 als Verbindungstyp für FQDN.
ActiveEstablishment	BOOL	Aktiver/passiver Verbindungsaufbau. Die CPU ist immer der SMTP-Client.
WatchDogTime	TIME	Zeitüberwachung während der Ausführung. Legen Sie mit diesem Parameter die maximale Ausführungszeit für den Sendevorgang fest. Dieser Wert bestimmt, wie lange die Anweisung TMAIL_C in der Ausführungsphase bleiben kann, bevor eine Zeitüberschreitung erreicht und die Ausführung der Anweisung TMAIL_C gestoppt wird. Ab Version 6.0 der Anweisung TMAIL_C ist jetzt ein WatchDogTime-Wert 0 zulässig, womit der Anweisung TMAIL_C signalisiert wird, die Zeit während der Ausführung zu deaktivieren. Sie können für WatchDogTime weiterhin einen Wert ungleich 0 eingeben, um die Ausführung der Anweisung TMAIL_C deterministischer zu machen. Hinweis: Der Verbindungsaufbau kann länger dauern (ca. eine Minute), wenn die Verbindung langsam ist. Berücksichtigen Sie bei Angabe des Parameters WatchDogTime einen Zeitraum für den Verbindungsaufbau.
MailServerQDN	STRING[254]	FQDN (F ully Q ualified D omain N ame, vollqualifizierter Domänenname) des Mailservers. Der Mailserver wird über den FQDN adressiert, der mit einem "." enden muss. Beispiel: "www.mymailserver.com."
UserName	STRING[254]	Benutzername Sie müssen für den Zugriff auf Ihr E-Mail-Postfach Ihren "Benutzernamen" verwenden, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
PassWord	STRING[254]	Benutzerpasswort Sie müssen für den Zugriff auf Ihr E-Mail-Postfach Ihr "Passwort" verwenden, um sich selbst gegenüber dem E-Mail-Anbieter als Eigentümer des E-Mail-Postfachs zu identifizieren.
From	EMAIL_ADDR	Adresse des Absenders der E-Mail, die über die beiden folgenden Parameter STRING festgelegt wird. Beispiel: "myname@mymailserver.com"

Parameter	Datentyp	Beschreibung
LocalPartPlusAtSign	STRING[64]	Lokaler Teil der Absenderadresse einschließlich @-Zeichen. Beispiel: "myname@"
FullQualifiedDomainName	STRING[254]	Vollqualifizierter Domänenname (Abkürzung FQDN) des Mail-servers. Beispiel: "mymailserver.com"
RemotePort	UINT	TCP-Port des Mailservers
ActivateSecureConn	BOOL	0: SMTP-Verbindung (nicht gesichert). In diesem Fall sind die folgenden Parameter irrelevant. 1: Sichere SMTP-Verbindung
ExtTLSCapabilities	BYTE	Derzeit nicht verwendet
TLSServerCertRef	UDINT	Verweis auf das Zertifikat X.509 V3 (CA) des Mailservers, mit dem der TLS-Client die Authentifizierung des TLS-Servers validiert

Parameter TO_S und CC

Bei TMAIL_C-Anweisungen vor Version 6.0 und S7-1200 CPU-Firmware V4.4 gelten bei der Eingabe der Parameter TO_S und CC die folgenden Regeln:

- Geben Sie vor jeder Adresse ein Leerzeichen und eine öffnende spitze Klammer "<" ein.
- Geben Sie nach jeder Adresse eine schließende spitze Klammer ">" ein.
- Geben Sie zwischen den Adressen in TO und CC ein Komma ein.

Nachfolgend finden Sie Beispiele für die Parameterzeichenketten TO_S und CC:

- <wenna@mydomain.com>, <ruby@mydomain.com>
- <admin@mydomain.com>, <judy@mydomain.com>

Bei der TMAIL_C-Anweisung der Version 6.0 und S7-1200 CPU-Firmware ab V4.4 gilt bei Eingabe der Parameter nur die folgende Regel:

- Geben Sie zwischen den Adressen in TO und CC ein Komma oder ein Semikolon ein.

Nachfolgend finden Sie Beispiele für die Parameterzeichenketten TO_S und CC:

- wenna@mydomain.com, ruby@mydomain.com
- admin@mydomain.com, judy@mydomain.com

Aus Gründen der Runtime und des Speicherplatzes führt die Anweisung TMAIL_C keine Syntaxprüfung der Parameter TO_S und CC durch.

Parameter DONE, BUSY und ERROR

Die Ausgangsparameter DONE, BUSY und ERROR werden jeweils nur für einen Zyklus angezeigt, wenn der Status des Ausgangsparameters BUSY von "1" zu "0" wechselt.

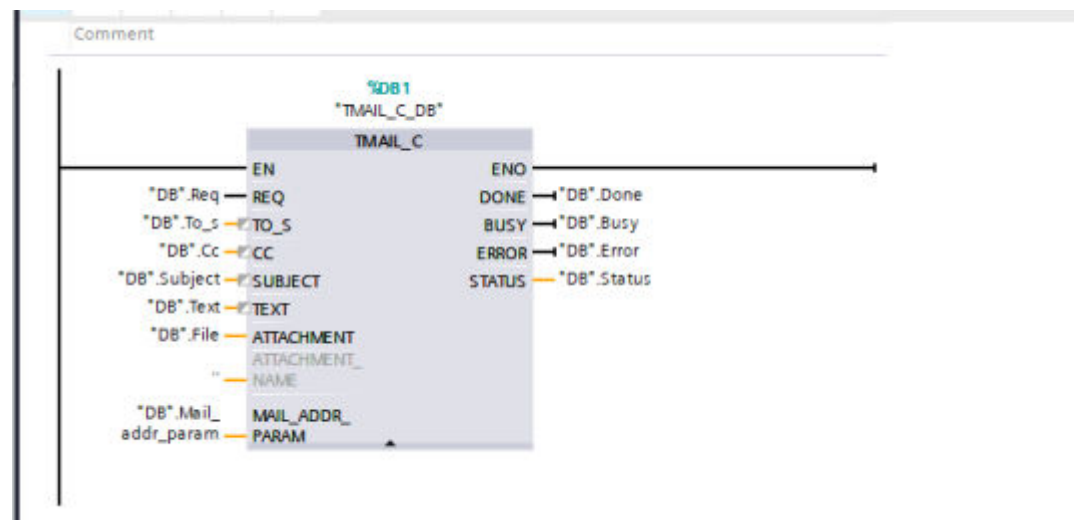
Die folgende Tabelle zeigt die Beziehungen zwischen DONE, BUSY und ERROR. Mithilfe dieser Tabelle können Sie den aktuellen Zustand der Anweisung TMAIL_C ermitteln und den Zeitpunkt feststellen, zu dem der Sendevorgang der E-Mail beendet ist.

DONE	BUSY	ERROR	Beschreibung
0	1	0	Auftrag wird bearbeitet.
1	0	0	Auftrag erfolgreich ausgeführt.
0	0	1	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS (Seite 706) hinterlegt.
0	0	0	Der Anweisung TMAIL_C wurde kein (neuer) Auftrag zugewiesen.

Senden von Datenprotokollen, Rezepten und Benutzerdateien in E-Mail-Anhängen

Ab TMAIL_C Version V6.0 und mit einer S7-1200 CPU-Firmware ab Version V4.4 können Sie über den Parameter Attachment der Anweisung TMAIL_C den SDT TMail_FileReference hinzufügen und auf ihn zugreifen. Sie können dann einen Dateipfad auf der SIMATIC Memory Card (SMC) adressieren. Wenn keine Memory Card vorhanden ist, können Sie trotzdem auf die Rezept- und Datenprotokollverzeichnisse auf dem internen Ladespeicher der PLC zugreifen.

Der SDT TMail_FileReference verwendet automatisch den Parameter FileName für den Parameter AttachmentName.



SDT TMail_FileReference

Der SDT TMail_FileReference besteht aus zwei Parametern und bei beiden handelt es sich um SIMATIC-Zeichenketten:

- Im Parameter DirectoryPath können Sie das Verzeichnis der gewünschten Datei adressieren.
- Der Parameter FileName gibt den Namen der Datei und die Erweiterung (sofern zutreffend) der Datei an, auf die Sie in dem Verzeichnis, das vom vorherigen Parameter angegeben wird, zugreifen möchten.

Name	Datentyp	Startwert	Kommentar
1	Static		
2	File	TMail_FileReference	
3	DirectoryPath	String [254]	'DataLogs/' relative directory of path (ex. 'DataLogs/')
4	FileName	String [254]	'datalog.csv' file name with extension (ex. 'datalog.csv')

Bei der Anweisung TMAIL_C können zur Adressierung des Parameters DirectoryPath nur die Verzeichnisse DataLogs, Recipes oder UserFiles verwendet werden. Innerhalb dieser Verzeichnisse können Sie Unterverzeichnisse adressieren.

Über die oben genannte Beschränkung auf das Basisverzeichnis hinaus werden im nachfolgenden Abschnitt "Regeln für die Adressierung von Dateien" die Regeln beschrieben, die Sie bei der Adressierung von Unterverzeichnissen und Dateinamen einhalten müssen.

Die Größe der Datei, die Sie senden können, wird von der Anweisung TMAIL_C nicht begrenzt. Berücksichtigen Sie dies beim Aufbau Ihres Programms.

Regeln für die Adressierung von Dateien

Es gibt bestimmte Regeln, die einzuhalten sind, um eine Datei mit der Anweisung TMAIL_C über den SDT TMail_FileReference ordnungsgemäß zu adressieren. In den folgenden Unterabschnitten werden jeweils die Einzelheiten für die Parameter DirectoryPath und FileName beschrieben. Im Allgemeinen gelten für beide Parameter im SDT TMail_FileReference folgende Regeln, deren Nichteinhaltung dazu führt, dass die Anweisung TMAIL_C einen Fehlerstatus erzeugt:

- Sie dürfen für ein Unterverzeichnis oder einen Dateinamen keine leere Zeichenkette angeben.
- Sie dürfen in keiner der Parameterzeichenketten ASCII-Steuerzeichen verwenden (Hexadezimalbereich: 0x00 bis 0x1F).
- Sie dürfen in keiner der Parameterzeichenketten die folgenden reservierten Zeichen verwenden:
 - < (kleiner als)
 - > (größer als)
 - : (Doppelpunkt)
 - " (doppelte Anführungszeichen)
 - / (Schrägstrich) (Dieses Zeichen ist im DirectoryPath als Trennzeichen zulässig.)
 - \ (Backslash)
 - | (vertikaler Strich bzw. Pipe)
 - ? (Fragezeichen)
 - * (Sternchen)
- Kein Unterverzeichnis und kein Dateiname darf mit einem Leerzeichen oder einem Punkt enden.

DirectoryPath

Beachten Sie bei Eingabe des gewünschten Verzeichnisses in den Parameter DirectoryPath des SDT die folgenden Hinweise. Das Stammverzeichnis wird durch die Firmwarelogik der PLC abgeleitet, sie brauchen es nicht zu kennen. Sie können optional einen führenden und einen nachfolgenden Schrägstrich (/) eingeben. Wird einer der beiden Schrägstriche nicht eingegeben, fügt die Firmware den Schrägstrich automatisch hinzu. Somit sind alle folgenden Einträge für DirectoryPath gültig:

- /DataLogs/
- /DataLogs
- DataLogs/
- DataLogs

Es ist außerdem möglich, auf einen Pfad mit größerer Tiefe als das Stammverzeichnis zuzugreifen. Herfür verwenden Sie das Format '/DataLogs/dir1/', wobei jeder Schrägstrich (/) ein neues Verzeichnis angibt. Die maximale Tiefe beträgt acht einschließlich Stammverzeichnis.

Neben den im Abschnitt "Regeln für die Adressierung von Dateien" aufgeführten Regeln müssen Sie beachten, dass die Verwendung relativer Pfade streng verboten ist. Somit ist '/DataLogs/' ein unzulässiger Pfadname. Ferner ist die Verwendung eines Punkts in einer Unterverzeichniskomponente zur Darstellung des aktuellen Verzeichnisses verboten (Beispiel: '/DataLogs/').

FileName

Beachten Sie bei Verwendung des Parameters FileName des SDT die im Abschnitt "Regeln für die Adressierung von Dateien" aufgeführten Regeln. Zusätzlich müssen Sie beachten, dass das Betriebssystem der PLC den Dateinamen auf eine Zeichenzahl kleiner als 60 Zeichen begrenzt. Wenn Sie einen Dateinamen mit 60 oder mehr Zeichen adressieren, bricht die Anweisung TMAIL_C ihre Ausführung ab und erzeugt einen Fehler.

Abgesehen von diesen Ausnahmen können Sie jede beliebige Datei unabhängig von Größe oder Dateierweiterung anhängen. Die adressierte Datei kann eine Dateierweiterung enthalten, muss dies aber nicht.

Fehlercodes

Parameter STATUS

Die folgende Tabelle zeigt die Rückgabewerte von TMAIL_C am Parameter STATUS:

Rückgabewert STATUS* (W#16#...):	Bedeutung	Hinweise
0000	Die Ausführung von TMAIL_C wurde fehlerfrei beendet.	Die fehlerfreie Beendigung von TMAIL_C bedeutet nicht, dass die gesendete E-Mail tatsächlich ankommt. Eine fehlerhafte Eingabe der Empfängeradressen erzeugt keinen Statusfehler der Anweisung TMAIL_C. In diesem Fall ist nicht gewährleistet, dass die E-Mail an andere Empfänger gesendet wird, auch wenn diese richtig eingegeben wurden.
7001	TMAIL_C ist aktiv (BUSY = 1).	Erster Aufruf: Auftrag gestartet.
7002	TMAIL_C ist aktiv (BUSY = 1).	Zwischenzeitlicher Aufruf: Auftrag bereits gestartet.
8xxx	Die Ausführung von TMAIL_C wurde mit einem Fehlercode der intern aufgerufenen Kommunikationsanweisungen beendet.	Ausführlichere Informationen enthalten die Beschreibungen des Parameters STATUS für die Kommunikationsanweisungen TCON, TDISCON, TSEND und TRCV (Seite 657).
8009	Fehler einer internen Funktion	Eine interne Funktion hat einen Fehler zurückgegeben. Ausführliche Informationen finden Sie im Parameter SFB_STATUS des Instanz-DB. Die möglichen Werte sind nachfolgend beschrieben.
8010	Fehler beim Verbindungsaufbau	Nähere Informationen zur Auswertung enthält der Parameter SFB_STATUS des Instanzdatenbausteins. Der am Parameter SFB_STATUS angezeigte Fehlercode wird in der Beschreibung des Parameters STATUS für die Anweisung TCON (Seite 657) erläutert.
8011	Fehler beim Senden der Daten	Nähere Informationen zur Auswertung enthält der Parameter SFB_STATUS des Instanzdatenbausteins. Der am Parameter SFB_STATUS angezeigte Fehlercode wird in der Beschreibung des Parameters STATUS für die Anweisung TSEND (Seite 657) erläutert.
8012	Fehler beim Empfangen der Daten	Nähere Informationen zur Auswertung enthält der Parameter SFB_STATUS des Instanzdatenbausteins. Der am Parameter SFB_STATUS angezeigte Fehlercode wird in der Beschreibung des Parameters STATUS für die Anweisung TRCV (Seite 657) erläutert.

Rückgabewert STATUS* (W#16#...):	Bedeutung	Hinweise
8013	Fehler beim Verbindungsaufbau	Nähere Informationen zur Auswertung enthält der Parameter SFB_STATUS des Instanzdatenbausteins. Der am Parameter SFB_STATUS angezeigte Fehlercode wird in der Beschreibung des Parameters STATUS für die Anweisungen TCON (Seite 657) und TDISCON (Seite 657) erläutert.
8014	Verbindungsaufbau nicht möglich.	Es wurde möglicherweise eine falsche Mailserver-IP-Adresse (MailServerAddress (Seite 693)) oder eine zu kurze Zeit (WatchDogTime (Seite 693)) für den Verbindungsaufbau eingegeben. Es ist auch möglich, dass die CPU keine Verbindung zum Netzwerk hat oder die CPU-Konfiguration fehlerhaft ist.
8015	Falscher Datentyp für MAIL_ADDR_PARAMETER	Gültige Datentypen sind nur die Systemdatentypen (Strukturen) Tmail_v4 und TMail_FQDN.
8016	Falscher Datentyp für Parameter ATTACHMENT	Die gültigen Datentypen finden Sie in der folgenden Liste: <ul style="list-style-type: none"> • ArrayOfChar • ArrayOfByte • ArrayOfWord • ArrayOfDWord • String Hinweis: Die Datentypen ArrayOfChar und String sind erst ab Version 5.0 der Anweisung TMAIL_C gültig.
8017	Falsche Datenlänge für Parameter ATTACHMENT	Die Datenlänge muss ≤ 65534 Byte betragen.
82xx, 84xx, oder 85xx	Die Fehlermeldung kommt vom Mailserver und entspricht, mit Ausnahme der "8", der Fehlernummer im SMTP-Protokoll. In den folgenden Zeilen sind mögliche Fehlercodes aufgeführt.	Ausführlichere Informationen über den SMTP-Fehlercode und andere Fehlercodes im SMTP-Protokoll finden Sie im Internet oder in der Fehlerdokumentation des Mailservers. Außerdem können Sie die letzte Fehlermeldung des Mailervers in Ihrem Instanz-DB im Parameter BUFFER1 anzeigen. Die zuletzt mit der Anweisung TMAIL_C gesendeten Daten finden Sie unter DATEN im Instanz-DB.
8450	Vorgang nicht ausgeführt: Mailbox nicht verfügbar/nicht erreichbar	Wiederholen Sie den Vorgang später.
8451	Vorgang abgebrochen: Lokaler Verarbeitungsfehler	Wiederholen Sie den Vorgang später.
8500	Syntaxfehler: Fehler nicht erkannt. Dies betrifft auch Fehler durch eine zu lange Befehlszeichenkette. Der Fehler kann auch auftreten, wenn der E-Mailserver den LOGIN-Authentifizierungsvorgang nicht unterstützt.	Prüfen Sie die Parameter von TMAIL_C. Versuchen Sie, eine E-Mail ohne Authentifizierung zu senden. Ersetzen Sie hierfür den Inhalt von Parameter UserName durch eine leere Zeichenkette. Wird kein Benutzername angegeben, wird das Authentifizierungsverfahren LOGIN nicht verwendet.

Rückgabewert STATUS* (W#16#...):	Bedeutung	Hinweise
8501	Syntaxfehler: Falsche Eingabe an einem Parameter	Mögliche Ursache: Falsche Adresse in Parameter TO_S oder CC (siehe auch: Parameter TO_S und CC (Seite 702)).
8502	Befehl ist unbekannt oder nicht implementiert	Prüfen Sie Ihre Eingaben, insbesondere den Parameter FROM. Möglicherweise ist er unvollständig und Sie haben "@" oder "." vergessen (siehe auch: Parameter TO_S und CC (Seite 702)).
8535	SMTP-Authentifizierung unvollständig	Möglicherweise wurde ein falscher Benutzername oder ein falsches Passwort eingegeben.
8550	Mailserver nicht erreichbar. Sie haben keine Zugriffsrechte.	Sie haben einen falschen Benutzernamen oder ein falsches Passwort eingegeben oder der Mailserver unterstützt Ihre Anmeldung nicht. Eine andere Fehlerursache kann ein Fehler im Domainnamen nach dem "@" in Parameter TO_S oder CC sein (siehe auch: Parameter TO_S und CC (Seite 702)).
8552	Vorgang abgebrochen: Zugewiesener Speicherplatz überschritten	Wiederholen Sie den Vorgang später.
8554	Übertragung fehlgeschlagen	Wiederholen Sie den Vorgang später.
* Fehlercodes können im Programmiereditor als Ganzzahlen oder Hexadezimalwerte angezeigt werden.		

Parameter SFB_STATUS des Instanz-DB

Ab Version 4.4 der S7-1200 CPU-Firmware mit Version 6.0 oder höher der Anweisung TMAIL_C sind die folgenden Rückgabewerte am Parameter SFB_STATUS des Instanz-DB möglich:

Rückgabewerte am Parameter SFB_STATUS des Instanz-DB (W#16#...)	Bedeutung
8085	Die Verbindungs-ID (Parameter ID) wird bereits von einer konfigurierten Verbindung verwendet.
8086	Der Parameter ID liegt außerhalb des gültigen Bereichs.
8087	Maximale Anzahl an Verbindungen erreicht; keine weitere Verbindung möglich.
8088 *	Die Datei ist nicht vorhanden oder derzeit nicht verfügbar.
8089 *	Die Datei kann nicht geöffnet werden, weil die Anzahl gleichzeitig geöffneter Dateien den Grenzwert des Systems überschritten hat. Bei der S7-1200 beträgt der Grenzwert pro Dateisystem 26 Dateien.
808A *	Der DirectoryPath enthält ein anderes Verzeichnis als DataLogs, Recipes oder UserFiles oder eines der adressierten Unterverzeichnisse verletzt die zuvor beschriebenen Regeln für die Adressierung. Weitere Informationen finden Sie unter Directory-Path (Seite 703).

Rückgabewerte am Parameter SFB_STATUS des Instanz-DB (W#16#...)	Bedeutung
808B *	Der FileName enthält eine unzulässige Zeichenkette oder wurde leer gelassen. Weitere Informationen finden Sie unter FileName (Seite 703).
808C *	Der Parameter AttachmentName muss bei der Adressierung eines Dateipfads als Anhang leer sein.
8092	Die Parameter TO_S und CC sind leer oder der Unterparameter "From" ist leer oder unvollständig.
8093	Der Parameter MAIL_ADDR_PARAM erfordert ein Hochrüsten der Verbindung auf eine sichere Verbindung, doch der Mailserver unterstützt den Befehl STARTTLS nicht.
8095	Ungültige Antwort vom Mailserver. Der Mailserver ist möglicherweise nicht RFC-konform.
809A	Die SDT-Struktur am Parameter MAIL_ADDR_PARAM wird an einer integrierten Schnittstelle nicht unterstützt.
809B	Ungültige Schnittstellen-ID im SDT am Parameter MAIL_ADDR_PARAM.
80A1	Die angegebene Verbindung oder der entfernte Port wird bereits verwendet.
80A3	Die ID wird von einer vom Anwenderprogramm erstellten Verbindung verwendet.
80A4	Die IP-Adresse des entfernten Endpunkts der Verbindung ist ungültig oder entspricht der IP-Adresse des lokalen Partners.
80A7	Kommunikationsfehler: Sie haben TDISCON ausgeführt, bevor TMAIL_C beendet war.
80B7	Der entfernte Port ist 0 oder für die IP-Adresse des Partner-Endpunkts wurde 0.0.0.0 eingestellt.
80C3	Ressourcen in der CPU reichen nicht aus
80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> Die Verbindung kann zur Zeit nicht hergestellt werden. Die Verbindung kann nicht hergestellt werden, weil die Firewalls auf dem Verbindungspfad nicht für die erforderlichen Ports geöffnet sind. Die Schnittstelle empfängt gerade neue Parameter. Die konfigurierte Verbindung wird gerade durch eine Anweisung TDISCON entfernt.
80C5	Der Mailserver lehnt den Verbindungsaufbau ab, hat die Verbindung beendet oder aktiv abgebrochen.
80C6	Der Verbindungspartner ist nicht erreichbar (Netzwerkfehler).
80C7	Zeitüberschreitung während der Ausführung
80C8	Es wird versucht, eine bestehende Verbindung erneut herzustellen.
80C9	Validierung des Verbindungspartners fehlgeschlagen. Der Mailserver entspricht nicht dem festgelegten Partner am Parameter MailServerAddress.
80CE	Die IP-Adresse der lokalen Schnittstelle lautet 0.0.0.0.

Rückgabewerte am Parameter SFB_STATUS des Instanz-DB (W#16#...)	Bedeutung
80D0	Der Parameter MailServerAddress enthält bei Verwendung von DNS eine leere Zeichenkette.
80D1	Der Parameter MailServerAddress ist kein vollqualifizierter Domänenname. Möglicherweise fehlt der Punkt am Ende.
80D2	Sie haben keine DNS-Serveradresse konfiguriert.
80D3	Der vollqualifizierte Domänenname (FQDN) konnte nicht aufgelöst werden. Mögliche Ursachen: <ul style="list-style-type: none"> • Der DNS-Server kann nicht erreicht werden (Beispiel: Der DNS-Server wurde heruntergefahren oder der entfernte Port ist nicht erreichbar). • Bei der Kommunikation mit dem DNS-Server ist ein Fehler aufgetreten. • Der DNS-Server hat eine gültige DNS-Antwort zurückgegeben, doch die Antwort enthielt keine IPv4-Adresse.
80E0	Die Kommunikation mit dem Mailserver ist aufgrund einer Meldung mit Fehlern fehlgeschlagen. Mögliche Ursachen: <ul style="list-style-type: none"> • Ungültiger Authentifizierungscode einer Meldung • Meldung konnte nicht entschlüsselt werden • Fehler beim Dekomprimieren einer Meldung • Interner Kapazitätsüberlauf
80E1	Fehler beim Handshake. Mögliche Ursachen: <ul style="list-style-type: none"> • Abbruch durch den Anwender • Sicherheit ist nicht hoch genug. • Verlängerte Negotiation wird nicht unterstützt • SSL/TLS-Version wird nicht unterstützt. • Entschlüsselungsfehler
80E2	Zertifikat wird nicht unterstützt / ist ungültig Mögliche Ursache: Die Uhrzeit des betroffenen Moduls ist nicht eingestellt oder das Modul ist nicht synchronisiert. Beispiel: Die Standardeinstellung für das Datum des Moduls ist 1/1/2012, und das Datum wurde während der Inbetriebnahme nicht eingestellt. Der Gültigkeitszeitraum des Zertifikats beginnt am 20. August 2016 und endet am 20. August 2024. In diesem Fall liegt das Datum des Moduls außerhalb des Gültigkeitszeitraums des Zertifikats, somit ist das Zertifikat für das Modul ungültig.
80E3	Mailserver-Zertifikat wurde verworfen.
80E4	Keine gültige Zertifizierungsstelle für das Mailserver-Zertifikat gefunden.
80E5	Mailserver-Zertifikat ist abgelaufen.
80E6	Integritätsfehler im Transport-Layer-Security-Protokoll
80E7	Nicht unterstützte Erweiterung im Mailserver-Zertifikat
80E9	TLS-Server ohne Serverzertifikat wird nicht unterstützt

* Diese Fehlercodes werden bei nicht ordnungsgemäßer Adressierung des Dateipfads der Anweisung TMAIL_C hinzugefügt, um die Diagnose zu erleichtern.

11.5.8.17 UDP

UDP ist ein im RFC 768 beschriebenes Standardprotokoll und steht für User Datagram Protocol. UDP bietet ein Verfahren, damit eine Anwendung ein Datengramm zu einer anderen Anwendung senden kann. Die Zustellung der Daten ist jedoch nicht garantiert. Dieses Protokoll hat die folgenden Merkmale:

- Schnelles Kommunikationsprotokoll
- Geeignet für kleine bis mittlere Datenmengen (bis 1472 Byte)
- UDP ist ein einfacheres Transportprotokoll als TCP, mit einer dünnen Schicht und wenig Verwaltungsaufwand
- Lässt sich sehr flexibel in vielen Fremdsystemen einsetzen
- Routing-fähig
- Leitet die Datagramme über Portnummern
- Meldungen werden nicht quittiert: Die Anwendung hat die Verantwortung für Wiederherstellung im Fehlerfall und für Sicherheit zu übernehmen
- Für die Datenverwaltung ist aufgrund der Programmierschnittstelle SEND/RECEIVE Programmieraufwand erforderlich

UDP unterstützt die Broadcast-Kommunikation. Um die Broadcast-Funktion zu nutzen, konfigurieren Sie den entsprechenden Teil der IP-Adresse in der ADDR-Konfiguration. Beispiel: Eine CPU mit der IP-Adresse 192.168.2.10 und der Subnetzmaske 255.255.255.0 hätte die Broadcast-Adresse 192.168.2.255.

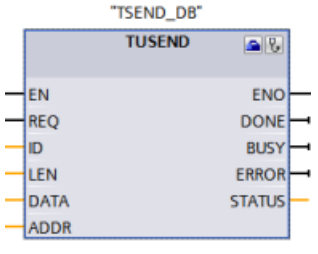
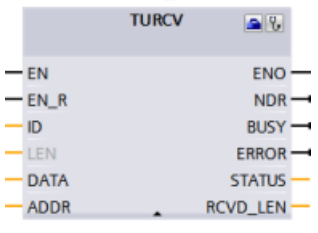
11.5.8.18 TUSEND und TURCV

Die folgenden Anweisungen steuern den UDP-Kommunikationsprozess:

- TCON baut die Kommunikation zwischen dem Client- und dem Server-PC (CPU) auf.
- TUSEND und TURCV senden und empfangen Daten.
- TDISCON trennt die Kommunikation zwischen dem Client und dem Server.

Weitere Informationen zu den Kommunikationsanweisungen TCON und TDISCON finden Sie im Abschnitt "TCP und ISO-on-TCP" unter TCON, TDISCON, TSEND und TRCV (Seite 657).

Tabelle 11-47 Anweisungen TUSEND und TURCV

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Die Anweisung TUSEND sendet Daten über UDP an den entfernten Partner, der vom Parameter ADDR angegeben wird.</p> <p>Um den Auftrag zum Senden von Daten zu starten, rufen Sie die Anweisung TUSEND mit REQ = 1 auf.</p>
	<pre>"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_);</pre>	<p>Die Anweisung TURCV empfängt Daten über UDP. Der Parameter ADDR zeigt die Adresse des Senders an. Nach erfolgreicher Durchführung von TURCV enthält der Parameter ADDR die Adresse des entfernten Partners (des Senders).</p> <p>TURCV unterstützt den Ad-hoc-Modus nicht.</p> <p>Um den Auftrag zum Empfangen von Daten zu starten, rufen Sie die Anweisung TURCV mit EN_R = 1 auf.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

TCON, TDISCON, TUSEND und TURCV laufen asynchron ab, so dass die Bearbeitung eines Auftrags mehrere Anweisungsausführungen umfasst.

Tabelle 11-48 Datentypen für die Parameter von TUSEND und TURCV

Parameter und Datentyp	Datentyp	Beschreibung
REQ (TUSEND)	IN	Bool
EN_R (TURCV)	IN	Bool
ID	IN	Word
LEN	IN	UDInt

Startet den Sendeauftrag bei einer steigenden Flanke. Die Daten werden aus dem mit DATA und LEN bezeichneten Bereich übertragen.

- 0: CPU kann nicht empfangen.
- 1: Aktiviert die CPU für den Empfang. Die Anweisung TURCV ist empfangsbereit und der Empfangsauftrag wird bearbeitet.

Verweis auf die zugehörige Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.
Wertebereich: W#16#0001 bis W#16#0FFF.

Anzahl der Bytes, die gesendet (TUSEND) oder empfangen (TURCV) werden sollen.

- Standard = 0. Der Parameter DATA legt die Länge der zu senden oder zu empfangenden Daten fest.
- Ansonsten, Wertebereich: 1 bis 1472

Parameter und Datentyp		Datentyp	Beschreibung
DONE (TUSEND)	IN	Bool	Statusparameter DONE (TUSEND): <ul style="list-style-type: none"> 0: Auftrag noch nicht gestartet oder läuft noch. 1: Auftrag fehlerfrei ausgeführt.
NDR (TURCV)	OUT	Bool	Statusparameter NDR (TURCV): <ul style="list-style-type: none"> 0: Auftrag noch nicht gestartet oder läuft noch. 1: Auftrag wurde erfolgreich ausgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 1: Auftrag noch nicht beendet. Es kann kein neuer Auftrag gestartet werden. 0: Auftrag ist beendet.
ERROR	OUT	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Kein Fehler 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	OUT	Word	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "BedingungsCodes für Error und Status" unten.)
RCVD_LEN	OUT	UDInt	Anzahl empfangener Bytes (TURCV)
DATA	IN_OUT	Variant	Adresse des Sendebereichs (TUSEND) oder Empfangsbereichs (TURCV): <ul style="list-style-type: none"> das Prozessabbild der Eingänge das Prozessabbild der Ausgänge ein Merkerbit einen Datenbaustein
ADDR	IN_OUT	Variant	Pointer auf die Adresse des Empfängers (bei TUSEND) oder des Senders (bei TURCV) (z. B. P#DB100.DBX0.0 Byte 8) Der Pointer kann auf einen beliebigen Speicherbereich verweisen. Erforderlich ist eine Struktur von 8 Byte, wie folgt: <ul style="list-style-type: none"> Die ersten 4 Byte enthalten die dezentrale IP-Adresse. Die nächsten 2 Byte geben die dezentrale Portnummer an. Die letzten 2 Byte sind reserviert.

Der Auftragszustand wird an den Ausgangsparametern BUSY und STATUS angezeigt. STATUS entspricht dem Ausgangsparameter RET_VAL bei asynchron funktionierenden Anweisungen.

Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE (TUSEND), NDR (TURCV) und ERROR. Mit Hilfe dieser Tabelle können Sie den aktuellen Zustand der Anweisung (TUSEND oder TURCV) ermitteln bzw. den Zeitpunkt, zu dem der Sendevorgang (die Übertragung) bzw. der Empfangsvorgang beendet ist.

Tabelle 11-49 Zustand der Parameter BUSY, DONE (TUSEND)/NDR (TURCV) und ERROR

BUSY	DONE / NDR	ERROR	Beschreibung
WAHR	Irrelevant	Irrelevant	Auftrag wird bearbeitet.
FALSCH	WAHR	FALSCH	Der Auftrag wurde erfolgreich ausgeführt.

BUSY	DONE / NDR	ERROR	Beschreibung
FALSCH	FALSCH	WAHR	Auftrag mit Fehler beendet. Die Fehlerursache ist im Parameter STATUS hinterlegt.
FALSCH	FALSCH	FALSCH	Der Anweisung wurde kein (neuer) Auftrag zugewiesen.

- ¹ Wegen der asynchronen Funktion der Anweisungen: Bei TUSEND müssen die Daten im Sendebereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt. Bei TURCV sind die Daten im Empfangsbereich nur konsistent, wenn der Parameter NDR den Wert WAHR hat.

Tabelle 11-50 Bedingungscode von TUSEND und TURCV für ERROR und STATUS

ERROR	STATUS	Beschreibung
0	0000	<ul style="list-style-type: none"> • Sendeauftrag fehlerfrei ausgeführt (TUSEND). • Neue Daten wurden angenommen. Die aktuelle Länge der empfangenen Daten wird in RCVD_LEN (TURCV) gezeigt.
0	7000	<ul style="list-style-type: none"> • Keine laufende Auftragsbearbeitung (TUSEND) • Baustein nicht empfangsbereit (TURCV)
0	7001	<ul style="list-style-type: none"> • Auftragsbearbeitung starten, Daten werden gesendet (TUSEND): Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu. • Baustein empfangsbereit, Empfangsauftrag wurde aktiviert (TURCV).
0	7002	<ul style="list-style-type: none"> • Folgeaufruf (REQ irrelevant), Auftrag wird bearbeitet (TUSEND): Während dieser Bearbeitung greift das Betriebssystem auf die Daten im Sendebereich DATA zu. • Folgeaufruf, Auftrag wird bearbeitet: Während der Bearbeitung schreibt die Anweisung TURCV Daten in den Empfangsbereich. Deshalb kann ein Fehler zu inkonsistenten Daten im Empfangsbereich führen.
1	8085	Der Parameter LEN ist größer als der größte zulässige Wert, hat den Wert 0 (TUSEND) oder Sie haben den Wert des Parameters LEN oder DATA seit der ersten Anweisungsausführung geändert (TURCV).
1	8086	Der Parameter ID ist nicht im zulässigen Adressbereich.
1	8088	<ul style="list-style-type: none"> • Der Parameter LEN ist größer als der in DATA festgelegte Speicherbereich (TUSEND) oder Empfangsbereich (TURCV). • Empfangsbereich ist zu klein (TURCV).
1	8089	Der Parameter ADDR verweist nicht auf einen Datenbaustein.
1	80A1	<p>Kommunikationsfehler:</p> <ul style="list-style-type: none"> • Die angegebene Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems wurde noch nicht aufgebaut. • Die angegebene Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems wird gerade beendet. Ein Sendeauftrag (TUSEND) oder Empfangsauftrag (TURCV) ist über diese Verbindung nicht möglich. • Schnittstelle wird neu initialisiert.
1	80A4	IP-Adresse des entfernten Verbindungspunkts ist ungültig, sie entspricht möglicherweise der lokalen IP-Adresse (TUSEND).
1	80B3	<ul style="list-style-type: none"> • Die festgelegte Protokollvariante (Parameter connection_type in der Verbindungsbeschreibung) ist kein UDP. Bitte verwenden Sie die Anweisung TSEND oder TRCV. • Parameter ADDR: Ungültige Einstellungen für die Portnummer (TUSEND)

ERROR	STATUS	Beschreibung
1	80C3	<ul style="list-style-type: none"> Ein Baustein mit dieser ID wird bereits in einer anderen Prioritätsklasse bearbeitet. Interner Mangel an Ressourcen
1	80C4	Temporärer Kommunikationsfehler: <ul style="list-style-type: none"> Die Verbindung zwischen dem Anwenderprogramm und der Kommunikationsschicht des Betriebssystems kann derzeit nicht hergestellt werden (TUSEND). Die Schnittstelle empfängt neue Parameter (TUSEND). Die angegebene Verbindung wird derzeit neu aufgebaut (TURCV).

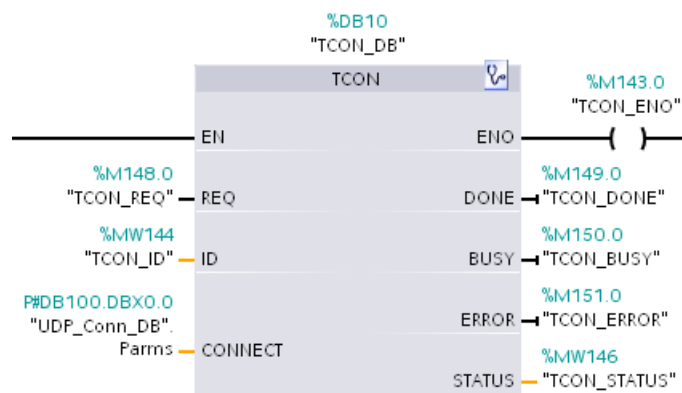
Ethernet-Verbindungsprotokolle

Jede CPU hat einen integrierten PROFINET-Anschluss, der die standardmäßige PROFINET-Kommunikation unterstützt. Die Anweisungen TUSEND und TURCV unterstützen das UDP-Ethernet-Protokoll.

Weitere Informationen finden Sie im Kapitel "Gerätekonfiguration" unter "Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren" (Seite 600).

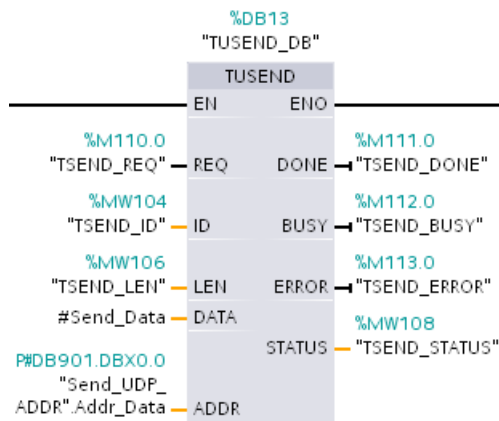
Betrieb

Beide Partner sind in der UDP-Kommunikation passiv. Typische Startwerte für Parameter des Datentyps "TCON_Param" werden in den folgenden Abbildungen gezeigt. Die Portnummern (LOCAL_TSAP_ID) sind im 2-Byte-Format geschrieben. Alle Ports mit Ausnahme von 161, 34962, 34963 und 34964 sind zulässig.



UDP_Conn_DB					
	Name	Datentyp	Offset	Startwert	Kommentar
1	Static				
2	Params	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		8#16#07	
15	LOCAL_TSAP_ID[2]	Byte		8#16#D0	

Die Anweisung TUSEND sendet Daten über UDP an den entfernten Partner, der vom Datentyp TADDR_Param angegeben wird. Die Anweisung TURCV empfängt Daten über UDP. Nach erfolgreicher Ausführung der Anweisung TURCV zeigt der Datentyp "TADDR_Param" Ihnen die Adresse des entfernten Partners (des Senders) an.



Send_UDP_ADDR					
	Name	Datentyp	Offset	Startwert	Kommentar
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	unused; has to be 0

11.5.8.19 T_CONFIG

Die Anweisung T_CONFIG kann die Ethernet-Adresse, den PROFINET-Gerätenamen oder die IP-Adressen der NTP-Server für die Uhrzeitsynchronisation durch das Anwenderprogramm ändern. Die folgenden Funktionen können dauerhaft oder temporär angepasst werden:

- IP-Adresse
- Subnetzmaske
- Router-Adresse
- Stationsname
- IP-Adressen von bis zu vier NTP-Servern

Hinweis

Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "IP-Adresse wird direkt am Gerät eingestellt" (Seite 723) die IP-Adresse online oder über die Anweisung "T_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben.

Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "PROFINET-Gerätename wird direkt am Gerät eingestellt" (Seite 724) den PROFINET-Gerätenamen online oder über die Anweisung "T_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben.

Auf der Seite "Uhrzeitsynchronisation" in den "Eigenschaften" der CPU können Sie über das Feld "Uhrzeitsynchronisation über NTP-Server aktivieren" (Seite 725) die IP-Adressen von bis zu vier NTP-Servern ändern.

Hinweis

Sie können nicht mehrere Anweisungen T_CONFIG gleichzeitig ausführen.

Hinweis

Änderungen an der IP-Adresse oder am Namen der Station der CPU können temporär oder dauerhaft sein. Änderungen an den IP-Adressen der NTP-Server können nur temporär sein:

- Eine dauerhafte Änderung weist darauf hin, dass die Änderung remanent ist, was bedeutet, dass die Änderung auch nach einem Spannungsausfall noch vorhanden ist.
- Eine temporäre Änderung weist darauf hin, dass die Änderung flüchtig ist und nach einem Spannungsverlust wieder der ursprüngliche Wert eingenommen wird.

Tabelle 11-51 Anweisung T_CONFIG


KOP/FUP	SCL	Beschreibung
 <p>Das Diagramm zeigt die Datenbankstruktur für die T_CONFIG-Anweisung. Es besteht aus einem zentralen Kasten 'T_CONFIG' mit den Parametern EN, Req, Interface und Conf_Data auf der linken Seite sowie ENQ, Done, Busy, Error, Status und Err_Loc auf der rechten Seite.</p>	<pre>"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_);</pre>	<p>Die Anweisung T_CONFIG ändert die Parameter der IP-Konfiguration über das Anwenderprogramm. T_CONFIG arbeitet asynchron. Die Ausführung erstreckt sich über mehrere Aufrufe.</p>

Tabelle 11-52 T_CONFIG-Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	Input	Bool	Startet die Anweisung an der steigenden Flanke.
INTERFACE	Input	HW_Interface	ID der Netzwerkschnittstelle
CONF_DATA	Input	Variant	Verweis auf die Struktur der Konfigurationsdaten; CONF_DATA ist über eine Struktur mit bis zu vier Systemdatentypen (SDT) festgelegt.
DONE	Output	Bool	<ul style="list-style-type: none"> 0: Auftrag wurde noch nicht gestartet oder läuft noch. 1: Auftrag wurde fehlerfrei ausgeführt.
BUSY	Output	Bool	<ul style="list-style-type: none"> 0: Auftrag ist beendet. 1: Der Auftrag ist noch nicht beendet. Es kann kein neuer Auftrag gestartet werden.
ERROR	Output	Bool	Statusparameter mit den folgenden Werten: <ul style="list-style-type: none"> 0: Kein Fehler 1: Fehler während der Verarbeitung aufgetreten. STATUS enthält Informationen zum Fehlertyp.
STATUS	Output	DWord	Statusinformationen umfassen Fehlerinformationen. (Siehe Tabelle "Bedingungscode für Error und Status" unten.)
ERR_LOC	Output	DWord	Fehlerstelle (Feld-ID und Unterfeldadresse in der Struktur CONF_DATA)

Die IP-Konfigurationsdaten werden im Datenbaustein CONF_DATA abgelegt, zusammen mit einem Variant-Pointer auf den oben angegebenen Parameter CONF_DATA. Die erfolgreiche Ausführung der Anweisung T_CONFIG endet mit der Übergabe der IP-Konfigurationsdaten an die Netzwerkschnittstelle.

Die Status- und Fehlermeldungen der Anweisung "T_CONFIG" werden an den Parametern STATUS und ERR_LOC ausgegeben:

- Die Fehlerursache wird am Parameter STATUS ausgegeben.
- Die Fehlerstelle wird am Parameter ERR_LOC ausgegeben. Folgende Optionen stehen hier zur Verfügung:
 - 16#0000_0000: Kein Fehler oder Fehler beim Aufrufen der Anweisung (z.B. Fehler beim Parametrieren der Anweisung oder bei der Kommunikation mit der PROFINET-Schnittstelle).
 - 16#0001_0000: Fehler in den Konfigurationsdaten in den Parametern des Systemdatentyps IF_CONF_HEADER.
 - 16#0001_000x: Fehler in den Konfigurationsdaten in den Parametern des Systemdatentyps IF_CONF_V4 oder IF_CONF_NOS oder IF_CONF_NTP (x gibt die Position des fehlerhaften Teilbausteins in der Struktur T_CONFIG an. Wenn die Struktur T_CONFIG beispielsweise einen Teilbaustein für die IP-Adresse und einen Teilbaustein für den Stationsnamen enthält und sich der Fehler im Teilbaustein für den Stationsnamen befindet, hat ERR_LOC den Wert 0001_0002.)

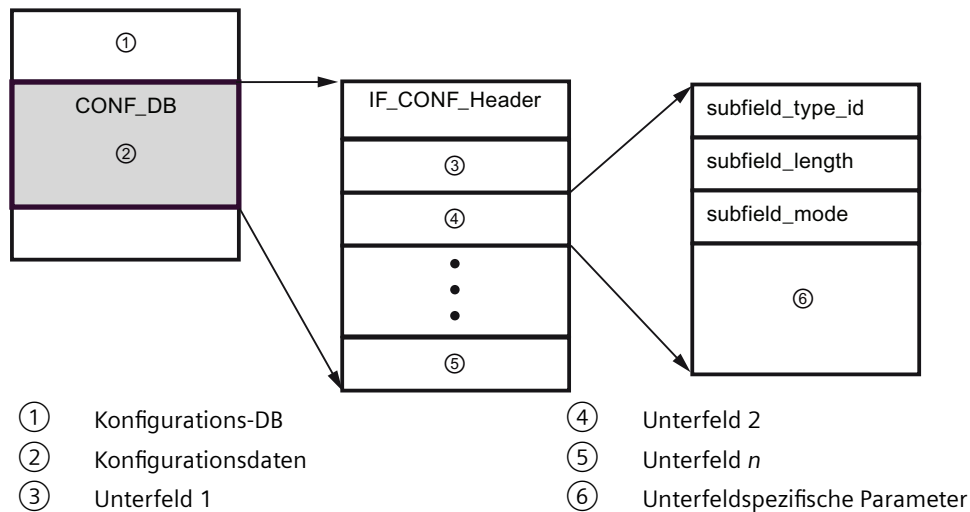
In der folgenden Tabelle werden die möglichen Wert für die Parameter STATUS und ERR_LOC gezeigt:

STATUS*	ERR_LOC*	Bedeutung
0000_0000	0000_0000	Auftragsbearbeitung fehlerfrei durchgeführt.
0070_0000	0000_0000	Keine laufende Auftragsbearbeitung.
0070_0100	0000_0000	Beginn der Auftragsbearbeitung.
0070_0200	0000_0000	Zwischenzeitlicher Aufruf (REQ irrelevant).
C08x_yy00	0000_0000	Allgemeine Fehlerinformationen.
C080_8000	0000_0000	Fehler bei Aufruf der Anweisung: Die Hardware-ID am Parameter Interface ist ungültig.
C080_8100	0000_0000	Fehler bei Aufruf der Anweisung: Die Hardware-ID am Parameter Interface adressiert keine PROFINET-Schnittstelle.
C080_8700	0000_0000	Fehler bei Aufruf der Anweisung: Falsche Länge des Datenbausteins am Parameter CONF_DATA.
C080_8800	0001_0000	Fehler im Systemdatentyp IF_CONF_HEADER: Der Parameter FieldType hat einen ungültigen Wert. Verwenden Sie für FieldType den Wert 0.
C080_8900	0001_0000	Fehler im Systemdatentyp IF_CONF_HEADER: Der Parameter FieldId hat einen ungültigen Wert oder wurde mehrmals verwendet. Verwenden Sie für FieldId den Wert 0.
C080_8A00	0001_0000	Fehler im Systemdatentyp IF_CONF_HEADER: Falsche Zahl am Parameter SubfieldCount. Geben Sie die richtige Nummer der verwendeten Systemdatentypen IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP ein.
C080_8B00	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Der Parameter Id hat einen ungültigen Wert. Verwenden Sie für IF_CONF_V4 den Wert 30, für IF_CONF_NOS den Wert 40, für IF_CONF_NTP den Wert 17.
C080_8C00	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Falscher Systemdatentyp verwendet, falsche Reihenfolge oder mehrmalige Verwendung eines Systemdatentyps.
C080_8D00	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Der Parameter Length hat einen falschen oder ungültigen Wert.
C080_8E00	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Der Parameter Mode hat einen falschen oder ungültigen Wert. <ul style="list-style-type: none"> • Bei IF_CONF_V4 und IF_CONF_NOS sind nur die Werte 1 (dauerhaft) oder 2 (temporär) zulässig. • Bei IF_CONF_NTP ist nur der Wert 2 (temporär) zulässig.
C080_9000	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Konfigurationsdaten können nicht übernommen werden. Mögliche Ursache: <ul style="list-style-type: none"> • Bei IF_CONF_V4: In der Hardwarekonfiguration wurde die Einstellung "IP-Adresse am Gerät einstellen" nicht ausgewählt. • Bei IF_CONF_NOS: In der Hardwarekonfiguration wurde die Einstellung "PROFINET-Gerätenamen am Gerät einstellen" nicht ausgewählt. • Bei IF_CONF_NTP: In der Hardwarekonfiguration wurde die Einstellung "Uhrzeitsynchronisation über NTP-Server aktivieren" nicht ausgewählt und für die NTP-Server wurde keine IP-Adresse eingestellt.

STATUS*	ERR_LOC*	Bedeutung
C080_9400	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Ein Parameterwert ist nicht festgelegt oder ungültig.
C080_9500	0001_000x	Fehler im Systemdatentyp IF_CONF_V4, IF_CONF_NOS oder IF_CONF_NTP: Die Werte von zwei Parametern sind inkonsistent.
C080_C200	0000_0000	Fehler bei Aufruf der Anweisung: Die Konfigurationsdaten können nicht übertragen werden. Mögliche Ursache: Die PROFINET-Schnittstelle ist nicht zugänglich.
C080_C300	0000_0000	Fehler bei Aufruf der Anweisung: Ressourcen nicht ausreichend (z.B. mehrere Aufrufe von "T_CONFIG" mit unterschiedlichen Parametern).
C080_C400	0000_0000	Fehler bei Aufruf der Anweisung: Temporärer Kommunikationsfehler. Uhrzeitanzeige für Wechsel zur Sommerzeit.
C080_D200	0000_0000	Fehler bei Aufruf der Anweisung: Aufruf nicht möglich. Die Anweisung wird von der PROFINET-Schnittstelle nicht unterstützt.

Datenbaustein CONF_DATA

Die folgende Abbildung zeigt, wie die zu übertragenden Konfigurationsdaten im Konfigurations-DB gespeichert werden.



Die Konfigurationsdaten von CONF_DB bestehen aus einem Feld mit einem Header (IF_CONF_Header) und mehreren Unterfeldern. IF_CONF_Header liefert die folgenden Elemente:

- field_type_id (Datentyp UInt): Null
- field_id (Datentyp UInt): Nullabgleich
- subfield_cnt (Datentyp UInt): Anzahl der Unterfelder

Jedes Unterfeld besteht aus einem Header (subfield_type_id, subfield_length, subfield_mode) und den unterfeldspezifischen Parametern. Jedes Unterfeld muss eine gerade Anzahl von Bytes umfassen. subfield_mode kann die Werte 1 und 2 unterstützen (siehe folgende Tabellen).

Hinweis

Derzeit ist nur ein Feld (IF_CONF_Header) zulässig. Dessen Parameter field_type_id und field_id müssen den Wert 0 haben. Andere Felder mit anderen Werten für field_type_id und field_id unterliegen künftigen Erweiterungen.

Tabelle 11-53 Unterstützte Unterfelder

subfield_type_id	Datentyp	Bedeutung
30	IF_CONF_V4	IP-Parameter: IP-Adresse, Subnetzmaske, Routeradresse
40	IF_CONF_NOS	PROFINET IO-Device-Name (Name of station)
17	IF_CONF_NTP	Network Time Protocol (NTP)

Tabelle 11-54 Elemente des Datentyps IF_CONF_V4

Name	Datentyp	Startwert	Beschreibung	
Id	UInt	30	subfield_type_id	
Length	UInt	18	subfield_length	
Mode	UInt	0	subfield_mode (1: Dauerhaft oder 2: Temporär)	
InterfaceAddress	IP_V4	-	Schnittstellenadresse	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	0		High Byte der IP-Adresse: 200
ADDR[2]	Byte	0		High Byte der IP-Adresse: 12
ADDR[3]	Byte	0		Low Byte der IP-Adresse: 1
ADDR[4]	Byte	0	Low Byte der IP-Adresse: 144	
SubnetMask	IP_V4	-	Subnetzmaske	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	0		High Byte der Subnetzmaske: 255
ADDR[2]	Byte	0		High Byte der Subnetzmaske: 255
ADDR[3]	Byte	0		Low Byte der Subnetzmaske: 255
ADDR[4]	Byte	0	Low Byte der Subnetzmaske: 0	
DefaultRouter	IP_V4	-	Standard-Router	
ADDR	Array [1..4] of Byte			
ADDR[1]	Byte	0		High Byte des Routers: 200
ADDR[2]	Byte	0		High Byte des Routers: 12
ADDR[3]	Byte	0		Low Byte des Routers: 1
ADDR[4]	Byte	0	Low Byte des Routers: 1	

Tabelle 11-55 Elemente des Datentyps IF_CONF_NOS

Name	Datentyp	Startwert	Beschreibung
Id	UInt	40	subfield_type_id
Length	UInt	246	subfield_length
Mode	UInt	0	subfield_mode (1: Dauerhaft oder 2: Temporär)
NOS (Name of station)	Array[1..240] of Byte	0	Stationsname: Sie müssen das ARRAY ab dem ersten Byte belegen. Wenn das ARRAY länger als der zuzuweisende Stationsname ist, müssen Sie nach dem eigentlichen Stationsnamen ein Null-Byte eingeben (in Übereinstimmung mit IEC 61158-6-10). Andernfalls wird NOS zurückgewiesen und die Anweisung "T_CONFIG (Seite 717)" gibt den Fehlercode DW#16#C0809400 in STATUS ein. Wenn Sie das erste Byte mit 0 belegen, wird der Stationsname gelöscht.

Der Stationsname unterliegt den folgenden Einschränkungen:

- Eine Namenskomponente innerhalb des Stationsnamens, d.h. eine Zeichenfolge zwischen zwei Punkten, darf nicht länger sein als 63 Zeichen.
- Sonderzeichen wie Umlaute, Klammern, Unterstriche, Schrägstriche, Leerzeichen usw. sind nicht zulässig. Das einzig zulässige Sonderzeichen ist der Bindestrich.
- Der Stationsname darf nicht mit dem Zeichen "-" beginnen oder enden.
- Der Stationsname darf nicht mit einer Zahl beginnen.
- Das Format n.n.n.n (n = 0 ... 999) ist für den Stationsnamen nicht zulässig.
- Der Stationsname darf nicht mit der Zeichenfolge "Port-xyz" oder "Port-xyz-abcde" (a, b, c, d, e, x, y, z = 0 ... 9) beginnen.

Hinweis

Sie können auch ein ARRAY "NOS" erstellen, das kürzer ist als 240 Byte. Die Mindestlänge ist jedoch 2 Byte. Dann müssen Sie die Variable "Length" (Länge des Unterfelds) entsprechend anpassen.

Tabelle 11-56 Elemente des Datentyps IF_CONF_NTP

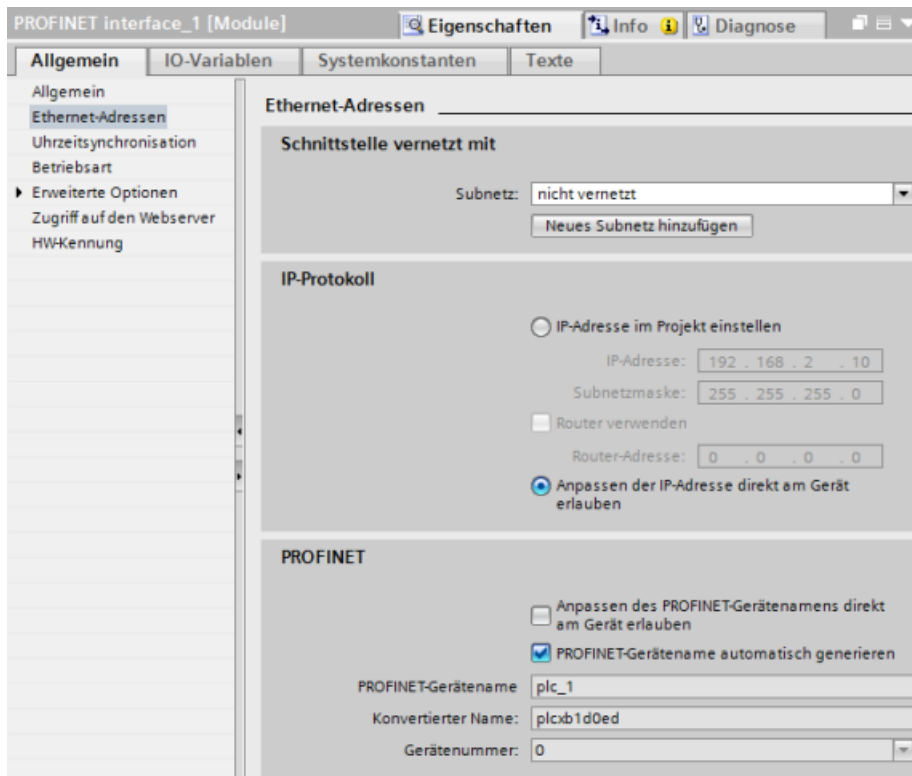
Name	Datentyp	Startwert	Beschreibung		
Id	UInt	17	subfield_type_id		
Length	UInt	22	subfield_length		
Mode	UInt	0	subfield_mode (2: Temporär)		
NTP_IP	Array[1...4] of IP_V4	-	IP-Adressen von NTP-Servern		
	NTP_IP[1]	IP_V4	IP-Adressen von NTP-Server 1		
	ADDR	Array[1...4] of Byte		0	
	ADDR[1]	Byte		0	High Byte der IP-Adresse
	ADDR[2]	Byte		0	High Byte der IP-Adresse
	ADDR[3]	Byte		0	Low Byte der IP-Adresse
	ADDR[4]	Byte	0	Low Byte der IP-Adresse	
	NTP_IP[2]	IP_V4	IP-Adressen von NTP-Server 2		
	ADDR	Array[1...4] of Byte		0	

Name		Datentyp	Startwert	Beschreibung
	ADDR[1]	Byte	0	High Byte der IP-Adresse
	ADDR[2]	Byte	0	High Byte der IP-Adresse
	ADDR[3]	Byte	0	Low Byte der IP-Adresse
	ADDR[4]	Byte	0	Low Byte der IP-Adresse
NTP_IP[3]		IP_V4		IP-Adressen von NTP-Server 3
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	
	ADDR[2]	Byte	0	
	ADDR[3]	Byte	0	
	ADDR[4]	Byte	0	Low Byte der IP-Adresse
NTP_IP[4]		IP_V4		IP-Adressen von NTP-Server 4
	ADDR	Array[1...4] of Byte	0	
	ADDR[1]	Byte	0	
	ADDR[2]	Byte	0	
	ADDR[3]	Byte	0	
	ADDR[4]	Byte	0	Low Byte der IP-Adresse

Beispiel: IP-Parameter mit Anweisung T_CONFIG ändern

Im folgenden Beispiel werden im Unterfeld "addr" die Parameter "InterfaceAddress" (IP-Adresse), "SubnetMask" und "DefaultRouter" (IP-Router) geändert. Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "IP-Adresse wird direkt am Gerät eingestellt" die IP-Parameter über die Anweisung "T_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben.

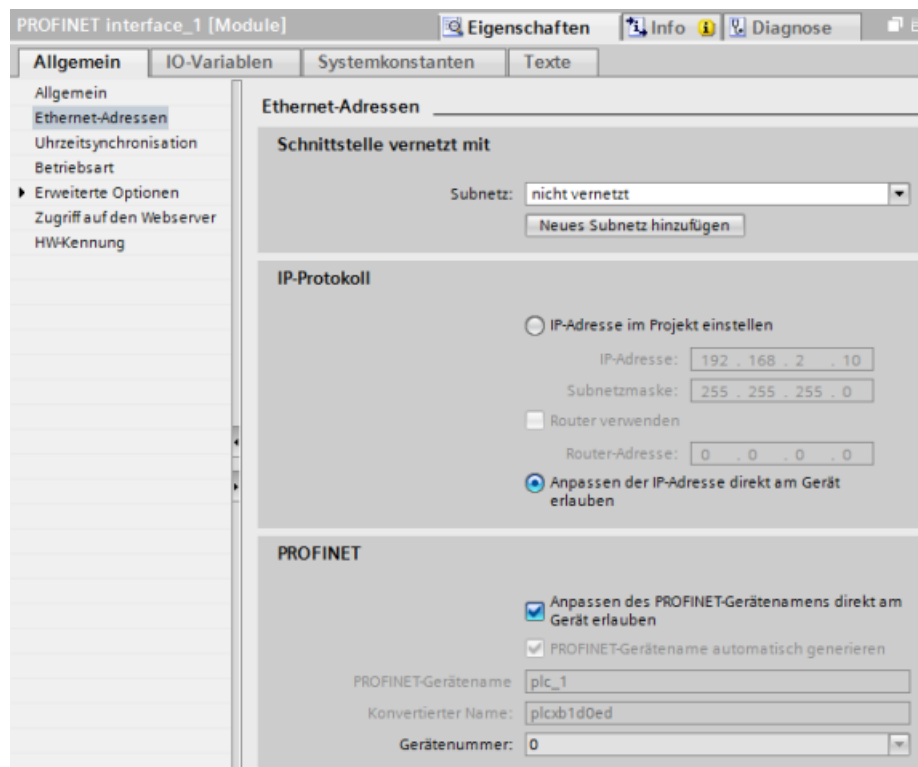
CONF_DATA_1			
	Name	Datentyp	Startwert
1	Static		
2	Conf_data	Struct	
3	header	IF_CONF_Header	
4	FieldType	UInt	0
5	FieldId	UInt	0
6	SubfieldCount	UInt	1
7	addr	IF_CONF_v4	
8	Id	UInt	30
9	Length	UInt	18
10	Mode	UInt	1
11	InterfaceAddress	IP_V4	
12	ADDR	array [1..4] of Byte	
13	ADDR[1]	Byte	192
14	ADDR[2]	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



Beispiel: IP-Parameter und PROFINET IO-Gerätenamen mit Anweisung T_CONFIG ändern

Im folgenden Beispiel werden beide Unterfelder "addr" und "nos" (Name of station) geändert. Auf der Seite "Ethernet-Adresse" in den "Eigenschaften" der CPU können Sie über das Optionsfeld "PROFINET-Gerätename wird direkt am Gerät eingestellt" den PROFINET-Gerätenamen über die Anweisung "T_CONFIG" ändern, nachdem Sie das Programm in die CPU geladen haben.

CONF_DATA_2			
	Name	Datentyp	Startwert
1	Static		
2	Conf_data	Struct	
3	header	IF_CONFIG_Header	
4	FieldType	UInt	0
5	Fieldid	UInt	0
6	SubfieldCount	UInt	2
7	addr	IF_CONFIG_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_CONFIG_NOS	
18	Id	UInt	40
19	Length	UInt	246
20	Mode	UInt	1
21	NOS	array [1..240] of Byte	



Beispiel: Ändern von IP-Adressen in den NTP-Servern über die Anweisung T_CONFIG

Im folgenden Beispiel ändert die Anweisung T_CONFIG im Unterfeld "ntp" (Network-Time-Protocol-Server (NTP)) die IP-Adressen von bis zu vier NTP-Servern.

Auf der Seite "Uhrzeitsynchronisation" in den "Eigenschaften" der CPU, PROFINET-Schnittstelle [X1], konfigurieren Sie die NTP-Synchronisation durch Aktivieren des Kontrollkästchens "Uhrzeitsynchronisation über NTP-Server aktivieren" (siehe folgende Abbildung). Dann können Sie die IP-Adressen der NTP-Server über die Anweisung T_CONFIG ändern, nachdem Sie das Programm in die CPU geladen haben.

CONF_DATA_3			
	Name	Datentyp	Startwert
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

PROFINET interface_1 [Module] Eigenschaften Info Diagnose

Allgemein | IO-Variablen | Systemkonstanten | Texte

Allgemein
 Ethernet-Adressen
Uhrzeitsynchronisation
 Betriebsart
 Erweiterte Optionen
 Zugriff auf den Webserver
 HWKennung

Uhrzeitsynchronisation

Uhrzeitsynchronisation über NTP-Server aktivieren

IP-Adressen

Server 1: 192 . 168 . 2 . 1
 Server 2: 192 . 168 . 2 . 2
 Server 3: 192 . 168 . 2 . 3
 Server 4: 192 . 168 . 2 . 4

Aktualisierungsintervall: 10 sec

Uhrzeitsynchronisation der CIMCP-Uhr mit den Uhren der CPU.

11.5.8.20 Gemeinsame Parameter für Anweisungen

Eingangsparameter REQ

Viele der Anweisungen für die offene Benutzerkommunikation haben einen Eingang REQ, der die Anweisung bei einer steigenden Flanke (0 nach 1) initiiert. Der Eingang REQ muss während der Ausführung einer Anweisung 1 (WAHR) sein, doch der Eingang REQ kann so lange wie gewünscht WAHR bleiben. Die Anweisung stößt erst dann eine andere Anweisung an, wenn sie mit Eingang REQ = FALSCH ausgeführt wird, so dass die Anweisung den Zustand des Eingangs REQ zurücksetzen kann. Dies ist erforderlich, damit die Anweisung die steigende Flanke zum Starten der nächsten Anweisung erkennen kann.

Wenn Sie eine dieser Anweisungen in Ihr Programm einfügen, werden Sie von STEP 7 aufgefordert, den Instanz-DB anzugeben. Verwenden Sie für jeden Anweisungsaufruf einen eindeutigen DB. Dadurch wird sichergestellt, dass jede Anweisung die Eingänge wie REQ ordnungsgemäß verarbeitet.

Eingangsparameter ID

Dies ist ein Verweis auf die "Lokale ID (hex)" in der "Netzsicht" unter "Geräte & Netze" in STEP 7 und es ist die ID des Netzwerks, das Sie für diesen Kommunikationsbaustein verwenden möchten. Die ID muss mit der zugehörigen Parameter-ID in der lokalen Verbindungsbeschreibung übereinstimmen.

Ausgangsparameter DONE, NDR, ERROR und STATUS

Diese Anweisungen haben Ausgänge, die den Ausführungszustand anzeigen:

Tabelle 11-57 Ausgabeparameter der Anweisungen für die offene Benutzerkommunikation

Parameter	Datentyp	Standard	Beschreibung
DONE	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehler abgeschlossen wurde; andernfalls FALSCH.
NDR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die angeforderte Aktion fehlerfrei abgeschlossen und neue Daten empfangen wurden; andernfalls FALSCH.
BUSY	Bool	FALSCH	Wird, wenn aktiv, auf WAHR gesetzt, um Folgendes anzuzeigen: <ul style="list-style-type: none"> • Der Auftrag ist noch nicht beendet. • Es kann kein neuer Auftrag gestartet werden. Wird auf FALSCH gesetzt, wenn der Auftrag beendet ist.

Parameter	Datentyp	Standard	Beschreibung
ERROR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehlern abgeschlossen wurde, der entsprechende Fehlercode befindet sich in STATUS; andernfalls FALSCH.
STATUS	Word	0	<p>Ergebniszustand:</p> <ul style="list-style-type: none"> • Wird das Bit DONE oder NDR gesetzt, wird STATUS auf 0 oder auf einen Informationscode gesetzt. • Wird das Bit ERROR gesetzt, wird STATUS auf einen Fehlercode gesetzt. • Wird keines der oben aufgeführten Bits gesetzt, kann die Anweisung Statusergebnisse zurückgeben, die den aktuellen Zustand der Funktion beschreiben. <p>Der Wert in STATUS wird während der Ausführung der Funktion gehalten.</p>

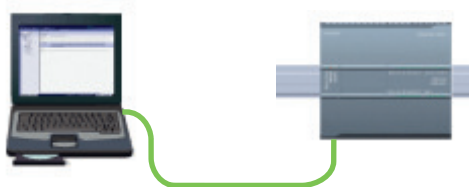
Hinweis

Beachten Sie, dass DONE, NDR und ERROR nur eine Ausführung lang gesetzt sind.

Eingeschränkte TSAPs und Portnummern für passive ISO- und TCP-Kommunikation

Wenn Sie mit der Operation TCON eine passive Kommunikationsverbindung einrichten und aufbauen, dürfen die folgenden Portadressen nicht verwendet werden:

- ISO TSAP (passiv):
 - 01.00, 01.01, 02.00, 02.01, 03.00, 03.01
 - 10.00, 10.01, 11.00, 11.01, ... BF.00, BF.01
- TCP-Port (passiv) und UDP-Port (passiv):
 - 25, 80, 102, 5001, 34962, 34963, 34964

11.5.9 Kommunikation mit einem Programmiergerät

Eine CPU kann mit einem Programmiergerät STEP 7 in einem Netzwerk kommunizieren.

Berücksichtigen Sie beim Einrichten der Kommunikation zwischen einer CPU und einem Programmiergerät Folgendes:

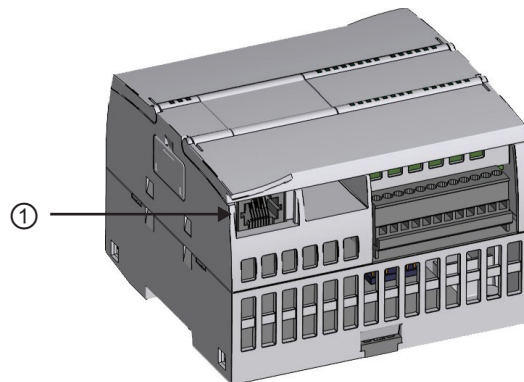
- Konfiguration/Setup: Eine Hardwarekonfiguration ist erforderlich.
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

11.5.9.1 Hardware-Kommunikationsverbindung herstellen

Die PROFINET-Schnittstelle stellt die physische Verbindung zwischen einem Programmiergerät und einer CPU her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die direkte Verbindung des Programmiergeräts mit der CPU nicht benötigt.

Zum Herstellen der Hardwareverbindung zwischen einem Programmiergerät und einer CPU gehen Sie folgendermaßen vor:

1. Installieren Sie die CPU (Seite 49).
2. Schließen Sie das Ethernet-Kabel wie unten gezeigt am PROFINET-Anschluss an.
3. Schließen Sie das Ethernet-Kabel an das Programmiergerät an.



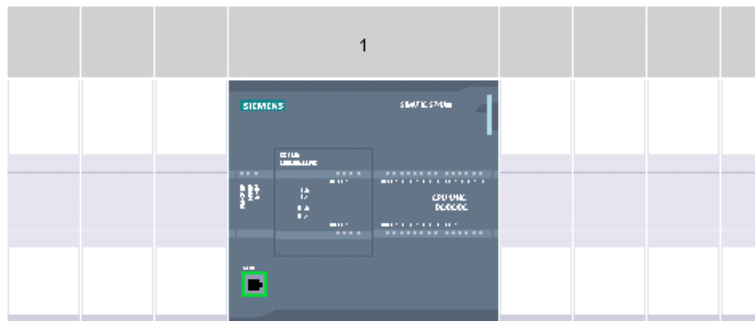
① PROFINET-Anschluss

Für die PROFINET-Verbindung ist eine optionale Zugentlastung verfügbar. Bestellinformationen finden Sie unter Ersatzteile und sonstige Hardware (Seite 1446).

11.5.9.2 Konfigurieren der Geräte

Wenn Sie bereits ein Projekt mit einer CPU angelegt haben, öffnen Sie Ihr Projekt in STEP 7.

Sonst legen Sie ein Projekt an und fügen eine CPU (Seite 136) in den Baugruppenträger ein. Im folgenden Projekt sehen Sie eine CPU in der "Gerätesicht".



11.5.9.3 IP-Adressen zuweisen

IP-Adressen zuweisen

In einem PROFINET-Netzwerk benötigt jedes Gerät außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern:

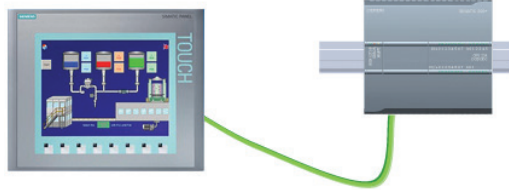
- Wenn Sie Programmier- oder andere Netzwerkgeräte nutzen, die eine integrierte Adapterkarte mit Anschluss an Ihr Anlagen-LAN oder eine Ethernet-USB-Adapterkarte mit Anschluss an ein getrenntes Netzwerk verwenden, müssen diesen Geräten IP-Adressen zugewiesen werden. Weitere Informationen finden Sie unter "IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen" (Seite 603).
- Sie können einer CPU oder einem Netzwerkgerät auch online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich. Weitere Informationen finden Sie unter "IP-Adresse zu einer CPU online zuweisen (Seite 603)".
- Nachdem Sie Ihre CPU oder Ihr Netzwerkgerät in Ihrem Projekt konfiguriert haben, können Sie die Parameter für die PROFINET-Schnittstelle konfigurieren, um die IP-Adresse aufzunehmen. Weitere Informationen finden Sie unter "IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 606).

11.5.9.4 Testen Ihres PROFINET-Netzwerks

Nach Abschluss der Konfiguration müssen Sie Ihr Projekt in die CPU laden. Alle IP-Adressen werden beim Laden des Projekts konfiguriert.

Die Funktion "Laden in Gerät" der CPU und der zugehörigen Dialog "Erweitertes Laden" kann alle erreichbaren Netzwerkgeräte anzeigen und zusätzlich angeben, ob allen Geräten eindeutige IP-Adressen zugeordnet wurden. Ausführliche Informationen finden Sie unter "Testen des PROFINET-Netzwerks" (Seite 612).

11.5.10 Kommunikation HMI/PLC



Die CPU unterstützt PROFINET-Kommunikationsverbindungen mit HMIs (Seite 32). Zum Einrichten der Kommunikation zwischen CPUs und HMIs müssen die folgenden Bedingungen erfüllt sein:

Konfiguration/Setup:

- Der PROFINET-Port der CPU muss für die Verbindung mit der HMI konfiguriert sein.
- Die HMI muss eingerichtet und konfiguriert sein.
- Die HMI-Konfigurationsdaten sind Teil des CPU-Projekts und können im Projekt konfiguriert und geladen werden.
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

Hinweis

Der auf dem Baugruppenträger montierte Ethernet-Switch CSM1277 mit 4 Ports dient zum Anschließen Ihrer CPUs und HMI-Geräte. Der PROFINET-Anschluss an der CPU enthält keinen Ethernet-Switch.

Unterstützte Funktionen:

- Die HMI kann Daten der CPU lesen/schreiben.
- Meldungen können anhand von Informationen der CPU ausgelöst werden.
- Systemdiagnose

Tabelle 11-58 Erforderliche Schritte zum Konfigurieren der Kommunikation zwischen einer HMI und einer CPU

Schritt	Aufgabenstellung
1	Hardware-Kommunikationsverbindung herstellen Eine PROFINET-Schnittstelle stellt die physische Verbindung zwischen einer HMI und einer CPU her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die direkte Verbindung der HMI mit der CPU nicht benötigt. Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Hardware-Kommunikationsverbindung herstellen (Seite 729)".
2	Geräte konfigurieren Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Geräte konfigurieren (Seite 729)".
3	Konfigurieren der logischen Netzwerkverbindungen zwischen einer HMI und einer CPU Weitere Informationen finden Sie unter "Kommunikation zwischen HMI und PLC: Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten" (Seite 732).

Schritt	Aufgabenstellung
4	IP-Adresse in Ihrem Projekt konfigurieren Gehen Sie auf dieselbe Weise vor. Sie müssen jedoch die IP-Adressen für die HMI und die CPU konfigurieren. Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 608).
5	Testen des PROFINET-Netzwerks Sie müssen die Konfiguration für jede CPU und jedes HMI-Gerät laden. Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Testen des PROFINET-Netzwerks" (Seite 612).

11.5.10.1 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten

Nachdem Sie den Baugruppenträger mit der CPU konfiguriert haben, können Sie Ihre Netzwerkverbindungen einrichten.

Im Portal "Geräte & Netze" können Sie in der "Netzsicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine ISO-on-TCP-Verbindung).

Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 600).

11.5.11 Kommunikation PLC/PLC



Eine CPU kann mit den Anweisungen TSEND_C und TRCV_C mit einer anderen CPU im Netzwerk kommunizieren.

Berücksichtigen Sie beim Einrichten der Kommunikation zwischen zwei CPUs Folgendes:

- Konfiguration/Setup: Eine Hardwarekonfiguration ist erforderlich.
- Unterstützte Funktionen: Lesen/Schreiben von Daten in eine vernetzte CPU
- Für die Kommunikation zwischen zwei Teilnehmern ist kein Ethernet-Switch erforderlich; erst wenn mehr als zwei Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

Tabelle 11-59 Erforderliche Schritte zum Konfigurieren der Kommunikation zwischen zwei CPUs

Schritt	Aufgabenstellung
1	<p>Hardware-Kommunikationsverbindung herstellen</p> <p>Eine PROFINET-Schnittstelle stellt die physische Verbindung zwischen zwei CPUs her. Da in der CPU die Auto-Cross-Over-Funktionalität implementiert ist, kann für die Schnittstelle ein Standard- oder Crossover-Ethernet-Kabel verwendet werden. Ein Ethernet-Switch wird für die Verbindung der zwei CPUs nicht benötigt.</p> <p>Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Hardware-Kommunikationsverbindung herstellen (Seite 729)".</p>
2	<p>Geräte konfigurieren</p> <p>Sie müssen zwei CPUs in Ihrem Projekt konfigurieren.</p> <p>Weitere Informationen finden Sie unter "Kommunikation mit einem Programmiergerät: Geräte konfigurieren (Seite 729)".</p>
3	<p>Konfigurieren der logischen Netzwerkverbindungen zwischen zwei CPUs</p> <p>Weitere Informationen finden Sie unter "Kommunikation zwischen PLC und PLC: Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten" (Seite 733).</p>
4	<p>IP-Adresse in Ihrem Projekt konfigurieren</p> <p>Gehen Sie auf dieselbe Weise vor. Sie müssen jedoch die IP-Adressen für zwei CPUs konfigurieren (z.B. PLC_1 und PLC_2).</p> <p>Weitere Informationen finden Sie unter "Gerätekonfiguration: IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 608).</p>
5	<p>Sende- und Empfangsparameter konfigurieren</p> <p>Sie müssen die Anweisungen TSEND_C und TRCV_C in beiden CPUs konfigurieren, um die Kommunikation zwischen ihnen zu ermöglichen.</p> <p>Weitere Informationen finden Sie unter "Einrichten der Kommunikation zwischen zwei CPUs: Send- und Empfangsparameter konfigurieren (Seite 734)".</p>
6	<p>Testen des PROFINET-Netzwerks</p> <p>Sie müssen die Konfiguration für jede CPU laden.</p> <p>Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Testen des PROFINET-Netzwerks" (Seite 612).</p>

11.5.11.1 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei Geräten

Nachdem Sie den Baugruppenträger mit der CPU konfiguriert haben, können Sie Ihre Netzwerkverbindungen einrichten.

Im Portal "Geräte & Netze" können Sie in der "Netzsicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine ISO-on-TCP-Verbindung).

Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 600).

11.5.11.2 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

Allgemeine Parameter konfigurieren

Sie legen die Kommunikationsparameter im Dialog "Eigenschaften" der Kommunikationsanweisung fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

Die zu verwendenden TSAPs oder Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adresdetails". Der TSAP oder Port einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP oder Port für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

11.5.11.3 Sende- und Empfangsparameter konfigurieren

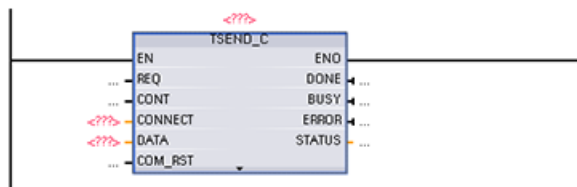
Über Kommunikationsbausteine (z. B. TSEND_C und TRCV_C) werden Verbindungen zwischen zwei CPUs hergestellt. Damit die PROFINET-Kommunikation von den CPUs aufgenommen werden kann, müssen die Parameter zum Senden und Empfangen von Meldungen eingerichtet werden. Diese Parameter legen fest, wie die Kommunikation abläuft, wenn Meldungen von einem Zielgerät gesendet oder von diesem empfangen werden.

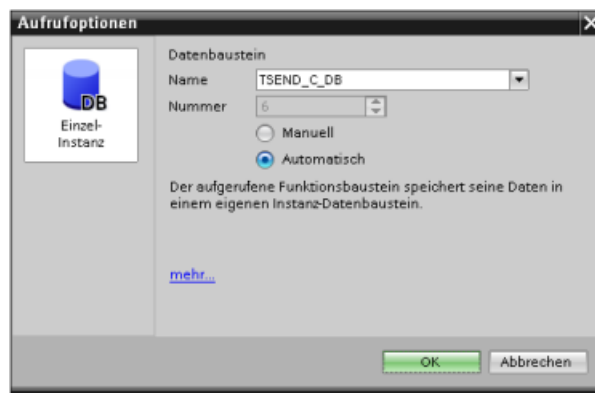
Sendeparameter für TSEND_C konfigurieren

Anweisung TSEND_C

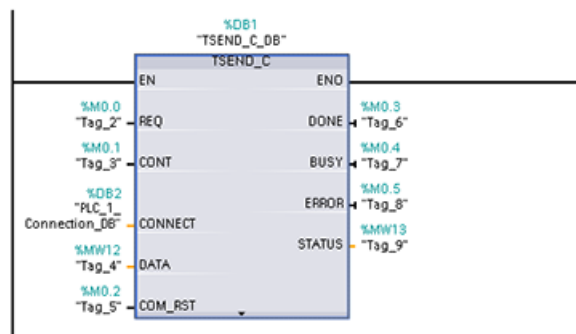
Die Anweisung TSEND_C (Seite 636) baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht, bis eine Anweisung die Verbindung trennt. Die Anweisung TSEND_C verbindet die Funktionen von TCON, TDISCON und TSEND.

In STEP 7 können Sie in der Gerätekonfiguration einstellen, wie Daten mit einer Anweisung TSEND_C gesendet werden sollen. Zuerst fügen Sie die Anweisung aus dem Ordner "Kommunikation" in der Taskcard "Anweisungen" in das Programm ein. Die Anweisung TSEND_C wird zusammen mit dem Dialog "Aufrufoptionen" angezeigt, in dem Sie einen DB zum Speichern der Parameter der Anweisung zuweisen können.





Sie können den Ein- und Ausgängen Speicherplätze im Variablenspeicher zuweisen. Dies wird in der folgenden Abbildung gezeigt:



Allgemeine Parameter konfigurieren

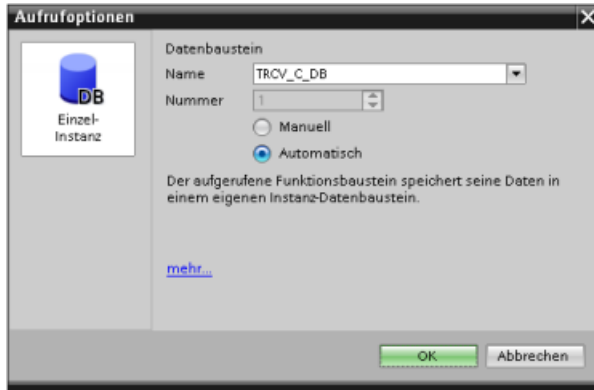
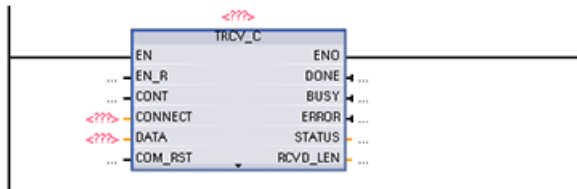
Sie können die Parameter im Dialog "Eigenschaften" der Anweisung TSEND_C festlegen. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung TSEND_C selektiert haben.

Empfangsparameter für TRCV_C konfigurieren

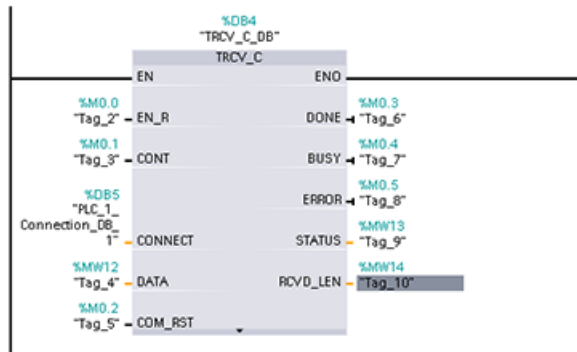
Anweisung TRCV_C

Die Anweisung TRCV_C (Seite 636) baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht, bis eine Anweisung die Verbindung trennt. Die Anweisung TRCV_C verbindet die Funktionen von TCON, TDISCON und TRCV.

In STEP 7 können Sie in der CPU-Konfiguration einstellen, wie eine Anweisung TRCV_C Daten empfangen soll. Zuerst fügen Sie die Anweisung aus dem Ordner "Kommunikation" in der Taskcard "Anweisungen" in das Programm ein. Die Anweisung TRCV_C wird zusammen mit dem Dialog "Aufrufoptionen" angezeigt, in dem Sie einen DB zum Speichern der Parameter der Anweisung zuweisen können.



Sie können den Ein- und Ausgängen Speicherplätze im Variablenspeicher zuweisen. Dies wird in der folgenden Abbildung gezeigt:



Allgemeine Parameter konfigurieren

Sie können die Parameter im Dialog "Eigenschaften" der Anweisung TRCV_C festlegen. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung TRCV_C selektiert haben.

11.5.12 CPU und PROFINET IO-Gerät konfigurieren

11.5.12.1 PROFINET IO-Gerät hinzufügen

PROFINET IO-Device hinzufügen



Verwenden Sie im Portal "Geräte & Netze" den Hardwarekatalog, um PROFINET IO-Devices hinzuzufügen.

Hinweis

Um ein PROFINET IO-Device hinzuzufügen, können Sie STEP 7 Professional oder Basic ab V11 verwenden.

Zum Einfügen eines ET200SP IO-Device erweitern Sie zum Beispiel die folgenden Behälter im Hardwarekatalog: Dezentrale Peripherie, ET 200SP, Schnittstellenmodule und PROFINET. Sie können dann das Schnittstellenmodul in der Liste der ET200SP-Geräte (die nach der Bestellnummer sortiert sind) auswählen und das ET200SP IO-Device hinzufügen.

Tabelle 11-60 ET200SP IO-Device zur Gerätekonfiguration hinzufügen

IO-Device einfügen	Ergebnis
	

Sie können nun das PROFINET IO-Device mit der CPU verbinden:

1. Klicken Sie auf dem Gerät mit der rechten Maustaste auf den Link "Nicht zugewiesen" und wählen Sie im Kontextmenü den Befehl "Neue IO-Steuerung zuweisen" aus, um den Dialog "IO-Steuerung auswählen" aufzurufen.
2. Wählen Sie in der Liste der IO-Steuerungen im Projekt Ihre S7-1200 CPU aus (in diesem Beispiel "PLC_1").
3. Klicken Sie auf "OK", um die Netzwerkverbindung anzulegen.

Sie können auch ins Portal "Geräte & Netze" gehen und dort in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen:

1. Um eine PROFINET-Verbindung zu erstellen, klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen eine Linie zum PROFINET-Feld auf dem zweiten Gerät.
2. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Weitere Informationen finden Sie unter "Gerätekonfiguration: CPU für die Kommunikation konfigurieren" (Seite 171).

11.5.12.2 CPUs und Gerätenamen zuweisen

CPUs und Gerätenamen zuweisen

Netzwerkverbindungen zwischen den Geräten weisen auch das PROFINET IO-Device der CPU zu, was erforderlich ist, damit die CPU das Gerät steuern kann. Um diese Zuweisung zu ändern, klicken Sie auf den PLC-Namen, der auf dem PROFINET IO-Device angezeigt wird. Daraufhin wird ein Dialog geöffnet, in dem Sie das PROFINET IO-Device von der aktuellen CPU trennen und neu zuweisen oder auch ohne Zuweisung lassen können.

Den Geräten in Ihrem PROFINET-Netzwerk muss ein Name zugewiesen sein, damit Sie eine Verbindung zur CPU herstellen können. Sie weisen Ihren PROFINET-Geräten in der "Netzsicht" Namen zu, wenn die Geräte noch keinen Namen haben oder wenn der Name eines Geräts geändert werden soll. Klicken Sie hierfür mit der rechten Maustaste auf das PROFINET IO-Device und wählen Sie "Gerätename zuweisen".

Den Namen eines PROFINET IO-Device müssen Sie sowohl im STEP 7-Projekt als auch dem PROFINET IO-Device im PROFINET-Netzwerk zuweisen. (Um den Gerätenamen im PROFINET-Netzwerk zuzuweisen, können Sie entweder das STEP 7-Tool "Online und Diagnose" oder das PRONETA-Tool für Inbetriebnahme, Konfiguration und Diagnose verwenden.) Fehlt ein Name oder entsprechen sich die Namen an den beiden Speicherorten nicht, funktioniert der Modus für den PROFINET IO-Datenaustausch nicht. Weitere Informationen finden Sie unter "Online- und Diagnose-Tools: einem PROFINET-Gerät online einen Namen zuweisen (Seite 1210)".

11.5.12.3 IP-Adressen zuweisen

IP-Adressen zuweisen

In einem PROFINET-Netzwerk benötigt jedes Gerät außerdem eine IP-Adresse (Internet-Protocol-Adresse). Mit Hilfe dieser Adresse kann ein Gerät Daten über ein komplexeres Netzwerk liefern:

- Wenn Sie Programmier- oder andere Netzwerkgeräte nutzen, die eine integrierte Adapterkarte mit Anschluss an Ihr Anlagen-LAN oder eine Ethernet-USB-Adapterkarte mit Anschluss an ein getrenntes Netzwerk verwenden, müssen diesen Geräten IP-Adressen zugewiesen werden. Weitere Informationen finden Sie unter "IP-Adressen zu Programmier- und Netzwerkgeräten zuweisen" (Seite 603).
- Sie können einer CPU oder einem Netzwerkgerät auch online eine IP-Adresse zuweisen. Dies ist besonders bei der ersten Gerätekonfiguration nützlich. Weitere Informationen finden Sie unter "IP-Adresse zu einer CPU online zuweisen (Seite 606)".
- Nachdem Sie Ihre CPU oder Ihr Netzwerkgerät in Ihrem Projekt konfiguriert haben, können Sie die Parameter für die PROFINET-Schnittstelle konfigurieren, um die IP-Adresse aufzunehmen. Weitere Informationen finden Sie unter "IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 608).

11.5.12.4 IO-Zykluszeit konfigurieren

IO-Zykluszeit konfigurieren

Ein PROFINET IO-Device wird innerhalb des Zeitraums "IO-Zyklus" von der CPU mit neuen Daten versorgt. Die Aktualisierungszeit kann für jedes Gerät separat konfiguriert werden und legt den Zeitraum fest, in dem Daten von der CPU zum und vom Gerät übertragen werden.

STEP 7 berechnet die Aktualisierungszeit für den "IO-Zyklus" automatisch in den Standardeinstellungen für jedes Gerät des PROFINET-Netzwerks und berücksichtigt dabei das Volumen der auszutauschenden Daten und die Anzahl der dieser Steuerung zugewiesenen Geräte. Wenn Sie nicht möchten, dass die Aktualisierungszeit automatisch berechnet wird, können Sie diese Einstellung ändern.

Sie legen die Parameter für den "IO-Zyklus" im Konfigurationsdialog "Eigenschaften" des PROFINET IO-Device fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

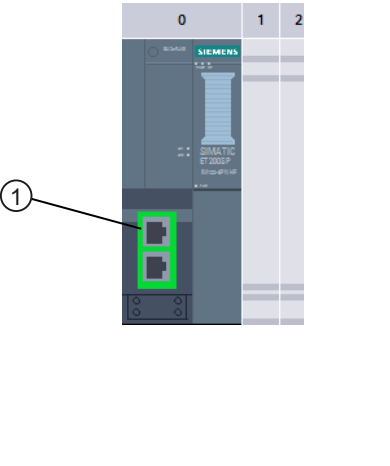
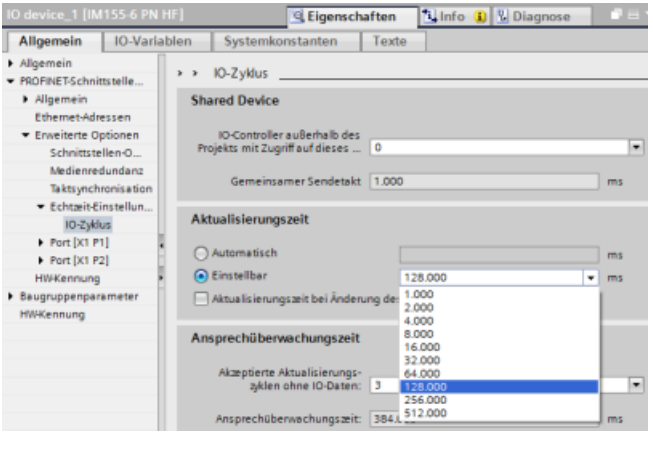
Klicken Sie in der "Gerätesicht" des PROFINET IO-Device auf den PROFINET-Port. Im Dialog "PROFINET-Schnittstelle" rufen Sie die Parameter "IO-Zyklus" mit der folgenden Menüauswahl auf:

- "Erweiterte Optionen"
- "Echtzeiteinstellungen"
- "IO-Zyklus"

Definieren Sie die "Aktualisierungszeit" für den IO-Zyklus wie folgt:

- Wenn eine geeignete Aktualisierungszeit automatisch berechnet werden soll, wählen Sie "Automatisch".
- Um die Aktualisierungszeit selbst einzustellen, wählen Sie "Kann eingestellt werden" und geben die gewünschte Aktualisierungszeit in ms ein.

Tabelle 11-61 PROFINET IO-Zykluszeit für das ET200SP konfigurieren

ET 200SP PROFINET IO-Device	Dialogfeld "ET 200SP PROFINET IO-Zyklus"
	

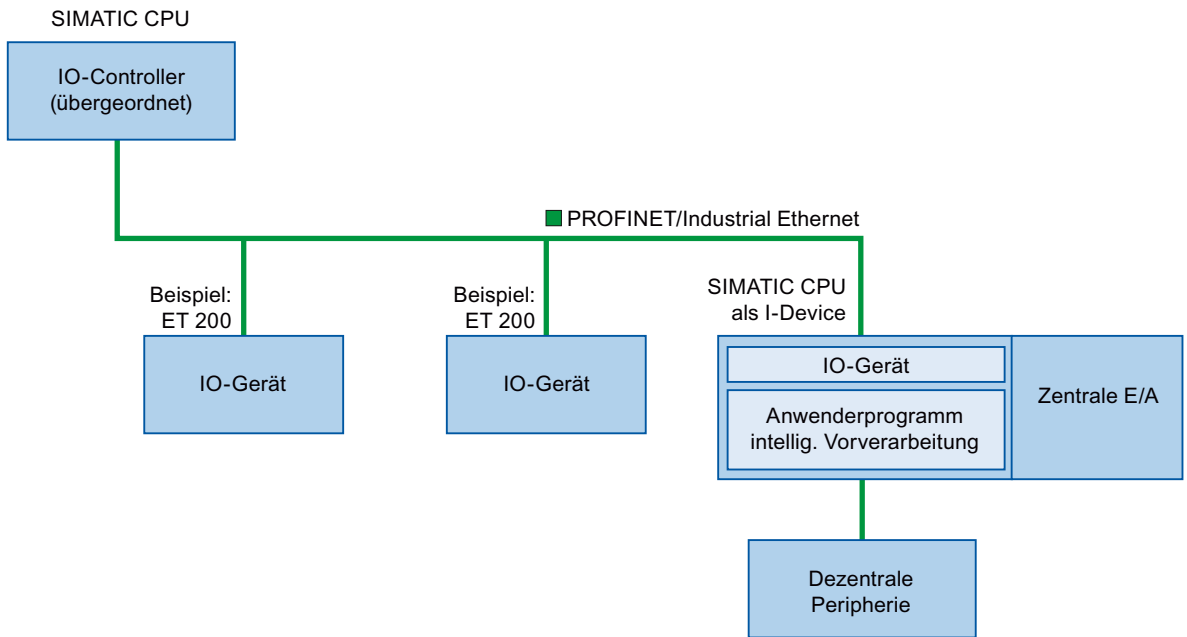
① PROFINET-Port

11.5.13 CPU und PROFINET I-Device konfigurieren

11.5.13.1 I-Device-Funktionalität

Die Funktion "I-Device" (intelligentes IO-Gerät) der CPU erleichtert den Datenaustausch mit einem IO-Controller und den Betrieb der CPU z. B. als intelligente Vorverarbeitungseinheit für Teilprozesse. Das I-Device ist als IO-Gerät mit einem übergeordneten IO-Controller verbunden.

Die Vorverarbeitung erfolgt durch das Anwenderprogramm der CPU. Die in den zentralen oder dezentralen E/A (PROFINET IO oder PROFIBUS DP) erfassten Prozesswerte werden vom Anwenderprogramm vorverarbeitet und der CPU einer übergeordneten Station über eine PROFINET IO-Schnittstelle zur Verfügung gestellt.



Benennungskonvention für "I-Device"

Eine CPU oder ein CP mit I-Device-Funktion wird in den folgenden Beschreibungen der Einfachheit halber als "I-Device" bezeichnet.

11.5.13.2 Eigenschaften und Vorteile des I-Device

Einsatzbereiche

Einsatzbereiche für ein I-Device:

- **Dezentrale Verarbeitung:**
Komplexe Automatisierungsaufgaben können in kleinere Teilanlagen/Teilprozesse aufgeteilt werden. Die Prozesse werden somit handlicher und die Teilaufgaben vereinfacht.
- **Abtrennen von Teilprozessen:**
Komplexe, weit verteilte und umfangreiche Prozesse können mit I-Devices in mehrere Teilprozesse mit schlankeren Schnittstellen aufgeteilt werden. Die Teilprozesse können bei Bedarf in einzelnen STEP 7-Projekten gespeichert werden, die später in einem Hauptprojekt zusammengeführt werden können.
- **Know-how-Schutz:**
Komponenten können mit lediglich einer GSD-Datei für die I-Device-Schnittstellenbeschreibung an Stelle eines STEP 7-Projekts geliefert werden. Der Anwender kann sein Programm schützen, weil es nicht mehr veröffentlicht werden muss.

Eigenschaften

Eigenschaften des I-Device:

- **Getrenntes Arbeiten mit STEP 7-Projekten:**
Ersteller und Benutzer eines I-Device können mit vollständig getrennten STEP 7-Automatisierungsprojekten arbeiten. Die GSD-Datei bildet die Schnittstelle zwischen den STEP 7-Projekten. Auf diese Weise kann durch eine standardisierte Schnittstelle eine Verbindung zu Standard-IO-Controllern hergestellt werden.
- **Real-Time-Kommunikation:**
Das I-Device wird über eine PROFINET IO-Schnittstelle mit einem deterministischen PROFINET IO-System geliefert.

Vorteile

Ein I-Device hat die folgenden Vorteile:

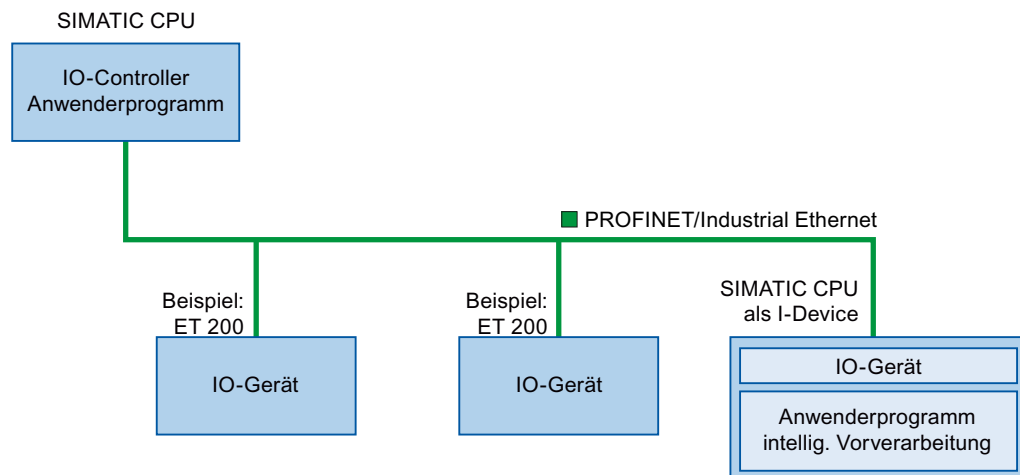
- Einfache Anbindung von IO-Controllern
- Echtzeit-Kommunikation zwischen IO-Controllern
- Entlastung des IO-Controllers durch Verteilung der Rechenleistung auf die I-Devices
- Geringere Kommunikationslast durch lokale Verarbeitung der Prozessdaten
- Handlichere Projekte durch die Verarbeitung von Teilaufgaben in separaten STEP 7-Projekten

11.5.13.3 Merkmale eines I-Device

Ein I-Device wird in einem E/A-System wie ein Standard-IO-Gerät eingefügt.

I-Device ohne untergeordnetes PROFINET IO-System

Ein I-Device hat keine eigene dezentrale Peripherie. Die Konfiguration und Parametrierung eines I-Device als IO-Gerät sind die gleichen wie bei einem dezentralen E/A-System (zum Beispiel ET 200).



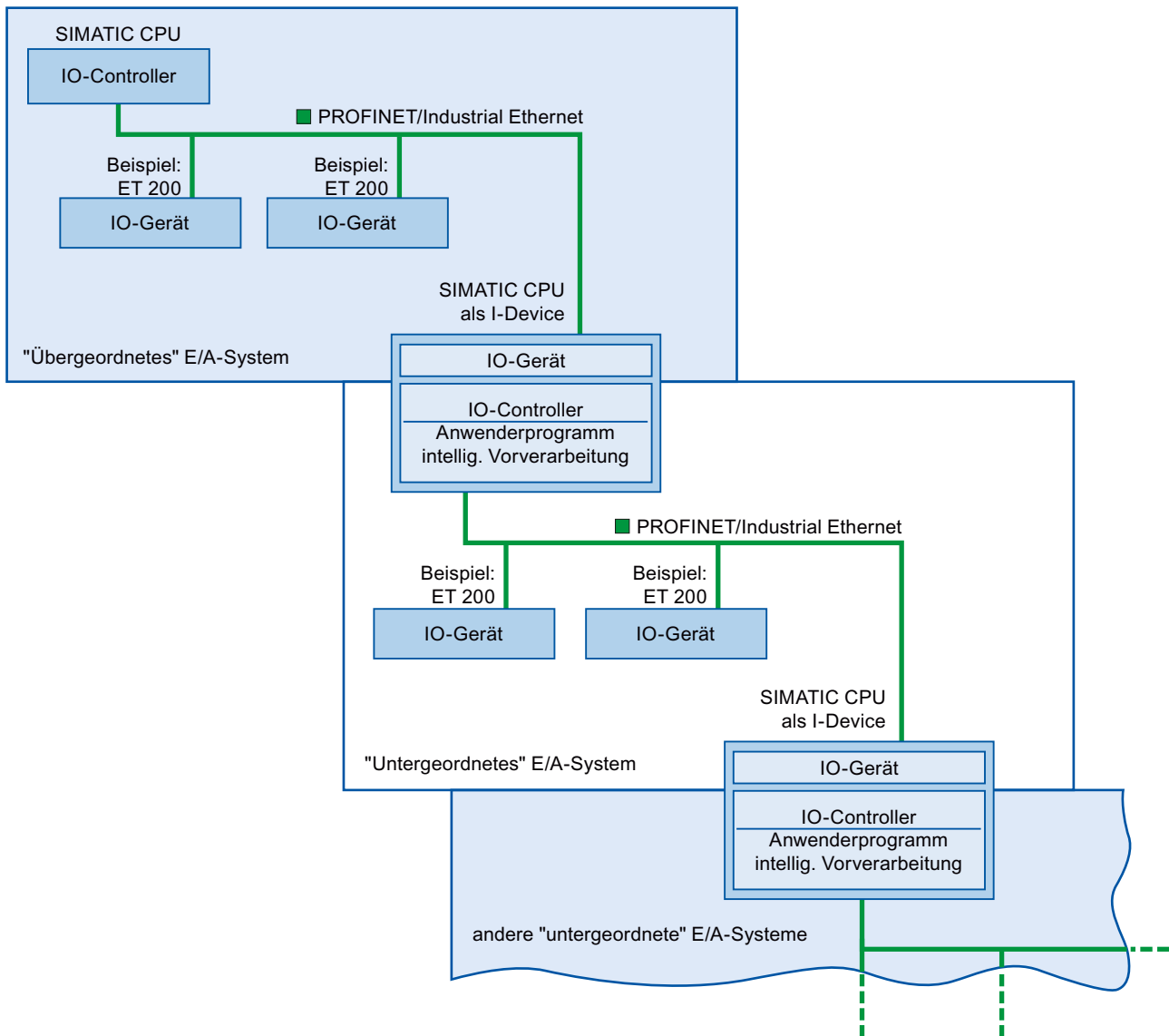
I-Device mit untergeordnetem PROFINET IO-System

Je nach Konfiguration kann ein I-Device zusätzlich zu seiner Funktion als IO-Gerät auch als IO-Controller an einer PROFINET-Schnittstelle genutzt werden.

Somit kann ein I-Device über seine PROFINET-Schnittstelle zu einem übergeordneten E/A-System gehören und als IO-Controller sein eigenes untergeordnetes E/A-System unterstützen.

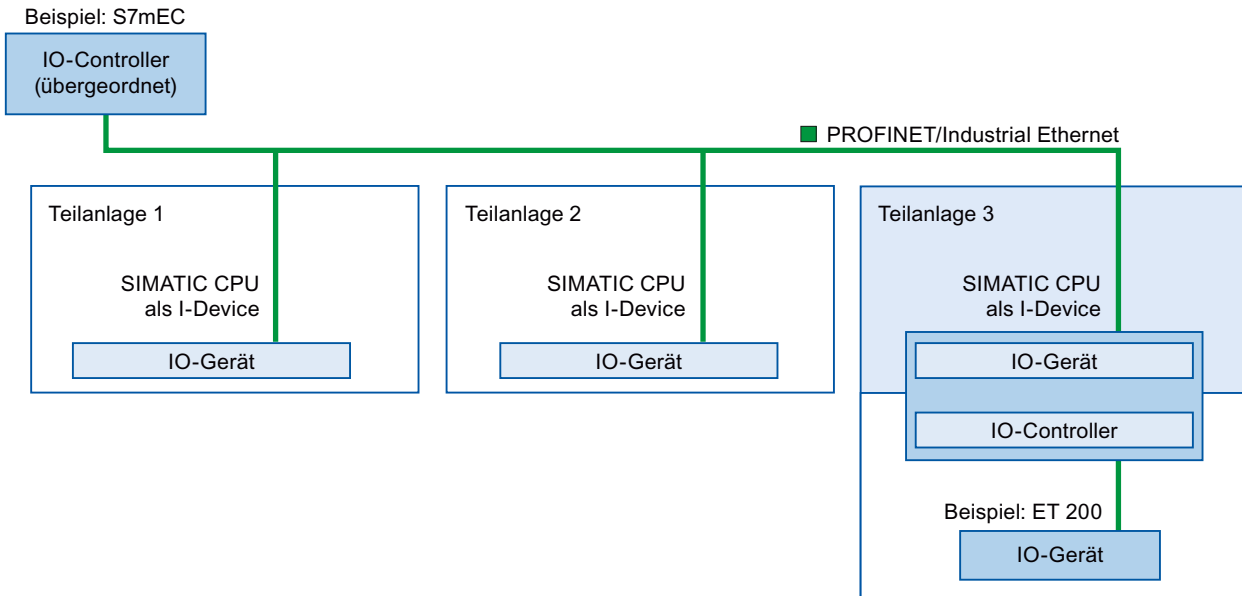
Das untergeordnete E/A-System kann wiederum eigene I-Devices enthalten (siehe Abbildung unten). Dies ermöglicht hierarchisch aufgebaute E/A-Systeme.

Zusätzlich zu seiner Funktion als IO-Controller kann ein I-Device auch über eine PROFIBUS-Schnittstelle als DP-Master für ein untergeordnetes PROFIBUS-System fungieren.



Beispiel: I-Device als IO-Gerät und IO-Controller

Der Funktionsablauf eines I-Device als IO-Gerät und IO-Controller wird hier am Beispiel eines Druckprozesses erklärt. Das I-Device steuert eine Teilanlage (einen Teilprozess). Eine von mehreren Teilanlagen wird beispielsweise verwendet, um zusätzliche Blätter wie Flyer oder Broschüren in ein Paket mit Druckerzeugnissen einzufügen.



Teilanlage 1 und Teilanlage 2 bestehen aus je einem I-Device mit zentralen E/A. Zusammen mit dem dezentralen E/A-System (zum Beispiel ET 200) bildet das I-Device die Teilanlage 3.

Das Anwenderprogramm im I-Device ist für die Vorverarbeitung der Prozessdaten verantwortlich. Für diese Aufgabe benötigt das Anwenderprogramm des I-Device Standardeinstellungen (zum Beispiel Steuerungsdaten) vom übergeordneten IO-Controller. Das I-Device liefert die Ergebnisse (zum Beispiel Status seiner Teilaufgabe) an den übergeordneten IO-Controller.

11.5.13.4 Datenaustausch zwischen über- und untergeordnetem E/A-System

Transferbereiche sind Schnittstellen zum Anwenderprogramm der I-Device-CPU. Die Eingänge werden im Anwenderprogramm verarbeitet, die Ausgänge sind das Ergebnis der Verarbeitung.

Die Daten für die Kommunikation zwischen IO-Controller und I-Device stehen in den Transferbereichen zur Verfügung. Ein Transferbereich enthält eine Informationseinheit, die konsistent zwischen IO-Controller und I-Device ausgetauscht wird. Weitere Informationen über die Konfiguration und Verwendung von Transferbereichen enthält das Kapitel "I-Device konfigurieren" (Seite 747).

Unterschiedliches Verhalten der Transferbereiche der Eingänge bei Steuerung und I-Device im Fall des Netzwerkverlusts

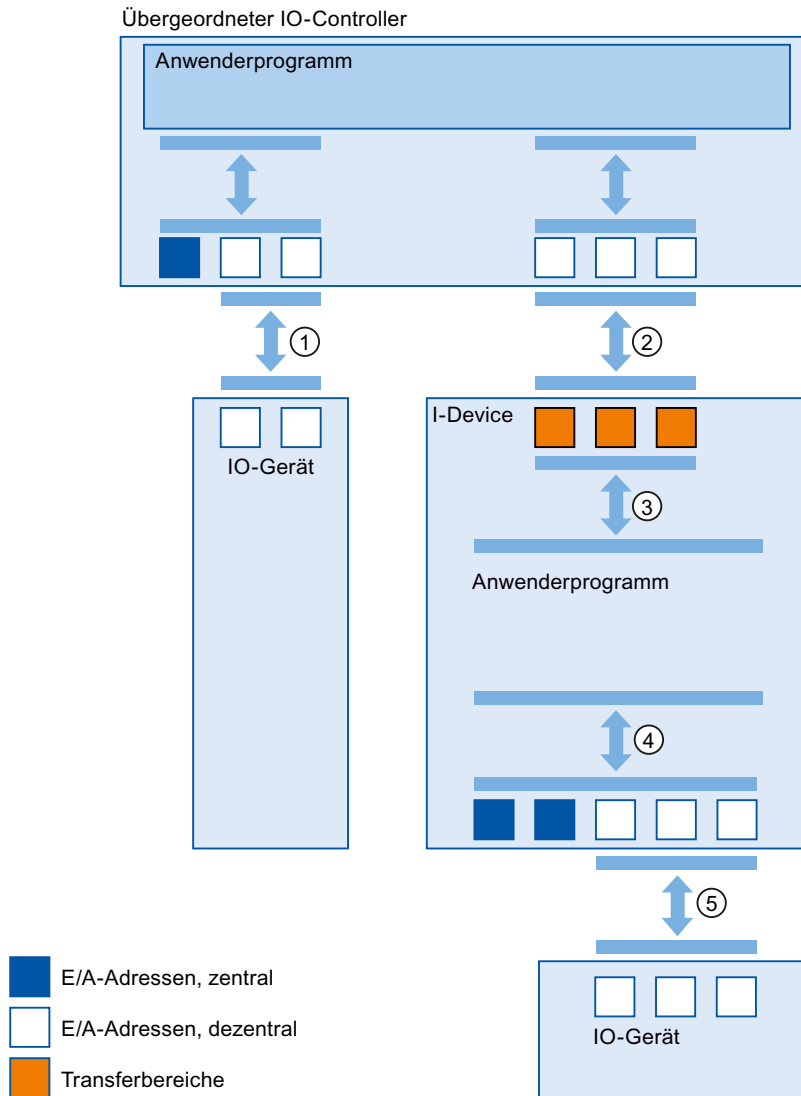
Bei der Steuerung schreibt die CPU bei Verlust der Netzwerkverbindung in die Transferbereiche der Eingänge den Wert Null. Auf dem I-Device behalten die Transferbereiche ihre letzten Werte bei.

Sie können Ihr System entsprechend konfigurieren, um diese Bedingung für den allgemeinen Fall des I-Device (nicht freigegebenes I-Device) zu vermeiden. Hierfür löschen Sie die Transferbereiche der Eingänge für das I-Device in einem OB für Baugruppenträger- oder Stationsausfall für ein kommendes Ereignis. Gehen Sie wie folgt vor:

1. Fügen Sie einen OB für Baugruppenträger- oder Stationsausfall in Ihr Projekt ein. (Dieser OB hat standardmäßig die Nummer OB 86.)
2. Fügen Sie dem OB Logik hinzu, um für die Werte der Eingänge des I-Device den Wert null zu schreiben, wenn die Anlaufvariable von KOP den Wert der Hardware-ID des I-Device anzeigt und die Anlaufvariable von Event_Class auf ein "kommendes" Ereignis hinweist:
 - Die Hardware-ID des I-Device finden Sie in der Standardvariablen­tabelle in der Registerkarte "Systemkonstanten". Die Hardware-ID ist eine Art von "HW_Device" und der Name der Variable zeigt an, dass es sich um ein I-Device handelt (zum Beispiel "Local~PROFINET_interface_1~IODevice").
 - Der Wert "16#39" in der Event_Class weist auf ein "kommendes" Ereignis hin. Wenn die Eingangsvariable "Event_Class" den Wert "16#39" enthält, weist dies darauf hin, dass der OB für Baugruppenträger- oder Stationsausfall jetzt aktiv ist (und nicht gelöscht ist).

Datenaustausch

Die nächste Abbildung zeigt den Datenaustausch zwischen über- und untergeordnetem E/A-System. Die einzelnen Kommunikationsbeziehungen sind nachstehend anhand von Nummern erläutert:



① **Datenaustausch zwischen übergeordnetem IO-Controller und normalem IO-Gerät**

Der IO-Controller und die IO-Geräte tauschen Daten überPROFINET aus.

② **Datenaustausch zwischen übergeordnetem IO-Controller und I-Device**

Der IO-Controller und das I-Device tauschen Daten über PROFINET aus.

Der Datenaustausch zwischen einem übergeordneten IO-Controller und einem I-Device beruht auf der üblichen Beziehung zwischen IO-Controller und IO-Gerät.

Für den übergeordneten IO-Controller stellen die Transferbereiche der I-Devices Teilmodule einer vorkonfigurierten Station dar.

Die Ausgangsdaten des IO-Controllers sind die Eingangsdaten des I-Device. Umgekehrt sind die Eingangsdaten des IO-Controllers die Ausgangsdaten des I-Device.

- ③ **Transferbeziehung zwischen Anwenderprogramm und Transferbereich**
Es werden Ein- und Ausgangsdaten zwischen Anwenderprogramm und Transferbereich ausgetauscht.
- ④ **Datenaustausch zwischen Anwenderprogramm und E/A des I-Device**
Es werden Ein- und Ausgangsdaten zwischen Anwenderprogramm und der zentralen/dezentralen Peripherie ausgetauscht.
- ⑤ **Datenaustausch zwischen I-Device und einem untergeordneten IO-Gerät**
Es werden Daten zwischen I-Device und seinen IO-Geräten ausgetauscht. Die Datenübertragung erfolgt überPROFINET.

11.5.13.5 I-Device konfigurieren

Für die Konfiguration gibt es grundsätzlich zwei Möglichkeiten:

- Konfiguration eines I-Device in einem Projekt
- Konfiguration eines I-Device, das in einem anderen Projekt oder Engineering-System verwendet wird.

In STEP 7 können Sie ein I-Device für ein anderes Projekt oder ein anderes Engineering-System konfigurieren, indem Sie ein konfiguriertes I-Device in eine GSD-Datei exportieren. Dann importieren Sie die GSD-Datei wie andere GSD-Dateien in andere Projekte oder Engineering-Systeme. In dieser GSD-Datei sind neben anderen Daten auch die Transferbereiche für den Datenaustausch gespeichert.

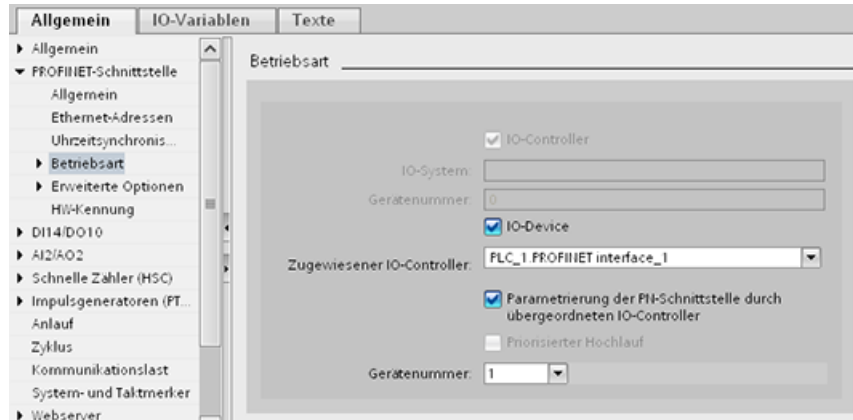
Hinweis

Wenn Sie die S7-1200 als Shared I-Device und als Controller verwenden, achten Sie darauf, dass Sie die Aktualisierungszeiten von PROFINET I-Device und PROFINET IO erhöhen, um die Auswirkungen auf die Kommunikationsleistung abzumildern. Das System ist sehr stabil und funktioniert gut, wenn Sie für ein einzelnes PROFINET I-Device eine Aktualisierungszeit von 2 ms auswählen und wenn Sie für ein einzelnes PROFINET IO eine Aktualisierungszeit von 2 ms auswählen.

Konfiguration eines I-Device in einem Projekt

1. Ziehen Sie mit der Maus eine PROFINET-CPU aus dem Hardwarekatalog in die Netzsicht.
2. Ziehen Sie mit der Maus eine PROFINET-CPU, die auch als IO-Device konfiguriert werden kann, aus dem Hardwarekatalog in die Netzsicht. Dieses Gerät wird als I-Device konfiguriert (z. B. eine CPU 1215C).
3. Wählen Sie die PROFINET-Schnittstelle für das I-Device.
4. Wählen Sie im Inspektorfenster im Navigationsbereich "Betriebsart" aus und aktivieren Sie das Kontrollkästchen "IO-Device".

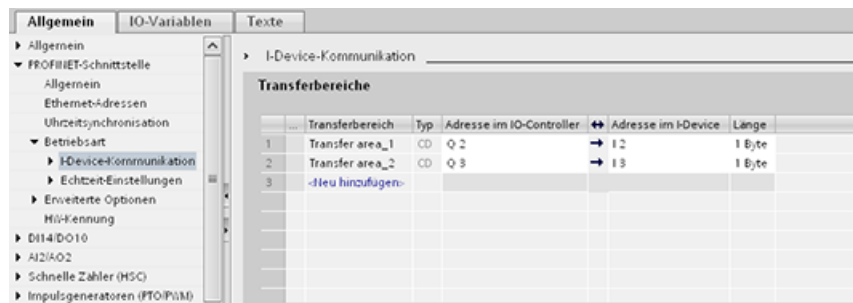
- Jetzt können Sie in der Klappliste "Zugewiesener IO-Controller" den IO-Controller auswählen. Nachdem Sie den IO-Controller ausgewählt haben, werden in der Netzsicht die Vernetzung und das IO-System zwischen beiden Geräten angezeigt.



- Über das Kontrollkästchen "Parametrierung der PN-Schnittstelle durch übergeordneten IO-Controller" geben Sie an, ob die Schnittstellenparameter vom I-Device selbst oder von einem übergeordneten IO-Controller zugewiesen werden. Wenn Sie das I-Device in einem untergeordneten IO-System betreiben, können die Parameter der PROFINET-Schnittstelle des I-Device (z. B. der Portparameter) nicht vom übergeordneten IO-Controller zugewiesen werden.

- Konfigurieren Sie die Transferbereiche. Die Transferbereiche finden Sie im Navigationsbereich "I-Device-Kommunikation":

- Klicken Sie in der Spalte "Transferbereich" in das erste Feld. STEP 7 weist einen Standardnamen zu, den Sie ändern können.
- Wählen Sie die Art der Kommunikationsbeziehung aus: Derzeit können Sie lediglich CD oder F-CD auswählen.
- Adressen werden automatisch voreingestellt. Sie können die Adressen bei Bedarf korrigieren und die Länge des Transferbereichs, der beständig übertragen wird, bestimmen.



- Für jeden Transferbereich wird im Navigationsbereich ein eigener Eintrag angelegt. Wenn Sie einen dieser Einträge auswählen, können Sie die Details des Transferbereichs anpassen, sie korrigieren und Kommentare dazu eingeben.

Hinweis

Wenn Sie eine S7-1200 als I-Device konfigurieren, beträgt die maximale Größe eines Übertragungsbereichs 1024 Eingangs- oder Ausgangsbytes. Je nach lokaler E/A und Einschränkungen beim Adressraum auf dem steuernden Gerät gibt es möglicherweise begrenzende Faktoren.

Konfigurieren eines I-Device mit einer GSD-Datei

Wenn Sie ein I-Device in einem anderen Projekt nutzen oder wenn das I-Device in einem anderen Engineering-System verwendet wird, konfigurieren Sie den übergeordneten IO-Controller und das I-Device wie oben beschrieben.

Klicken Sie jedoch nach der Konfiguration der Transferbereiche auf die Schaltfläche "Exportieren", damit eine neue GSD-Datei für das I-Device erstellt wird. Diese GSD-Datei stellt das konfigurierte I-Device in anderen Projekten dar.

Die Schaltfläche "Exportieren" finden Sie im Inspektorfenster im Bereich "I-Device-Kommunikation".

Die Hardwarekonfiguration wird übersetzt und der Export-Dialog geöffnet.

Weisen Sie dem I-Device-Proxy einen Namen zu und geben Sie in die vorgesehenen Felder eine Beschreibung ein. Klicken Sie auf die Schaltfläche "Exportieren", um den Vorgang abzuschließen.

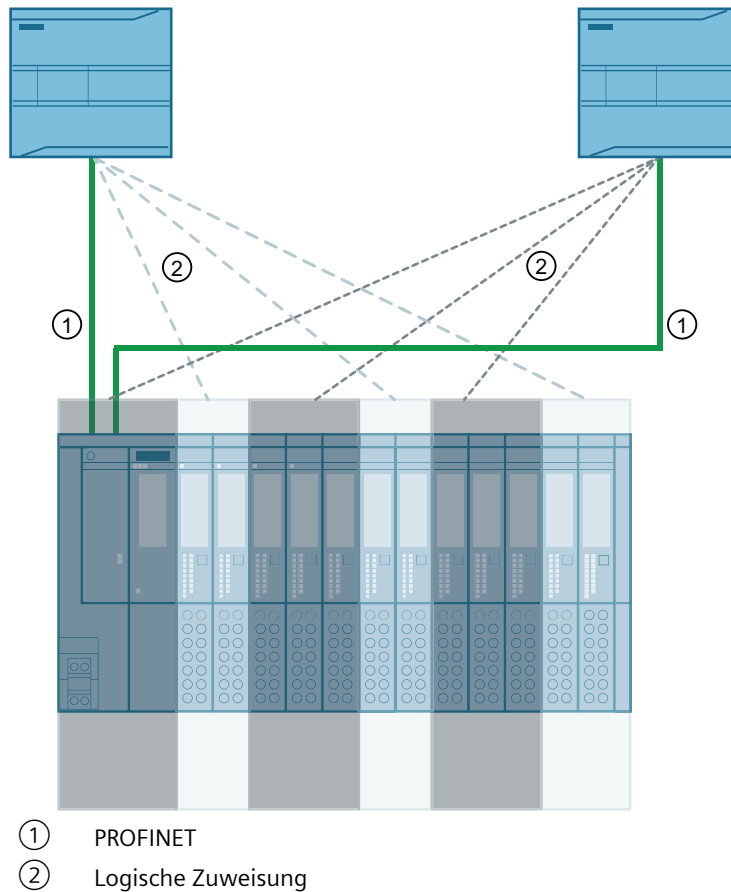
Dann können Sie die GSD-Datei beispielsweise in ein anderes Projekt importieren.

11.5.14 Shared Devices**11.5.14.1 Shared-Device-Funktion**

In größeren und weiter verteilten Systemen werden häufig zahlreiche IO-Controller eingesetzt.

Ohne die "Shared-Device"-Funktion ist jedes E/A-Modul eines IO-Device dem gleichen IO-Controller zugewiesen. Wenn Sensoren, die sich physisch nahe beieinander befinden, verschiedenen IO-Controllern Daten liefern müssen, sind dafür mehrere IO-Devices erforderlich.

Mit der "Shared-Device"-Funktion können die Module oder Submodule eines IO-Device auf verschiedene IO-Controller aufgeteilt werden. Somit sind flexible Automatisierungskonzepte möglich. Sie haben beispielsweise die Möglichkeit, E/A-Module, die sich nahe beieinander befinden, in einem IO-Device zu verbinden.



Prinzip

Der Zugriff auf die Submodule des Shared Device wird dann auf die einzelnen IO-Controller aufgeteilt. Jedes Submodul des Shared Device wird exklusiv einem IO-Controller zugewiesen.

Voraussetzung (GSD-Konfiguration)

- STEP 7 V12 Service Pack 1 oder höher
- S7-1200 CPU mit Firmware ab V4.1 als IO-Controller
- Unterstützung der Shared-Device-Funktion durch das IO-Device, z. B. Schnittstellenmodul IM 155-5 PN ST
- GSD-Datei für die Konfiguration des IO-Device ist installiert
- Eine als I-Device konfigurierte S7-1200 CPU unterstützt die Shared-Device-Funktion. Sie müssen die PROFINET GSD-Datei für das I-Device aus STEP 7 (ab V5.5) exportieren und dann in STEP 7 (TIA-Portal) importieren.

Zugriff konfigurieren

Das IO-Device muss in verschiedenen Projekten vorhanden sein, so dass die Module oder Submodule eines IO-Device verschiedenen IO-Controllern zugewiesen werden können. Für jeden IO-Controller ist ein eigenes Projekt erforderlich.

Mit dem Parameter "Shared Device" des Schnittstellenmoduls bestimmen Sie die Module und Submodule, auf die der IO-Controller Zugriff hat:

- Wenn der lokale IO-Controller Zugriff auf das konfigurierte Modul hat, wählen Sie den Namen des IO-Controllers in der Liste aus.
- Wenn der IO-Controller eines anderen Projekts und nicht der lokale IO-Controller auf das konfigurierte Modul Zugriff haben soll, wählen Sie den Eintrag "---".

Die Konfiguration ist bezüglich des Zugriffs konsistent, wenn jedes Modul oder Submodul in genau einem Projekt einem IO-Controller zugewiesen ist.

Modul oder Submodul ist einem anderen IO-Controller zugewiesen

Im folgenden Abschnitt werden die Konsequenzen der Einstellung "---" für den Parameter "Shared Device" aus Sicht des lokalen IO-Controllers beschrieben.

In diesem Fall hat der lokale IO-Controller keinen Zugriff auf das in dieser Weise konfigurierte Modul. Dies bedeutet im Einzelnen:

- Kein Datenaustausch mit dem Modul oder Submodul
- Kein Empfang von Alarm- und Diagnosemeldungen, also keine Anzeige des Diagnosestatus in der Onlinesicht
- Keine Parametrierung des Moduls oder Submoduls

Einstellung der Echtzeiteigenschaften

STEP 7 berechnet die Kommunikationslast und somit die resultierenden Aktualisierungszeiten. Sie müssen die Anzahl der projektexternen IO-Controller in dem Projekt, in dem die PROFINET-Schnittstelle des Shared Device dem IO-Controller zugewiesen ist, angeben, damit eine Berechnung mit Shared-Device-Konfigurationen möglich ist.

Die maximal mögliche Anzahl von IO-Controllern für das Shared Device ist vom Gerät abhängig. Diese Anzahl ist in der GSD-Datei des Shared Device gespeichert.

Sie können einen sehr kurzen Sendetakt (mindestens 1 ms) mit einer S7-1200 CPU als IO-Controller festlegen. Der Sendetakt kann kürzer sein als der kürzeste vom Shared Device unterstützte Sendetakt. In diesem Fall wird das Shared Device vom IO-Controller mit einem Sendetakt betrieben, der vom Gerät unterstützt wird (Sendetaktanpassung).

Beispiel: Eine CPU unterstützt Sendetakte ab 1 ms. Ein konfiguriertes IO-Device unterstützt Sendetakte ab 1,25 ms, ein anderes IO-Device unterstützt Sendetakte ab 1 ms. In diesem Fall haben Sie die Möglichkeit, den kurzen Sendetakt von 1 ms für die CPU einzustellen. Die CPU betreibt das "langsame" IO-Device mit dem Sendetakt von 1,25 ms.

Regeln für die Konfiguration

- IO-Controller, die das Shared Device verwenden, werden in verschiedenen Projekten angelegt. In jedem Projekt muss darauf geachtet werden, dass das Shared Device in jeder Station identisch konfiguriert ist. Nur ein einziger IO-Controller darf jemals vollständigen Zugriff auf ein Submodul haben. Inkonsistenzen in der Konfiguration führen zu einem Fehler des Shared Device.
- E/A-Adressen eines Moduls oder Submoduls dürfen nur bearbeitet werden, wenn ein Modul oder Submodul dem IO-Controller im gleichen Projekt zugewiesen ist.
- Das Shared Device muss in jedem Projekt die gleichen IP-Parameter und den gleichen Gerätenamen haben.
- Der Sendetakt muss bei allen IO-Controllern, die Zugriff auf das Shared Device haben, identisch sein.
- Die S7-Subnetz-ID des Subnetzes, mit dem das Shared Device verbunden ist, muss in allen Projekten identisch sein.
- Die folgenden Funktionen sind nur verfügbar, wenn die PROFINET-Schnittstelle des Shared Device dem lokalen IO-Controller zugewiesen ist:
 - Priorisierter Anlauf
 - Parametrierung der Porteigenschaften

Grenzbedingungen

Die folgenden Grenzbedingungen entstehen, weil eine Shared-Device-Konfiguration über mehrere Projekte verteilt ist:

- Die Adressen von Modulen und Submodulen, die diesem IO-Controller nicht zugewiesen sind, fehlen in der Adressübersicht jedes IO-Controllers, der Zugriff auf ein Shared Device hat.
- Die nicht zugewiesenen Module und Submodule werden bei der Konsistenzprüfung nicht in der Grenzwertberechnung der Konfiguration für das Shared Device berücksichtigt. Deshalb müssen Sie selbst prüfen, ob die maximale Anzahl Submodule bzw. die maximale Menge zyklischer IO-Daten für das Shared Device nicht überschritten wird. Informationen zu den maximalen Werten entnehmen Sie der Dokumentation der Geräte, die Sie verwenden.
- Konfigurationsfehler wie die Zuweisung eines Moduls oder Submoduls zu mehreren IO-Controllern werden in STEP 7 nicht erkannt.
- CPUs, in die eine Shared-Device-Konfiguration geladen wurde, haben keine Informationen dazu, ob es sich bei dem IO-Device um ein Shared Device handelt. Module und Submodule, die anderen IO-Controllern und damit anderen CPUs zugewiesen sind, fehlen in der geladenen Konfiguration. Diese Module und Submodule werden deshalb weder auf dem CPU-Webserver noch in der CPU-Anzeige angezeigt.

11.5.14.2 Beispiel: Ein Shared Device konfigurieren (GSD-Konfiguration)

In diesem Beispiel wird beschrieben, wie Sie mit STEP 7 ab V13 SP1 ein dezentrales E/A-System als Shared Device konfigurieren.

Eine "dezentrale" Konfiguration mit unterschiedlichen Engineering-Tools für unterschiedliche IO-Controllerfamilien ist möglich. Die nachfolgend beschriebene Vorgehensweise basiert auf

STEP 7 V13 SP1 und ist begrenzt auf die Konfiguration mit zwei IO-Controllern der S7-1200-Serie, die ein Shared Device gemeinsam nutzen.

Im Beispiel werden zwei Projekte mit jeweils einem IO-Controller angelegt:

- Controller1
- Controller2

Sie müssen das Shared Device in beiden Projekten erstellen, auch wenn es sich physisch um ein und dasselbe IO-Gerät handelt.

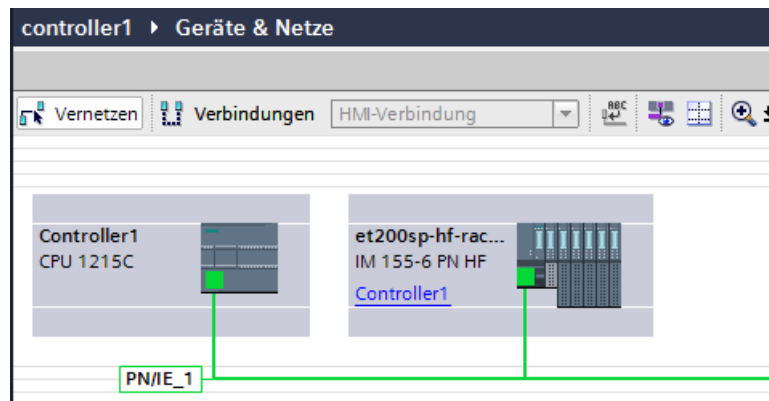
Voraussetzungen

- STEP 7 ab V13 SP1
- Unterstützung der Shared-Device-Funktion durch das IO-Gerät (z. B. ET 200SP IM 155-6 PN HF V3.1).
- GSD-Datei für die Konfiguration des IO-Geräts als Shared Device ist installiert.

Vorgehensweise: Projekt 1 anlegen

Um das erste Projekt mit einem Shared Device anzulegen, gehen Sie wie folgt vor:

1. Starten Sie STEP 7.
2. Legen Sie ein neues Projekt mit dem Namen "Controller1" an.
3. Fügen Sie eine CPU 1215C aus dem Hardwarekatalog in die Netzsicht ein. Benennen Sie es "Controller1".
4. Fügen Sie ein IO-Gerät mit der "Shared-Device"-Funktion (zum Beispiel ein ET 200SP) aus dem Hardwarekatalog ein (Hardwarekatalog: Weitere Feldgeräte > PROFINET IO > E/A).
5. Weisen Sie den IO-Controller "Controller1" dem IO-Gerät zu.

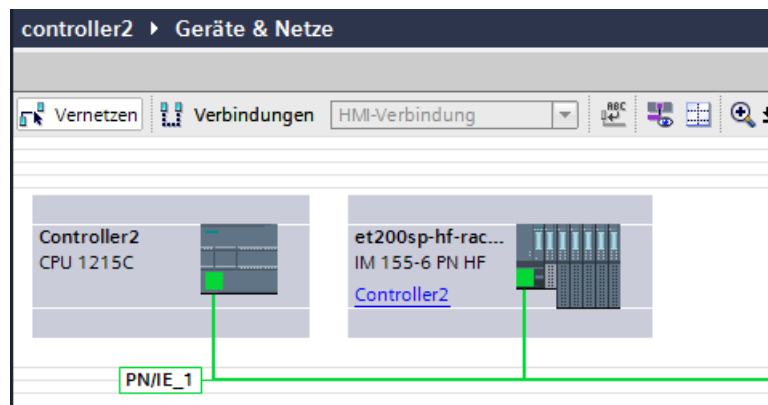


6. Doppelklicken Sie auf das IO-Gerät und fügen Sie alle erforderlichen Module und Submodule aus dem Hardwarekatalog in die Geräteübersichtstabelle ein.
7. Weisen Sie die Modulparameter zu.
8. Speichern Sie das Projekt.

Vorgehensweise: Projekt 2 anlegen

Um das zweite Projekt mit einem Shared Device anzulegen, gehen Sie wie folgt vor:

1. Starten Sie STEP 7 erneut.
Eine neue Instanz von STEP 7 wird geöffnet.
2. Legen Sie in der neuen Instanz ein neues Projekt mit dem Namen "Controller2" an.
3. Fügen Sie eine CPU 1215C in die Netzsicht ein. Benennen Sie sie "Controller2".
4. Kopieren Sie das IO-Gerät aus dem Projekt "Controller1" und fügen Sie es in die Netzsicht von Projekt "Controller2" ein.
5. Weisen Sie den IO-Controller "Controller2" dem IO-Gerät zu.



6. Speichern Sie das Projekt.

Beide Projekte haben jetzt ein identisch strukturiertes IO-Gerät, das im nächsten Schritt für die unterschiedlichen Arten des IO-Controllerzugriffs konfiguriert werden muss.

Vorgehensweise: Zugriff auf das Shared Device konfigurieren

Die Module und Submodule, die Sie in das Shared Device einfügen, werden automatisch der lokalen CPU zugewiesen. Um die Zuweisung zu ändern, gehen Sie wie folgt vor:

1. Wählen Sie in der Netz- oder Gerätesicht von Projekt "Controller1" das Schnittstellenmodul aus.
2. Wählen Sie den Bereich "Shared Device" im Inspektorfenster aus.
Eine Tabelle zeigt für alle konfigurierten Module, welche CPU auf die Module und Submodule Zugriff hat. Die Standardeinstellung ist, dass die lokale CPU auf alle Module und Submodule Zugriff hat.

- Behalten Sie die Einstellung "Controller1" für alle Module und Submodule bei, die im Adressbereich der lokalen CPU verbleiben sollen.
Wählen Sie die Einstellung "---" für alle Module und Submodule aus, die sich im Adressbereich der CPU aus dem Projekt "Controller2" befinden sollen. Dies bedeutet, dass ein IO-Controller außerhalb des Projekts Zugriff auf das Modul bzw. Submodul erhält.

Name	Zugriff
et200sp-hf-rack-comm	Controller1
PROFINET interface	Controller1
Port_1	Controller1
Port_2	Controller1
CM PtP_1	Controller1
CM PtP_2	---
Server module_1	Controller1

- Wählen Sie in der Netz- oder Gerätesicht von Projekt "Controller2" das Schnittstellenmodul aus.
- Wählen Sie den Bereich "Shared Device" im Inspektorfenster aus.
Eine Tabelle zeigt für alle konfigurierten Module, welche CPU auf die Module und Submodule Zugriff hat.
- Wählen Sie die Einstellung "---" für alle Module und Submodule, die sich im Adressbereich der CPU aus dem Projekt "Controller1" befinden sollen.

Name	Zugriff
et200sp-hf-rack-comm	---
PROFINET interface	---
Port_1	---
Port_2	---
CM PtP_1	---
CM PtP_2	Controller2
Server module_1	---

7. Prüfen Sie schließlich, ob sich die Einstellungen für den Zugriff auf die einzelnen Module und Submodule in beiden Projekten ergänzen. Damit ist gemeint, dass, wenn die lokale CPU in einem Projekt Zugriff hat, im anderen Projekt die Option "---" eingestellt sein muss und umgekehrt.
Hinweis: Die Option "---" für die PROFINET-Schnittstelle und somit für die Ports bewirkt, dass die zugehörigen Parameter schreibgeschützt sind und sich nicht ändern lassen. Parameter der PROFINET-Schnittstelle und Portparameter können nur in dem Projekt bearbeitet werden, in dem die PROFINET-Schnittstelle der lokalen CPU zugewiesen ist. Die Ports können unabhängig hiervon in beiden Projekten miteinander verschaltet sein.
8. Prüfen Sie, ob in allen Projekten die gleichen IP-Adressparameter und der gleiche Gerätenamen für das Shared Device festgelegt sind.
Prüfen Sie, ob in allen Projekten für das Subnetz, mit dem das Shared Device verbunden ist, die gleiche S7-Subnetz-ID eingestellt ist (Subnetzeigenschaften, Bereich "Allgemein" im Inspektorfenster).

Hinweis

Wenn Sie Änderungen am Shared Device vornehmen: Nehmen Sie die gleichen Änderungen in jedem Projekt auf dem Shared Device vor. Achten Sie darauf, dass nur genau ein IO-Controller Zugriff auf ein Modul oder Submodul hat.

Vorgehensweise: Echtzeiteinstellungen anpassen

Um zu gewährleisten, dass alle IO-Controller und Shared Devices mit dem passenden Sendetakt betrieben werden und dass die Aktualisierungszeiten anhand der Kommunikationslast richtig berechnet werden, müssen Sie die folgenden Einstellungen prüfen und anpassen:

1. Wählen Sie das Projekt aus, dessen IO-Controller Zugriff auf die PROFINET-Schnittstelle und die Ports des Shared Device hat.
2. Wählen Sie in der Netzansicht das Schnittstellenmodul des Shared Device aus.
3. Navigieren Sie im Inspektorfenster zum Bereich "PROFINET-Schnittstelle > Erweiterte Optionen > Echtzeit-Einstellungen > IO-Zyklus".
4. Legen Sie im Bereich "Shared Device" die Anzahl der projektexternen IO-Controller fest. Die maximale Anzahl ist vom IO-Gerät abhängig (in der GSD-Datei angegeben).
5. Sie müssen für jeden IO-Controller, der Zugriff auf die Module und Submodule des Shared Device hat, den gleichen Sendetakt einstellen:

- Wenn Sie den IO-Controller mit STEP 7 (TIA Portal) konfigurieren:
 - Öffnen Sie das entsprechende Projekt.
 - Wählen Sie die PROFINET-Schnittstelle des IO-Controllers aus.
 - Wählen Sie den Bereich "Erweiterte Optionen > Echtzeit-Einstellungen > IO-Kommunikation" im Inspektorfenster aus und legen Sie den Sendetakt fest.
- Wenn Sie den IO-Controller mit einem anderen Engineering-Tool konfigurieren:
 - Wählen Sie die PROFINET-Schnittstelle des Shared Device in STEP 7 (TIA Portal) aus und lesen Sie den Sendetakt des Shared Device aus (Bereich "Erweiterte Optionen > Echtzeit-Einstellungen").
 - Geben Sie den ausgelesenen Sendetakt in das Engineering-Tool ein.

Hinweis

Wenn Sie alle IO-Controller, die Zugriff auf das Shared Device haben, in STEP 7 (TIA Portal) konfigurieren, können Sie kürzere Sendetakte auf dem IO-Controller einstellen, als vom Shared Device unterstützt werden (Sendetaktanpassung).

Übersetzen und Laden

Sie müssen die Konfigurationen für die verschiedenen IO-Controller übersetzen und nacheinander in die CPUs laden.

Aufgrund der dezentralen Konfiguration bei getrennten Projekten gibt STEP 7 bei falscher Zugriffsparametrierung keine Konsistenzfehler aus. Hier einige Beispiele für falsche Zugriffsparametrierung:

- Mehrere IO-Controller haben Zugriff auf das gleiche Modul
- IP-Adressparameter oder Sendetakte sind nicht identisch

Diese Fehler werden erst im Controllerbetrieb gemeldet und als Konfigurationsfehler ausgegeben.

11.5.14.3 Beispiel: Ein I-Device als Shared Device konfigurieren

In diesem Beispiel wird beschrieben, wie Sie eine S7-1200 mit STEP 7 ab Version V13 SP1 als I-Device konfigurieren und sie dann in zwei Projekten als Shared Device einsetzen.

Eine "dezentrale" Konfiguration mit unterschiedlichen Engineering-Tools für unterschiedliche IO-Controllerfamilien ist möglich. Die nachfolgend beschriebene Vorgehensweise basiert auf STEP 7 V13 SP1 und ist begrenzt auf eine Konfiguration mit zwei IO-Controllern der S7-1200-Familie, die die Übertragungsbereiche eines I-Device als Shared Device gemeinsam nutzen. Das I-Device selbst ist eine CPU 1215C.

Im Beispiel werden drei Projekte mit jeweils einem IO-Controller angelegt:

- S7-1200-I-Device
- Controller1
- Controller2

Mit dem Projekt S7-1200-I-Device konfigurieren Sie das I-Device. Sie verwenden die PROFINET GSD-Variante von S7-1200-I-Device in den Projekten Controller1 und Controller2, um die Übertragungsbereiche in dem jeweiligen übergeordneten IO-Controller zuzuweisen.

Konzept des Shared I-Device

Für das Konzept des Shared I-Device werden mindestens drei getrennte Projekte benötigt:

- I-Device-Projekt: Sie konfigurieren und programmieren ein I-Device, um eine bestimmte Automatisierungsaufgabe durchzuführen. Sie definieren Übertragungsbereiche als die E/A-Schnittstelle für die übergeordneten Controller und weisen diese Übertragungsbereiche unterschiedlichen IO-Controllern zu. Für die Verbindung zu übergeordneten IO-Controllern stellen Sie eine PROFINET GSD-Datei bereit und verwenden die Übertragungsbereiche für den Zugriff auf das I-Device.
- Controller, die das I-Device gemeinsam nutzen (zwei Projekte): Sie verwenden das I-Device als eine PROFINET GSD-Variante bei der Konfiguration des PROFINET IO-Systems und geben dabei die E/A-Adressen an, unter denen die IO-Controller auf die Übertragungsbereiche zugreifen.

Hinweis

Wenn Sie eine S7-1200 als I-Device konfigurieren, beträgt die maximale Größe eines Übertragungsbereichs 1024 Eingangs- oder Ausgangsbytes. Je nach lokaler E/A und Einschränkungen beim Adressraum auf dem steuernden Gerät gibt es möglicherweise begrenzende Faktoren.

I-Device

Sie weisen die folgenden Parameter für eine S7-1200 CPU als I-Device zu:

- Zentrale und dezentrale Peripherie
- Gewünschte Übertragungsbereiche
- Anzahl der IO-Controller mit Zugriff auf dieses I-Device (bei einem Shared Device immer größer als 1)

Hinweis

Sie konfigurieren das I-Device ohne einen übergeordneten IO-Controller. Das Ergebnis ist, dass Sie nur die lokalen E/A-Adressen des Übertragungsbereichs (entspricht der "Adresse im I-Device") verwenden können, um das Anwenderprogramm zur Bearbeitung der Adressen aus dem Übertragungsbereich anzulegen. Sie laden das I-Device, vollständig konfiguriert mit Ausnahme der Verbindung zum übergeordneten IO-Controller, in die S7-1200 CPU.

Sie exportieren eine PROFINET GSD-Datei aus der I-Device-Konfiguration.

Controller, die das I-Device gemeinsam nutzen

Sie müssen die aus der I-Device-Konfiguration erstellte PROFINET GSD-Datei in alle Engineering-Systeme laden, die Sie bei der Konfiguration eines PROFINET IO-Systems mit diesem Shared I-Device verwenden. Wenn Sie alle Verwendungen dieses I-Device mit STEP 7 V13 SP1 konfigurieren, ist es ausreichend, die GSD-Datei in STEP 7 zu installieren.

Sie konfigurieren das I-Device als GSD-Variante auf dem PROFINET IO-System in den beteiligten Projekten. In STEP 7 V13 SP1 finden Sie dieses I-Device nach der Installation unter "Weitere Feldgeräte > PROFINET IO > PLCs & CPs".

In jedem der beteiligten Projekte weisen Sie ausschließlich den übergeordneten IO-Controllern Übertragungsbereiche zu (Standardeinstellung: alle). Für die anderen Übertragungsbereiche stellen Sie "---" ein (nicht zugewiesen). Hierdurch kann der lokale IO-Controller nicht auf diesen Übertragungsbereich zugreifen und Sie können den Übertragungsbereich einem anderen IO-Controller in einem anderen Projekt zuweisen.

Voraussetzungen

- STEP 7 ab V13 SP1
- Unterstützung der Shared-Device-Funktion durch das IO-Device (z. B. ET 200SP IM 155-6 PN HF V3.1).
- GSD-Datei für die Konfiguration des IO-Device als Shared Device ist installiert.

Vorgehensweise: Projekt S7-1200-I-Device anlegen

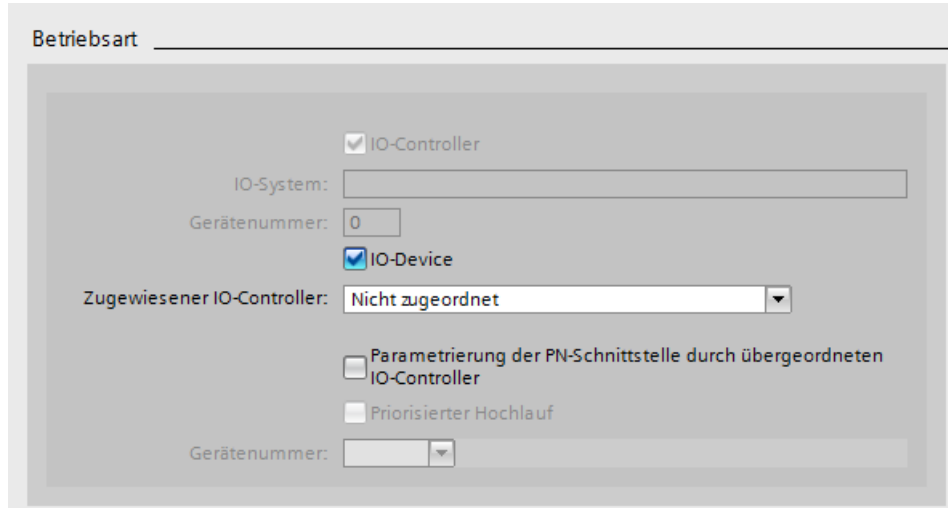
Um das Projekt mit einem Shared I-Device anzulegen, gehen Sie wie folgt vor:

1. Starten Sie STEP 7.
2. Legen Sie ein neues Projekt mit dem Namen "S7-1200-I-Device" an.
3. Fügen Sie eine CPU 1215C aus dem Hardwarekatalog in die Netzsicht ein. Weisen Sie den Namen "S7-1200-I-Device" zu.



4. Doppelklicken Sie auf das IO-Device und konfigurieren Sie alle erforderlichen Module und Submodule.

- 5. Weisen Sie die Modulparameter zu. Insbesondere müssen Sie die folgenden Einstellungen für die CPU im Bereich der PROFINET-Schnittstelle [X1] konfigurieren:
 - Aktivieren Sie die Option "IO-Device" im Bereich "Betriebsart".



- Konfigurieren Sie die Übertragungsbereiche im Bereich "Betriebsart > I-Device-Konfiguration". Die Spalte "Adresse im IO-Controller" bleibt leer, weil kein IO-Controller zugewiesen ist.

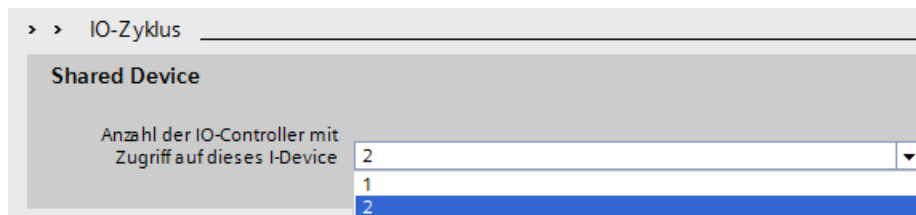
> I-Device-Kommunikation

Transferbereiche

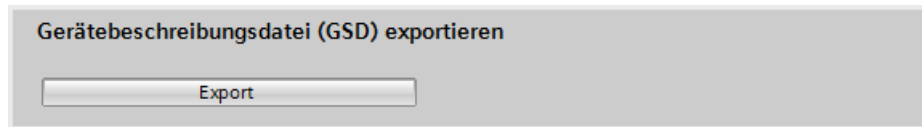
...	Transferbereich	Typ	Adresse im IO-Controller	↔	Adresse im I-Device	Länge
1	Transfer_area_1	CD		→	I 200...299	100 Byte
2	Transfer_area_2	CD		←	Q 200...299	100 Byte
3	Transfer_area_3	CD		→	I 300...399	100 Byte
4	Transfer_area_4	CD		←	Q 300...399	100 Byte
5	<Neu hinzufügen>					

Hinweis: Um einen Eingangsbereich in einen Ausgangsbereich zu ändern oder umgekehrt, müssen Sie in den Bereich des entsprechenden Übertragungsbereichs navigieren.

- Wählen Sie die Anzahl der IO-Controller (mindestens zwei), die im Betrieb auf das Shared I-Device zugreifen (Bereich "Betriebsart > Echtzeit-Einstellungen", Bereich "Shared Device").



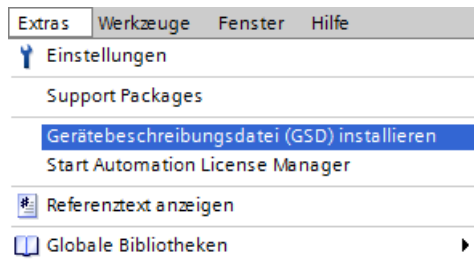
6. Speichern Sie das Projekt.
7. Klicken Sie auf die Schaltfläche "Exportieren" (Bereich "Betriebsart > I-Device-Konfiguration", Abschnitt "Allgemeine Gerätebeschreibungsdatei (GSD) exportieren"). Wenn Sie den Namen im Exportdialog nicht ändern, erhält die GSD-Datei einen Namen im zugewiesenen Format (zum Beispiel "GSDML-V2.31-#Siemens-PreConf_S7-1200-I-Device-20130925-123456").



Vorgehensweise: Projekt Controller1 anlegen

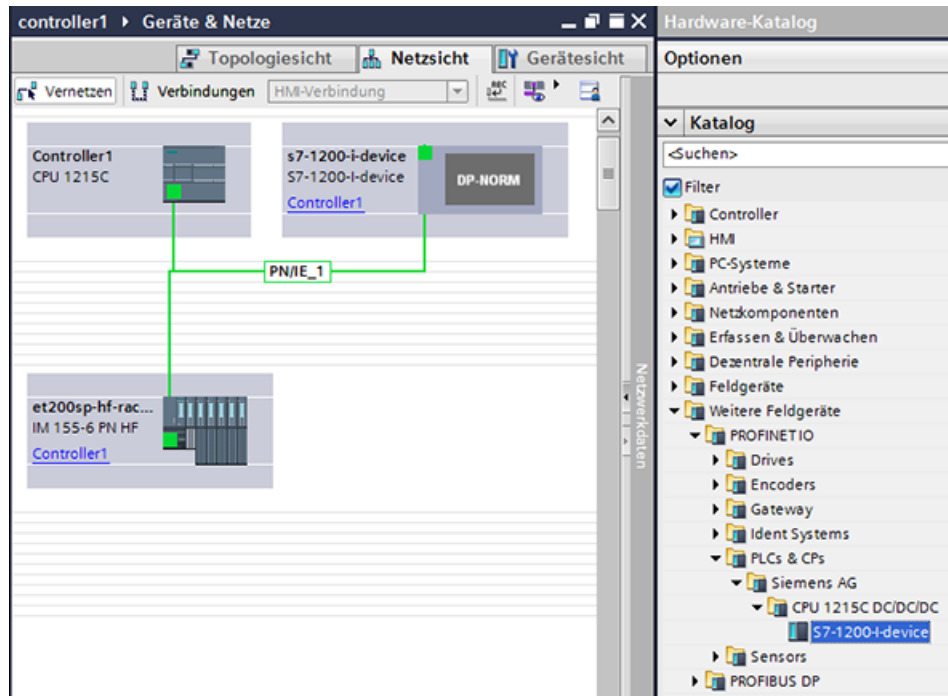
Um das erste Projekt mit einem Shared I-Device anzulegen, gehen Sie wie folgt vor:

1. Starten Sie STEP 7.
2. Installieren Sie die aus der I-Device-CPU exportierte PROFINET GSD-Datei (S7-1200-I-Device).



3. Legen Sie ein neues Projekt mit dem Namen "Controller1" an.
4. Fügen Sie die CPU 1215C in die Netzsicht ein. Der Name der CPU müsste "Controller1" lauten.
5. Fügen Sie das I-Device aus dem Hardwarekatalog ein (Hardwarekatalog: Weitere Feldgeräte > PROFINET IO > PLCs & CPs).

6. Weisen Sie den IO-Controller "Controller1" dem I-Device zu.



7. Wählen Sie den Bereich "Shared Device" in den Eigenschaften des I-Device aus:
- In der Tabelle sind alle Übertragungsbereiche und die PROFINET-Schnittstelle dem lokalen IO-Controller zugewiesen (Controller1).
 - Definieren Sie die Übertragungsbereiche, auf die die Controller1-CPU **keinen** Zugriff haben soll. Wählen Sie für diese Bereiche den Eintrag "---" aus. Diese Übertragungsbereiche werden Controller2 zur Verfügung gestellt.

The screenshot displays the SIMATIC Manager configuration environment. The top window, titled 'controller1 > Geräte & Netze', shows a network diagram with three main components: 'Controller1 CPU 1215C', 's7-1200-i-device S7-1200-I-device' (containing a 'DP-NORM' module), and 'et200sp-hf-rac... IM 155-6 PN HF'. A green line labeled 'PN/IE_1' connects the Controller1 CPU to the s7-1200-i-device. Below this, the 's7-1200-i-device [Module]' configuration window is open, showing the 'Shared Device' tab. A table lists the shared device areas and their access permissions.

Name	Zugriff
s7-1200-i-device	Controller1
Interface	---
Port 1	---
Port 2	---
Transfer_area_1	Controller1
Transfer_area_2	Controller1
Transfer_area_3	---
Transfer_area_4	Controller1

- Sie können die Adressen aus der Gerätesicht des IO-Controllers in der Geräteübersicht anpassen. Um die Geräteübersicht aufzurufen, doppelklicken Sie auf das I-Device.

Baugruppe	Baugr...	Steck...	E-Adresse	A-Adres...	Typ	Artikelnummer	Firmware	K...	Zugriff
s7-1200-i-device	0	1		256...355	S7-1200-I-device	6ES7 215-1AG40-0XB0	V4.1		Controller1
Transfer_area_1	0	1 1000		256...355	Transfer_area_1				Controller1
Transfer_area_2	0	1 1001	256...355		Transfer_area_2				Controller1
Transfer_area_3	0	1 1002			Transfer_area_3				--
Transfer_area_4	0	1 1003			Transfer_area_4				--
Interface	0	1 X1			s7-1200-i-device				--

- Speichern Sie das Projekt.

Vorgehensweise - Projekt Controller2 anlegen

Um das zweite Projekt mit einem Shared Device anzulegen, gehen Sie wie folgt vor:

- Starten Sie STEP 7 erneut.
Eine neue Instanz von STEP 7 wird geöffnet.
- Legen Sie in der neuen Instanz ein neues Projekt mit dem Namen "Controller2" an.
- Fügen Sie die CPU 1215C in die Netzsicht ein. Weisen Sie den Namen "Controller2" zu.
- Fügen Sie das I-Device aus dem Hardwarekatalog ein (Hardwarekatalog: Weitere Feldgeräte > PROFINET IO > PLCs & CPs).
- Weisen Sie den IO-Controller "Controller2" dem I-Device zu.
- Passen Sie den Zugriff auf die Übertragungsbereiche im Projekt Controller1 an. Stellen Sie sicher, dass keine doppelten Zuweisungen zustande kommen.
- Passen Sie die Parameter des Subnetzes und der PROFINET-Schnittstelle an. Weil das Shared I-Device dasselbe Gerät in verschiedenen Projekten verwendet, müssen diese Daten übereinstimmen.
- Speichern Sie das Projekt.

Beide Projekte haben jetzt ein identisch konfiguriertes Shared I-Device. Der IO-Controllerzugriff und die Parameter der PROFINET-Schnittstelle müssen während des nächsten Schritts trotzdem in den verschiedenen Projekten geprüft werden.

Zusammenfassung - Parametrierung des Zugriffs auf das Shared Device

Die Übertragungsbereiche werden automatisch dem lokalen IO-Controller zugewiesen. Um die Zuweisung zu ändern, gehen Sie wie folgt vor:

- Klicken Sie in der Netzsicht des Projekts "Controller1" auf das Gerät "S7-1200-I-Device" und wählen Sie den Bereich "Shared Device" aus.
- Eine Tabelle zeigt an, welche CPU auf die einzelnen konfigurierten Übertragungsbereiche Zugriff hat. Die Standardeinstellung ist, dass die lokale CPU auf alle Module und Submodule Zugriff hat.

3. Behalten Sie die Einstellung "Controller1" für alle Übertragungsbereiche bei, die im Adressbereich der lokalen CPU verbleiben sollen.
Wählen Sie die Einstellung "---" für alle Übertragungsbereiche aus, die sich im Adressbereich der CPU "Controller2" aus dem Projekt "Controller2" befinden sollen. Dies bedeutet, dass ein IO-Controller außerhalb des Projekts Zugriff auf den Übertragungsbereich erhält.
4. Gehen Sie für die übrigen Projekte genauso vor.
5. Prüfen Sie schließlich, ob sich die Einstellungen für den Zugriff auf die einzelnen Module und Submodule in beiden Projekten ergänzen. Damit ist gemeint, dass, wenn die lokale CPU in einem Projekt Zugriff hat, im anderen Projekt die Option "---" eingestellt sein muss und umgekehrt.
Hinweis: Die Option "---" für die PROFINET-Schnittstelle und somit für die Ports bewirkt, dass die zugehörigen Parameter schreibgeschützt sind und sich nicht ändern lassen. Parameter der PROFINET-Schnittstelle und Portparameter können nur in dem Projekt bearbeitet werden, in dem die PROFINET-Schnittstelle der lokalen CPU zugewiesen ist. Die Ports können unabhängig hiervon in beiden Projekten miteinander verschaltet sein.
6. Prüfen Sie, ob in allen Projekten die gleichen IP-Adressparameter und der gleiche Geräte name für das Shared Device festgelegt sind.
Prüfen Sie, ob in allen Projekten für das Subnetz, mit dem das Shared Device verbunden ist, die gleiche S7-Subnetz-ID eingestellt ist (Subnetzeigenschaften, Bereich "Allgemein" im Inspektorfenster).

Hinweis

Wenn Sie Änderungen am I-Device vornehmen (zum Beispiel die Anzahl oder Länge der Übertragungsbereiche ändern), exportieren Sie das I-Device erneut als GSD-Datei. Installieren Sie die GSD-Datei in jedem Projekt neu, in dem das I-Device als Shared Device verwendet wird. Achten Sie darauf, dass nur genau ein IO-Controller Zugriff auf einen Übertragungsbereich hat.

Hinweis

Wenn Sie die S7-1200 als Shared I-Device und als Controller verwenden, achten Sie darauf, dass Sie die Aktualisierungszeiten von PROFINET I-Device und PROFINET IO erhöhen, um die Auswirkungen auf die Kommunikationsleistung abzumildern. Das System ist sehr stabil und funktioniert gut, wenn Sie für ein einzelnes PROFINET I-Device eine Aktualisierungszeit von 2 ms auswählen und wenn Sie für ein einzelnes PROFINET IO eine Aktualisierungszeit von 2 ms auswählen.

Sie legen die Parameter für den "IO-Zyklus" im Konfigurationsdialog "Eigenschaften" des PROFINET IO-Device oder IO fest. Weitere Informationen finden Sie unter "IO-Zykluszeit konfigurieren" (Seite 739).

Vorgehensweise - Echtzeiteinstellungen anpassen

Um zu gewährleisten, dass alle IO-Controller und Shared Devices mit dem passenden Sendetakt betrieben werden und dass die Aktualisierungszeiten anhand der Kommunikationslast richtig berechnet werden, müssen Sie die folgenden Einstellungen prüfen und anpassen:

1. Sie müssen für jeden IO-Controller, der Zugriff auf die Module und Submodule des Shared Device hat, den gleichen Sendetakt einstellen:

- Wenn Sie den IO-Controller mit STEP 7 (TIA Portal) konfigurieren, gehen Sie wie folgt vor:
 - Öffnen Sie das entsprechende Projekt.
 - Wählen Sie die PROFINET-Schnittstelle des IO-Controllers aus.
 - Wählen Sie den Bereich "Erweiterte Optionen > Echtzeit-Einstellungen > IO-Kommunikation" im Inspektorfenster aus und legen Sie den Sendetakt fest.
- Wenn Sie den IO-Controller mit einem anderen Engineering-Tool konfigurieren, gehen Sie wie folgt vor:
 - Wählen Sie die PROFINET-Schnittstelle des Shared Device in STEP 7 (TIA Portal) aus und lesen Sie den Sendetakt des Shared Device aus (Bereich "Erweiterte Optionen > Echtzeit-Einstellungen").
 - Geben Sie den ausgelesenen Sendetakt in das Engineering-Tool ein.

Hinweis

Wenn Sie **alle** IO-Controller, die Zugriff auf das Shared I-Device haben, in STEP 7 (TIA Portal) konfigurieren, können Sie kürzere Sendetakte auf dem IO-Controller einstellen, als vom Shared Device unterstützt werden (Sendetaktanpassung).

Übersetzen und Laden

Sie müssen die Konfigurationen für die verschiedenen IO-Controller übersetzen und nacheinander in die CPUs laden.

Aufgrund der dezentralen Konfiguration bei getrennten Projekten gibt STEP 7 bei falscher Zugriffsparametrierung keine Konsistenzfehler aus. Hier einige Beispiele für falsche Zugriffsparametrierung:

- Mehrere IO-Controller haben Zugriff auf das gleiche Modul.
- IP-Adressparameter oder Sendetakte sind nicht identisch.

Diese Fehler werden erst im Controllerbetrieb gemeldet und als Konfigurationsfehler ausgegeben.

11.5.15 Medienredundanzprotokoll (MRP)

Bei der S7-1200 V4.5 und TIA Portal V17 unterstützen die folgenden CPUs die MRP-Funktionalität als MRP-Manager und als Client.

- CPU 1215C
- CPU 1217C
- CPU 1215FC

Als "Client" muss die 1215/1217 in einem MRP-Ring betrieben werden, indem MRP-Pakete über die Schnittstelle weitergeleitet und Verbindungsunterbrechungen an einen Manager im Netzwerk gemeldet werden.

Als "Manager" muss die 1215/1217 MRP-Pakete im Netzwerk senden, offene Ports im Ring erkennen, blockierte Ports verwalten und den Managerstatus mit anderen potentiellen Managern verhandeln.

Die Familie der S7-1200 CPUs hat drei Modelle (siehe oben), die das MRP-Protokoll und die Konfigurationsparameter für die Initialisierung des Betriebs als MRP-Client oder MRP-Manager unterstützen. Diese CPUs haben zwei PN-Ports.

Mit diesen drei CPUs können Sie einen MRP-Ring mit der S7-1200 als Client oder als Manager einrichten. Wenn die CPU als Manager fungiert, verwendet sie Testtelegramme, um zu prüfen und sicherzustellen, dass der Ring nicht unterbrochen ist. Wenn sie als Client fungiert, leitet sie diese Testtelegramme weiter, statt die Prüfung selbst vorzunehmen. So kann die S7-1200 bei Bedarf in beiden Rollen verwendet werden.

Hinweis

S7-1200 V4.4 und TIA Portal V16 und früher

Ab Version V4.4 und früher kann die S7-1200 nur als Client in einem MRP-Ring eingerichtet werden.

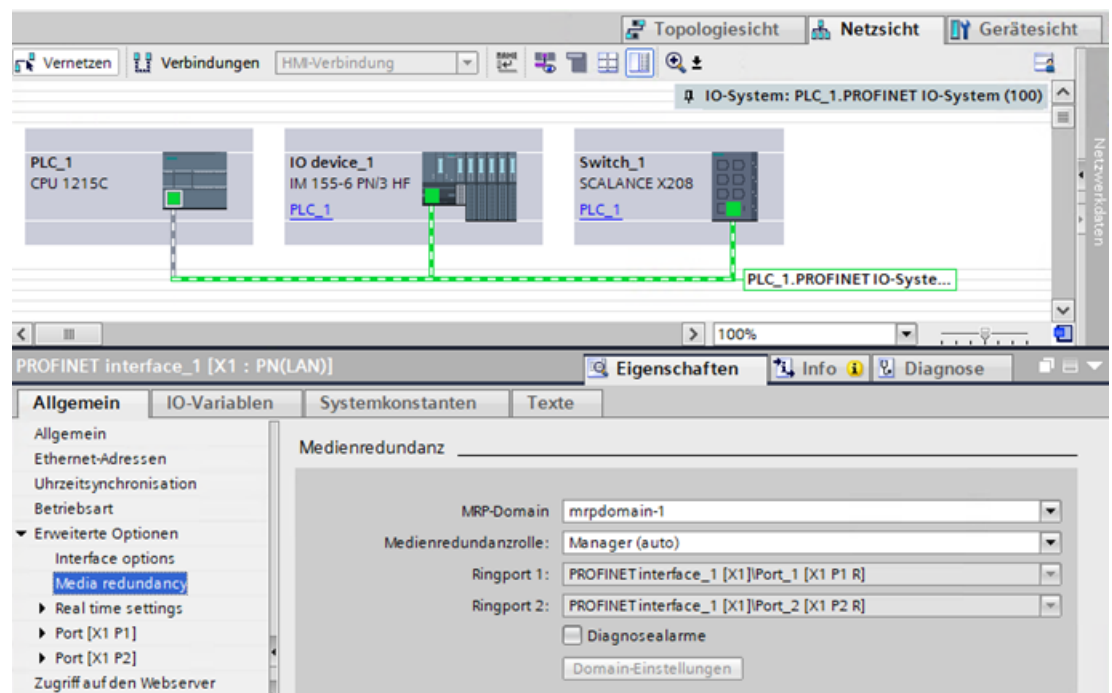


Bild 11-1 Medienredundanzkonfiguration im TIA Portal

MRP-Manager (Auto) und Manager

Sie können die S7-1200 (1215C/1217C) als Manager und Manager (Auto) in einem Media Redundancy Ring (MRP) verwenden. MRP ermöglicht, die Geräte in einer Ringkonfiguration zu verbinden. Der MRP-Manager zwingt Daten üblicherweise, in einer linearen Richtung zu fließen, indem ein Port des Rings blockiert wird. Wenn in dieser Ringkonfiguration eine Unterbrechung auftritt, erkennt der Manager die Unterbrechung und hebt die Blockierung des Ports auf, so dass die Daten in die andere Richtung fließen können. Mit MRP kann ein Netzwerk auch bei Drahtbruch oder Geräteausfall funktionsfähig bleiben. Die MRP-Spezifikation gestattet maximal 50 Geräte in einer Ringkonfiguration.

MRP-Manager

Der MRP-Manager gestattet der S7-1200 PLC, als Redundanzmanager zu fungieren. Bei einer Konfiguration in dieser Rolle arbeitet die S7-1200 mit Testtelegrammen, um mit Client-PLCs zu kommunizieren und sicherzustellen, dass die Verbindung im Ring nicht unterbrochen ist. Wird eine Unterbrechung zu einem Client erkannt, informiert die als MRP-Manager fungierende S7-1200 die Clientgeräte im Ring über die Veränderung und verwendet sofort die Ringports. Im TIA Portal kann nur ein Gerät als MRP-Manager in einem MRP-Ring eingerichtet werden.

MRP-Manager (Auto)

Wenn mehreren Geräten die Rolle MRP-Manager (Auto) zugewiesen ist, handeln sie den Managerstatus untereinander aus. Wenn der ausgehandelte MRP-Manager von der Konfiguration getrennt wird, verhandeln die verbleibenden Geräte mit der Einstellung "Manager (Auto)" die Managerrolle untereinander neu, bis die ursprüngliche Konfiguration wiederhergestellt wird. Wenn der ursprüngliche Manager erfolgreich zurückkehrt, holt er sich den Managerstatus zurück und die ursprüngliche Konfiguration wird wieder aufgenommen. MRP-Managerkonfigurationen sind auf der S7-1200 (1215C/1217C) erst ab V4.5 möglich.

Hinweis

Neukonfiguration des Rings

Die Neukonfiguration des Rings kann bis zu 200 ms in Anspruch nehmen. Deshalb muss für die PROFINET-Überwachungszeit für jedes Gerät ein Wert größer als 200 ms eingestellt werden.

Hinweis

MRP-Manager (Auto) ist die Standardeinstellung

Ist kein Projekt vorhanden, befindet sich ein Gerät V4.5 standardmäßig im Modus MRP-Manager (Auto). Das ist wichtig zu wissen, wenn Sie ein betriebsbereites Gerät in einer Nicht-Ring-Topologie anschließen und Testtelegramme in Ihrem Netzwerk bemerken.

S7-1200 (1214C, 1212C und 1211C CPUs) kann die Funktionen MRP-Manager und MRP-Manager (Auto) nicht aktivieren. Die Menüoptionen im TIA Portal stehen bei diesen CPUs nicht zur Verfügung.

Die S7-1200 unterstützt MRPD nicht, da die S7-1200 nicht IRT-fähig ist.

Diagnosealarme können im Fenster der Medienredundanzkonfiguration im TIA Portal ein- und ausgeschaltet werden, damit Manager und Client relevante MRP-Änderungsberichte liefern können wie:

- Ein Nachbargerät des Ringports unterstützt MRP nicht.
- Ein Ringport wird mit einem Portnachbarn verbunden, der kein Ringport ist.
- Ein Ringport ist mit einer anderen MPR-Domäne verbunden.
- (Nur Manager) Diagnosealarme bei Unterbrechung/Rückkehr, wenn der MPR-Ring unterbrochen wird und wenn die ursprüngliche Konfiguration zurückkehrt.

Siehe auch

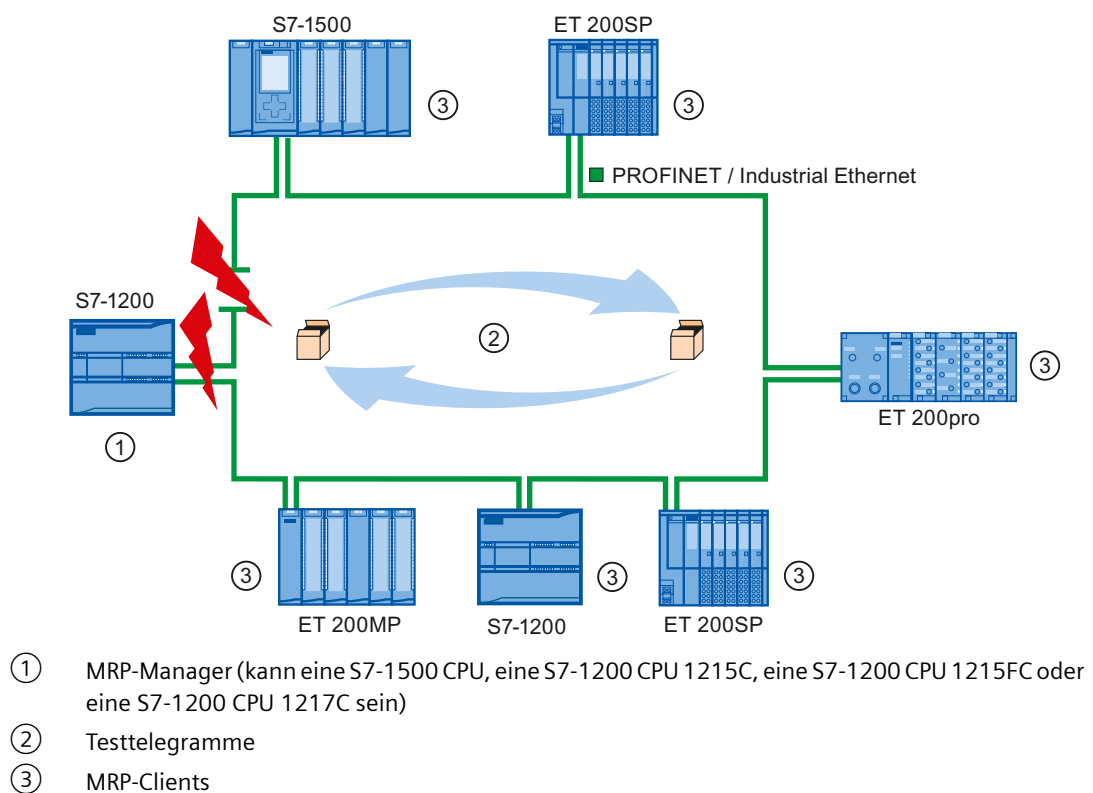
Neue Funktionen (Seite 35)

11.5.15.1 Medienredundanz bei Ringtopologien

Um die Netzwerkverfügbarkeit eines Industrial-Ethernet-Netzwerks mit optischen oder elektrischen linearen Bustopologien zu erhöhen, können Sie eine lineare Bustopologie in eine Ringtopologie umwandeln, indem Sie die Enden miteinander verbinden.

Geräte in einer Ringtopologie können IO-Devices, IO-Controller, externe Switches und/oder die integrierten Switches von Kommunikationsmodulen sein.

Um eine Ringtopologie mit Medienredundanz einzurichten, müssen Sie die beiden freien Enden einer linearen Bustopologie in einem Gerät zusammenbringen. Zum Schließen der linearen Bustopologie und Bilden eines Rings benötigen Sie zwei Ports (Ringports) eines Geräts im Ring. Ein Gerät im resultierenden Ring übernimmt dann die Rolle des MRP-Managers. Alle anderen Geräten im Ring sind MRP-Clients.



Die Ringports eines Geräts sind die Ports, die die Verbindung zu den zwei benachbarten Geräten in der Ringtopologie herstellen. Die Ringports werden in der Konfiguration des jeweiligen Geräts ausgewählt und festgelegt (ggf. auch voreingestellt).

Funktionsweise von Medienredundanz in einer Ringtopologie

Wenn der Ring an einer beliebigen Stelle unterbrochen wird, werden die Datenpfade zwischen den einzelnen Geräten automatisch neu konfiguriert. Die Geräte stehen nach der Neukonfiguration wieder zur Verfügung.

Im MRP-Manager ist einer der beiden Ringports im ununterbrochenen Netzwerkbetrieb für die normale Kommunikation blockiert, so dass keine Datentelegramme zirkulieren. Der MRP-Manager überwacht den Ring auf Unterbrechungen. Hierfür sendet er Testtelegramme von

Ringport 1 und Ringport 2. Die Testtelegramme laufen in beide Richtungen um den Ring, bis sie am anderen Ringport des MRP-Managers ankommen.

Eine Unterbrechung des Rings kann durch Verlust der Verbindung zwischen zwei Geräten oder durch Ausfall eines Geräts im Ring verursacht werden.

Wenn die Testtelegramme des MRP-Managers während einer Unterbrechung des Rings nicht mehr am anderen Ringport ankommen, verbindet der MRP-Manager seine zwei Ringports. Dieser Ersatzpfad stellt erneut eine funktionierende Verbindung zwischen allen übrigen Geräten in Form einer linearen Bustopologie her.

Der Zeitraum zwischen der Ringunterbrechung und der Wiederherstellung einer funktionsfähigen linearen Topologie wird Neukonfigurationszeit genannt.

Sobald die Unterbrechung beseitigt ist, werden die ursprünglichen Übertragungspfade wieder hergestellt, die zwei Ringports des MRP-Managers werden getrennt und die MRP-Clients über die Änderung informiert. Die MRP-Clients nutzen dann wieder die ursprünglichen Pfade zu den anderen Geräten.

Medienredundanzverfahren

Das Standardverfahren der Medienredundanz in SIMATIC ist das Medienredundanzprotokoll (MRP) mit einer typischen Neukonfigurationszeit von 200 ms. Bis zu 50 Geräte können pro Ring teilnehmen.

Hinweis

Neukonfiguration des Rings

Die Neukonfiguration des Rings kann bis zu 200 ms in Anspruch nehmen. Für die PROFINET-Überwachungszeit für jedes Gerät muss ein Wert größer als 200 ms eingestellt werden.

11.5.15.2 Einsetzen des Medienredundanzprotokolls (MRP)

Der MRP-Prozess funktioniert entsprechend dem Medienredundanzprotokoll (MRP), das in IEC 61158 Typ 10 "PROFINET" spezifiziert wird.

Voraussetzungen

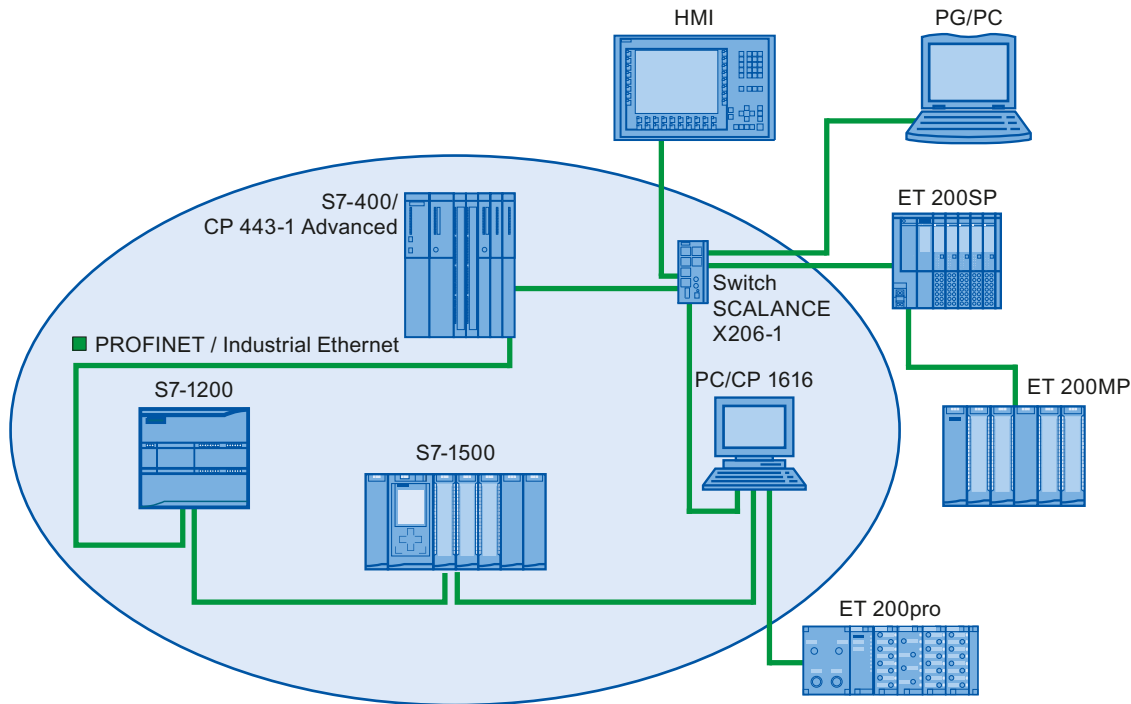
Die folgenden Voraussetzungen müssen für einen fehlerfreien Betrieb mit MRP erfüllt sein:

- Der Ring, in dem Sie das MRP verwenden möchten, darf nur Geräte enthalten, die diese Funktion unterstützen.
- "MRP" muss für alle Geräte im Ring aktiviert sein.
- Alle Geräte müssen über ihre Ringports miteinander verbunden sein.
- Mindestens ein MRP-Manager oder die Rolle "Manager" muss vorhanden sein.
- Der Ring darf maximal 50 Geräte umfassen. Ansonsten können Neukonfigurationszeiten von 200 ms oder mehr auftreten.
- Alle Partnerports im Ring müssen identische Einstellungen haben.

Topologie

Die folgende Abbildung zeigt eine mögliche Topologie für Geräte in einem Ring mit MRP. Die Geräte im schattiert dargestellten Oval befinden sich in der Redundanzdomäne.

Dies ist ein Beispiel für eine Ringtopologie mit MRP:



Die folgenden Regeln gelten für eine Ringtopologie mit Medienredundanz und MRP:

- Alle Geräte im Ring gehören zur gleichen Redundanzdomäne.
- Ein Gerät im Ring übernimmt die Rolle des MRP-Managers.
- Alle anderen Geräten im Ring sind MRP-Clients.

Sie können Geräte, die nicht MRP-fähig sind, über nicht als Ringports konfigurierte Ports mit dem Netzwerk verbinden. Diese Möglichkeit besteht nur bei Geräten mit mehr als zwei Ports (z.B. ein SCALANCE X Switch oder ein PC mit einem CP1616).

Grenzbedingungen

Sie haben die folgenden Kommunikationsmöglichkeiten:

- Mit dem MRP ist der MRP- und der RT: RT-Betrieb möglich.

Hinweis

Die RT-Kommunikation wird unterbrochen (Stationsausfall), wenn die Neukonfigurationszeit des Rings größer ist als die ausgewählte Überwachungszeit des IO-Device. Sie müssen für Ihre IO-Devices eine Überwachungszeit größer als 200 ms auswählen. Weitere Informationen finden Sie im nachstehenden Abschnitt "Überwachungszeit".

- MRP und TCP/IP (TSEND, HTTP, ...): Die TCP/IP-Kommunikation mit dem MRP ist möglich, da verloren gegangene Datenpakete gegebenenfalls erneut gesendet werden.
- MRP und priorisierter Anlauf:
 - Wenn Sie das MRP in einem Ring konfigurieren, können Sie bei den beteiligten Geräten die Funktion "priorisierter Anlauf" in PROFINET-Anwendungen nicht verwenden.
 - Wenn Sie die Funktion "priorisierter Anlauf" verwenden möchten, müssen Sie das MRP in der Konfiguration deaktivieren (das Gerät kann nicht Bestandteil des Rings sein).
- MRP auf PROFINET-Geräten mit mehr als zwei Ports: Wenn Sie ein PROFINET-Gerät mit mehr als zwei Ports in einem Ring betreiben, müssen Sie an den Ports, die nicht im Ring sind, eine Synchronisierungsgrenze festlegen. Durch Festlegen der Synchronisierungsgrenze definieren Sie eine Grenze für eine Synchronisierungsdomäne. Sie können keine Synchronisierungstelegramme zum Synchronisieren von Geräten in einer Synchronisierungsdomäne weiterleiten.

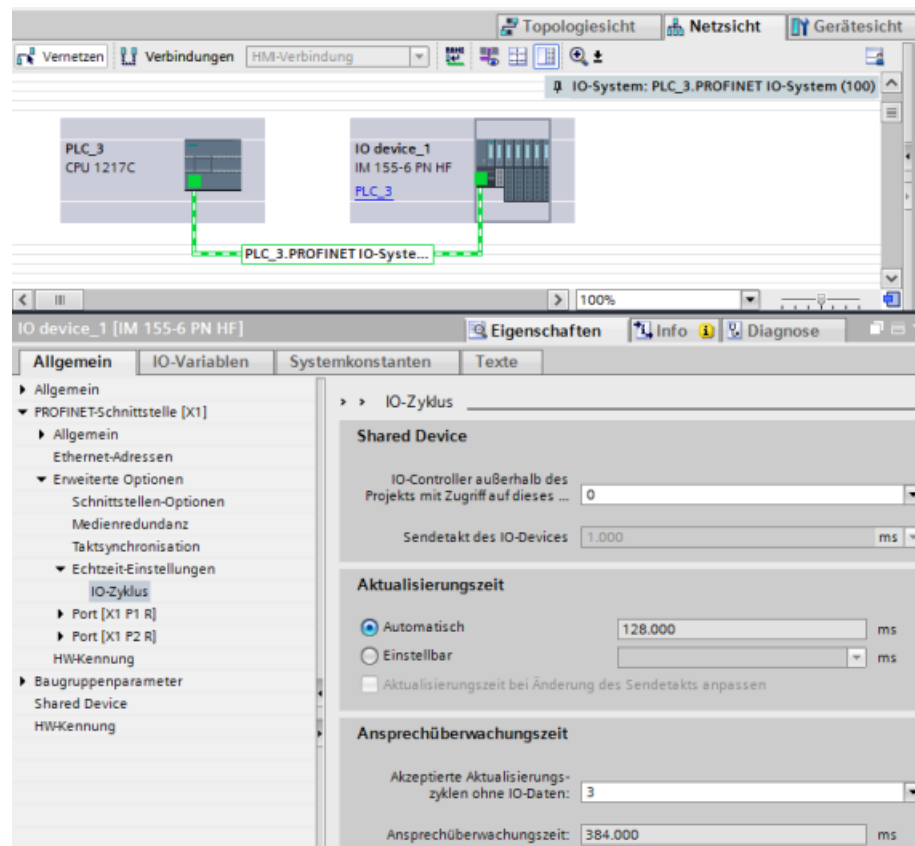
Überwachungszeit

Die PROFINET-Überwachungszeit ist das Zeitintervall, das ein IO-Controller oder IO-Device zulässt, ohne IO-Daten zu empfangen. Wenn das IO-Device vom IO-Controller nicht innerhalb der Überwachungszeit mit Daten versorgt wird, erkennt das Gerät die fehlenden Telegramme und gibt Ersatzwerte aus. Dies wird im IO-Controller als Stationsausfall gemeldet.

Sie können die Überwachungszeit für PROFINET IO-Devices konfigurieren. Geben Sie die Überwachungszeit nicht direkt ein, sondern als "Akzeptierte Anzahl von Aktualisierungszyklen, wenn IO-Daten fehlen". Die resultierende Überwachungszeit wird dann automatisch aus der Anzahl der Aktualisierungszyklen berechnet.

Um die Überwachungszeit zuzuweisen, gehen Sie wie folgt vor:

1. Wählen Sie in der Netz- oder Gerätesicht die PROFINET-Schnittstelle des IO-Device aus.
2. Navigieren Sie in den Eigenschaften der Schnittstelle zu: Erweiterte Optionen > Echtzeiteinstellungen > IO-Zyklus.
3. Wählen Sie die erforderliche Anzahl von Zyklen aus der Klappliste aus.



11.5.15.3 Medienredundanz konfigurieren

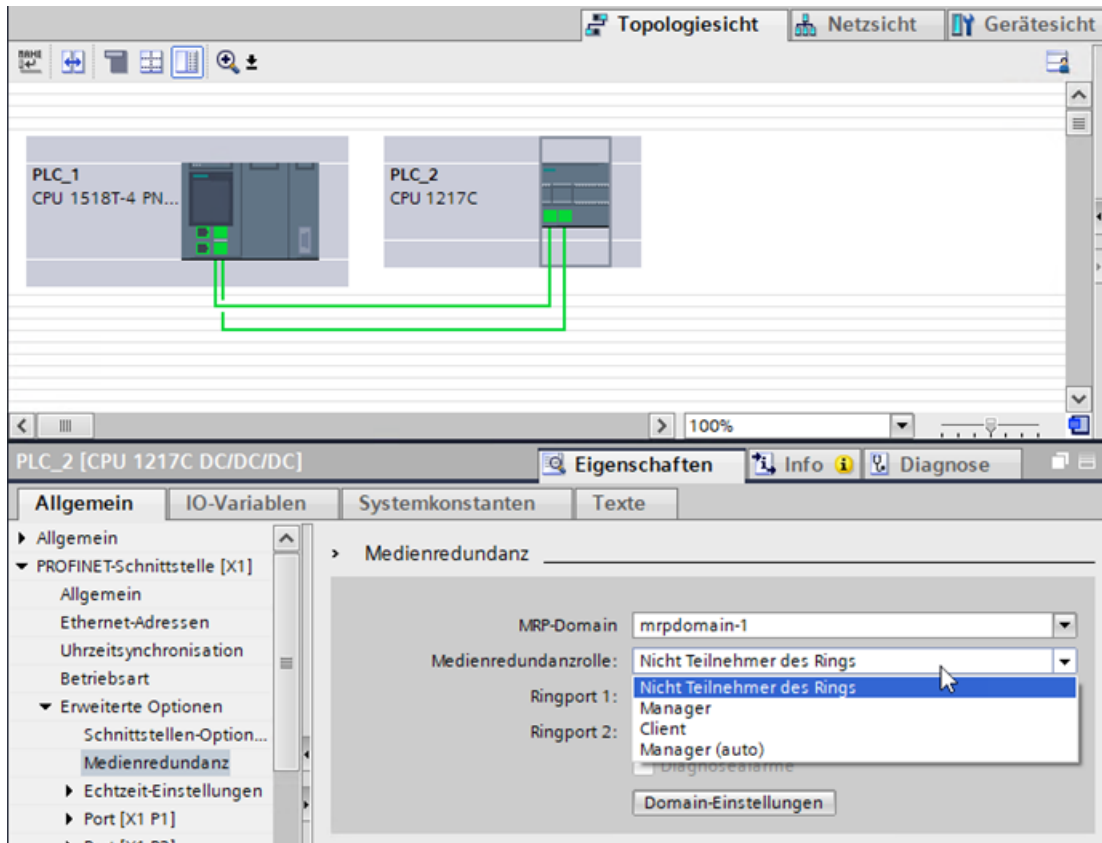
Alle Komponenten in Ihrer Anwendung müssen das Medienredundanzprotokoll (MRP) unterstützen.

Vorgehen

Um Medienredundanz zu konfigurieren, gehen Sie wie folgt vor:

1. Stellen Sie durch geeignete Portverschaltungen einen Ring her (z.B. in der Topologiesicht).
2. Wählen Sie ein PROFINET-Gerät aus, für das Sie Medienredundanz konfigurieren möchten.

3. Navigieren Sie im Inspektorfenster zu "PROFINET-Schnittstelle [X1] > Erweiterte Optionen > Medienredundanz".



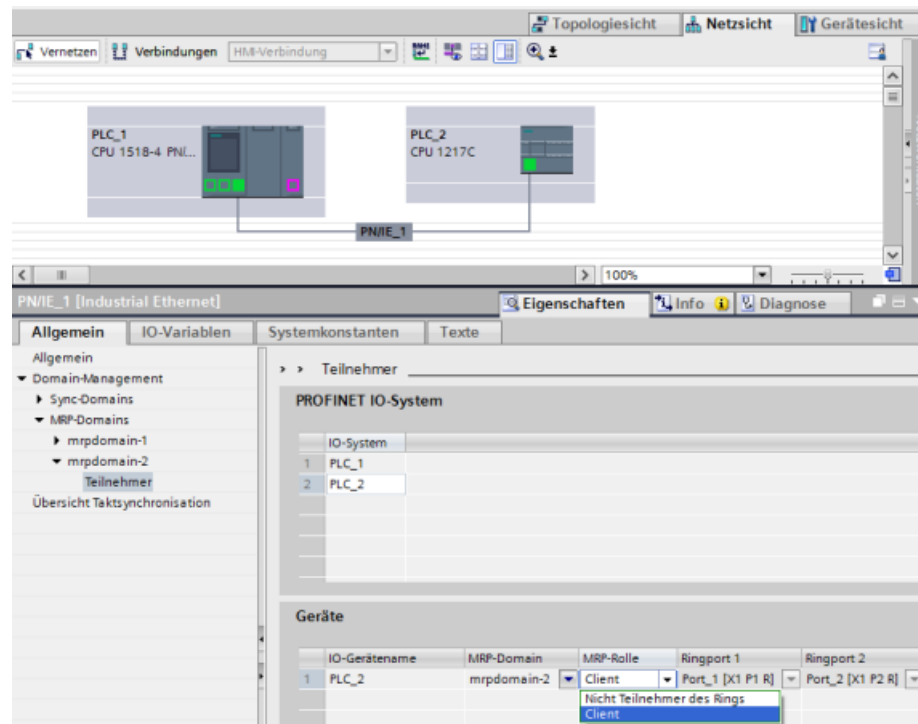
4. Weisen Sie unter „Medienredundanzrolle“ dem Gerät eine der Rollen "Manager (Auto)", "Client" oder "Kein Gerät im Ring" zu.
 Wenn Sie in der Topologiesicht im TIA Portal einen Ring konfigurieren, legt das TIA Portal die Medienredundanzrolle automatisch für Sie fest. Wenn ein Gerät ein Manager sein kann, legt das TIA Portal die Medienredundanzrolle als "Manager (Auto)" fest. Bei der S7-1200 wird als Medienredundanzrolle automatisch "Client" oder "Manager" festgelegt.

Hinweis

Bei V4.5 und TIA Portal V17 können Sie jetzt der S7-1200 CPU 1215C/1215FC/1217C die MRP-Rollen "Manager" oder "Manager (Auto)" zuweisen.

5. Wiederholen Sie die Schritte 2 bis 4 für alle PROFINET-Geräte im Ring.
 Oder:
 1. Markieren Sie das PROFINET IO-System in der Netzsicht.
 2. Klicken Sie auf das PROFINET IO-System.

3. Navigieren Sie zum Gerät der erforderlichen MRP-Domäne im Inspektorfenster.



4. Legen Sie für die PROFINET-Geräte eine der Rollen "Manager (Auto)", "Client" oder "Nicht Teilnehmer des Rings" zu.

Option für "Medienredundanz": MRP-Rolle

Abhängig vom verwendeten Gerät sind die Rollen "Manager", "Manager (Auto)", "Client" und "Nicht Teilnehmer des Rings" verfügbar.

Regeln:

- Ein Ring kann nur ein Gerät mit der Rolle "Manager" haben. Keine weiteren Geräte mit der Rolle "Manager" oder "Manager (Auto)" sind zulässig. Alle anderen Geräte im Ring können nur die Rolle "Client" haben. Geräte, die nicht im Ring sind, können die Rolle "Nicht Teilnehmer des Rings" haben.
- Wenn sich im Ring kein Gerät mit der Rolle "Manager" befindet, muss der Ring mindestens ein Gerät mit der Rolle "Manager (Auto)" haben. Ein Ring kann eine beliebige Anzahl von Geräten mit den Rollen "Client" und "Manager (Auto)" enthalten.

Option für "Medienredundanz": Ringport 1 und Ringport 2

Wählen Sie jeweils einzeln die Ports aus, die Sie als Ringport 1 oder Ringport 2 konfigurieren möchten. Die Klappliste zeigt die Auswahl möglicher Ports für jeden Gerätetyp an. Wenn die Ports im Werk eingestellt wurden, sind die Felder nicht verfügbar.

Hinweis

Die Konfiguration der Ringports ist bei der S7-1200 nicht erforderlich, weil die S7-1200 CPU nur zwei Ports hat.

Diagnosealarme

Wenn Diagnosealarme zum MRP-Zustand in der lokalen CPU ausgegeben werden sollen, aktivieren Sie das Kontrollkästchen "Diagnosealarme". Die folgenden Diagnosealarme können konfiguriert werden:

- **Verdrahtungs- oder Portfehler:**
Die CPU erzeugt Diagnosealarme für die folgenden Fehler an den Ringports:
 - Ein Nachbar des Ringports unterstützt das MRP nicht.
 - Ein Ringport ist mit einem Port verbunden, der kein Ringport ist.
 - Ein Ringport ist mit dem Ringport einer anderen MRP-Domäne verbunden.
- **Unterbrechung/Rückgabe (nur beim MRP-Manager):**
Wenn der Ring unterbrochen wird und die ursprüngliche Konfiguration ausgegeben wird, erzeugt die CPU Diagnosealarme. Wenn beide dieser Alarme innerhalb von 0,2 Sekunden voneinander auftreten, deutet dies auf eine Unterbrechung des Rings hin.

Sie können auf diese Ereignisse im Anwenderprogramm reagieren, indem Sie die angemessene Reaktion im Diagnosefehler-OB (OB 82) programmieren.

Hinweis

Fremdgeräte als MRP-Manager

Um fehlerfreien Betrieb zu gewährleisten, wenn ein Fremdgerät als MRP-Manager in einem Ring verwendet wird, müssen Sie, bevor Sie den Ring schließen, allen anderen Geräten im Ring die feste Rolle "Client" zuweisen. Ansonsten kann es zu Zirkulation von Datentelegrammen und Netzwerkausfall kommen.

11.5.16 S7-Routing

In der Netzsicht in STEP 7 können Sie eine komplexe Kommunikationstopologie erstellen, indem Sie Geräte in verschiedenen S7-Subnetzen miteinander verbinden. Sie können klassische S7-300/S7-400 CPUs und CPs sowie die neuesten S7-CPU und CPs vernetzen und HMI-Geräte und PC-Stationen wie einen OPC-Server aufnehmen.

Sobald Sie entschieden haben, welche Geräte kommunizieren müssen und die erforderlichen Verbindungen in STEP 7 hergestellt haben, kann das Engineering System (ES) die entsprechenden Routingtabellen als Bestandteil der Hardwarekonfiguration in die

verschiedenen S7-Router laden. Nachdem Sie die Routingtabelle in die verschiedenen S7-Router geladen haben, können das ES und andere Kommunikationspartner mit jedem Gerät kommunizieren, selbst wenn sich die Geräte in unterschiedlichen S7-Subnetzen befinden. Das ist möglich, weil die zwischengeschalteten CPUs und/oder CPs als S7-Router fungieren. Die CPUs und/oder CPs leiten eingehende Verbindungsanfragen solange an den nächsten S7-Router weiter, bis die Verbindungsanfrage das Zielgerät erreicht und die Geräte die S7-Verbindung aufbauen.

Die CPU nutzt das Verfahren zum Schreiben von Datensätzen, um die von den CP-Geräten im lokalen Grundgerät benötigten Routingtabellen zu übertragen. Die Routingtabellen bestimmen die Route von einem Gerät zu einem anderen zum Zeitpunkt einer Verbindungsanfrage, die die ID eines entfernten S7-Subnetzes enthält. Das Gerät, das die Verbindungsanfrage empfängt, befragt seine Routingtabelle, findet die nächste Station auf dem Pfad zum S7-Zielsubnetz und leitet die Verbindungsanfrage weiter. Auf diese Weise erreicht die Verbindungsanfrage schließlich das beabsichtigte Ziel, und die Antwort durchläuft die Route in umgekehrter Richtung.

Die S7-1200 CPUs haben eine einzige PN-Schnittstelle und bis zu drei CP-Geräte, die an den lokalen Kommunikationsbus angeschlossen sind. Deshalb haben Sie zwei Möglichkeiten des Routing innerhalb der S7-1200 Station:

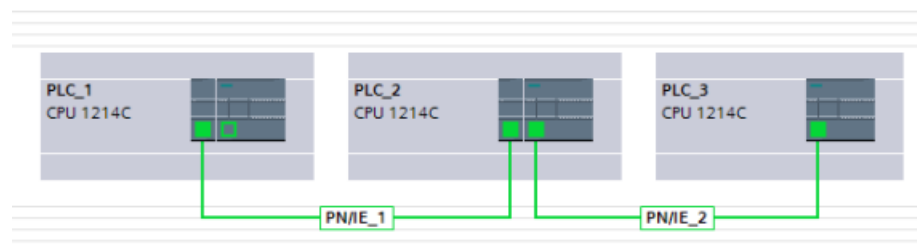
- Routing zwischen der CPU und einem CP
- Routing von einem CP zu einem anderen CP

Weitere Informationen zu allen S7-1200 CPs, die die Funktion "S7-Routing" unterstützen, finden Sie auf der Website Siemens Industry Online-Support, Produkt-Support. Der CP 1243-1 (<https://support.industry.siemens.com/cs/de/de/view/584459>) wird als Beispiel für einen CP gezeigt. Suchen Sie nach "S7-Routing".

11.5.16.1 S7-Routing zwischen CPU und CP-Schnittstellen

Da die S7-1200 CPUs auf eine einzige PN-Schnittstelle begrenzt sind, kann eine unabhängige CPU nicht als Router fungieren. Sie können eine unabhängige CPU immer nur jeweils an ein S7-Subnetz anschließen. Wenn Sie CP-Module im lokalen Grundgerät der CPU installieren, können Sie mehrere S7-Subnetze vernetzen und Routing einsetzen.

Im nachfolgenden Beispielsystem muss das Engineering-System (ES), damit PLC_1 mit PLC_3 kommunizieren kann, Meldungen über PLC_2 leiten. Das ES muss die Routingtabelle für PLC_2 laden, und PLC_2 muss die Routingtabelle für das CP-Modul in seinem lokalen Grundgerät bereitstellen. Sind diese Routingtabellen vorhanden, können PLC_1 und PLC_3 miteinander kommunizieren, auch wenn sie nicht direkt miteinander verbunden sind.



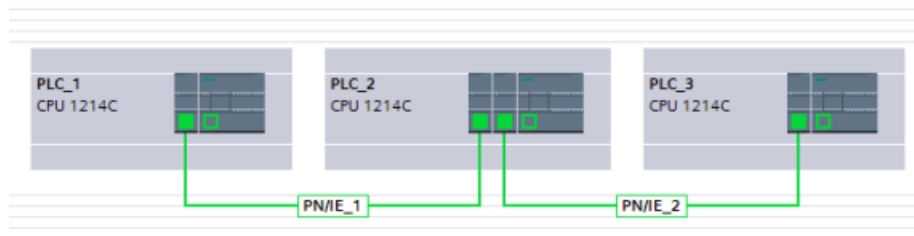
Um das Routing von einem S7-Subnetz zu dem anderen S7-Subnetz zu prüfen, muss PLC_1 eine Transportverbindung mit PLC_3 herstellen, und PLC_3 muss eine Verbindung mit PLC_1 aufbauen. Achten Sie hierbei darauf, dass das Routing von der PN/IE-Schnittstelle des PLC zu

einem CP-Modul sowie das Routing von einem CP-Modul zu der PN/IE-Schnittstelle des PLC möglich ist.

11.5.16.2 S7-Routing zwischen zwei CP-Schnittstellen

Da die S7-1200 CPUs bis zu CP-Module unterstützen, können Sie alle drei Module an unterschiedliche S7-Subnetze anschließen. Wenn Sie mindestens zwei CP-Module im lokalen Grundgerät der CPU installieren und mit verschiedenen S7-Subnetzen verbinden, können Sie Routing einsetzen.

Im nachfolgenden Beispielsystem muss das Engineering-System (ES), damit PLC_1 mit PLC_3 kommunizieren kann, Meldungen von PLC_2 vom CP-Modul zum CP-Modul im lokalen Grundgerät leiten. Das ES muss die Routingtabelle für PLC_2 laden, und PLC_2 muss die Routingtabelle für die zwei CP-Module bereitstellen. Sind diese Routingtabellen vorhanden, können PLC_1 und PLC_3 miteinander kommunizieren, auch wenn sie nicht direkt miteinander verbunden sind. Beachten Sie außerdem, dass das Routing von CP-Modul zu CP-Modul stattfindet, ohne dass Meldungen über die PN/IE-Schnittstelle von PLC_2 gesendet werden.



11.5.17 SNMP deaktivieren

Das Simple Network Management Protocol (SNMP) ist ein Internet-Standardprotokoll zum Erfassen und Organisieren von Informationen über verwaltete Geräte in IP-Netzwerken und zum Modifizieren dieser Informationen, um das Geräteverhalten zu ändern. Geräte, die üblicherweise SNMP unterstützen, sind Router, Switches, Server, Workstations, Drucker, Modemracks und mehr.

Der Einsatz von SNMP ist weit verbreitet in Netzwerkmanagementsystemen, um ans Netzwerk angeschlossene Geräte auf Bedingungen zu überwachen, die die Aufmerksamkeit des Administrators benötigen. SNMP verwendet verschiedene Dienste und Werkzeuge für die Erkennung und Diagnose der Netzwerktopologie. Informationen zu den Eigenschaften von SNMP-fähigen Geräten sind in MIB-Dateien (Management Information Base) enthalten, für die der Anwender die entsprechenden Rechte benötigt. SNMP liefert Managementdaten in Form von Variablen zu den verwalteten Systemen, in denen die Systemkonfiguration beschrieben wird. Diese Variablen können dann von verwaltenden Anwendungen abgefragt (und gelegentlich festgelegt) werden.

SNMP nutzt das UDP-Transportprotokoll und hat zwei Netzwerkkomponenten:

- SNMP-Manager: Überwacht die Netzwerkteilnehmer.
- SNMP-Client: Erfasst die verschiedenen netzwerkspezifischen Informationen in den einzelnen Netzwerkteilnehmern und speichert sie in strukturierter Form in der Management Information Base (MIB). Mit diesen Daten kann eine ausführliche Netzwerkdiagnose durchgeführt werden.

Unter bestimmten Bedingungen ist es in Ihrer Anwendung möglicherweise erforderlich, SNMP zu deaktivieren. Beispiele hierfür:

- Die Sicherheitseinstellungen in Ihrem Netzwerk gestatten den Einsatz von SNMP nicht.
- Sie nutzen Ihre eigene SNMP-Lösung (z. B. mit Ihren eigenen Kommunikationsanweisungen).

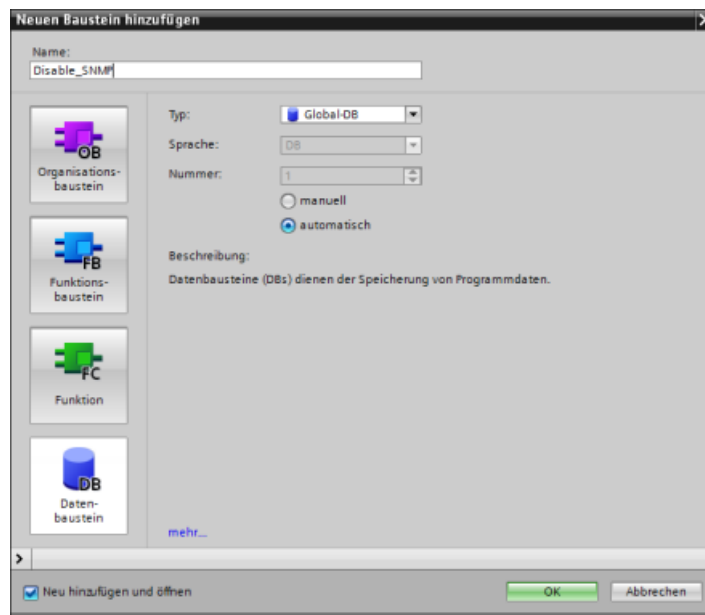
Wenn Sie SNMP für ein Gerät deaktivieren, stehen einige Optionen für die Diagnose der Netzwerktopologie nicht mehr zur Verfügung (z. B. das PRONETA-Tool oder der Webserver der CPU).

11.5.17.1 SNMP deaktivieren

SNMP deaktivieren

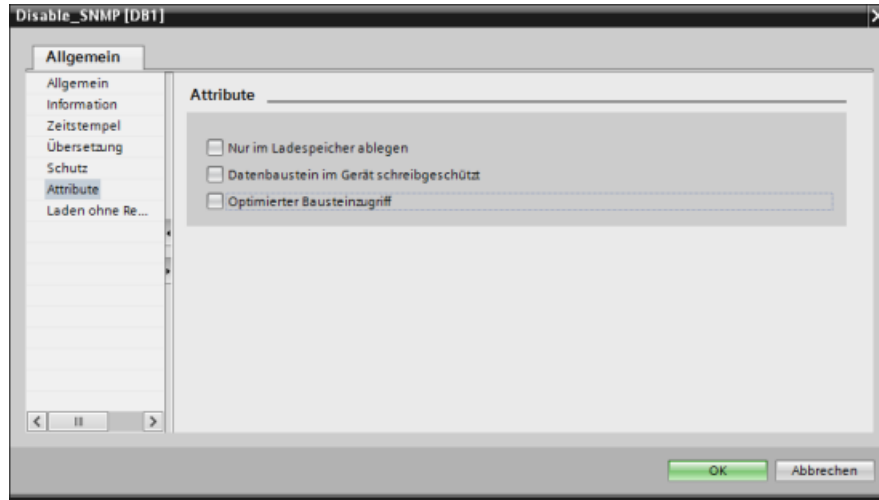
Gehen Sie wie folgt vor, um SNMP in der S7-1200 CPU zu deaktivieren:

1. Erstellen Sie einen klassischen Datenbaustein (DB):

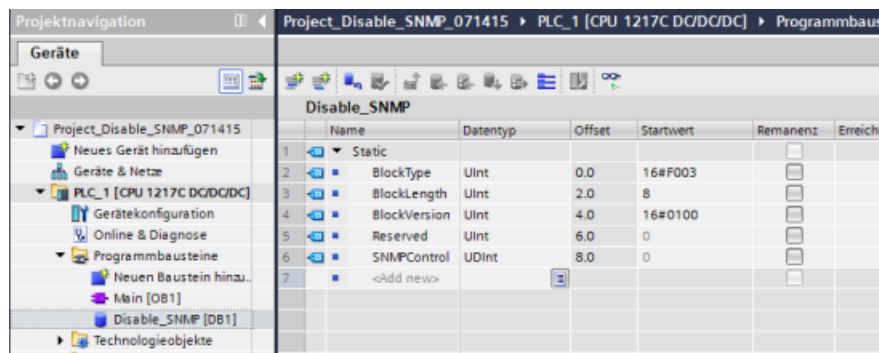


2. Wählen Sie die Eigenschaften des neu erstellten DBs aus.

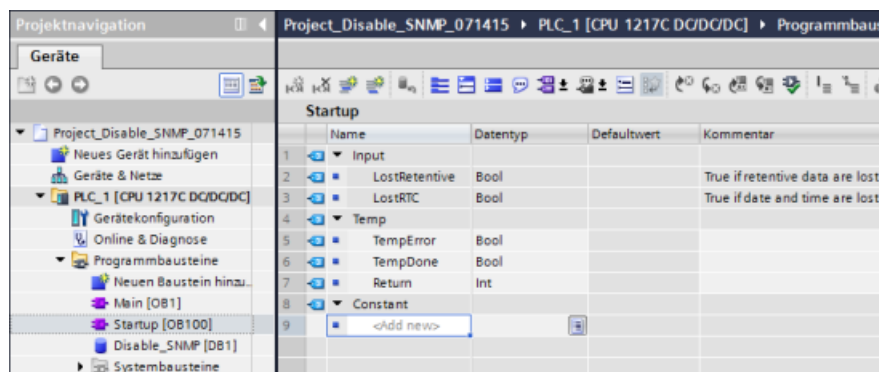
- Öffnen Sie das Register "Attribute". Deaktivieren Sie das Kontrollkästchen "Optimierter Bausteinzugriff":



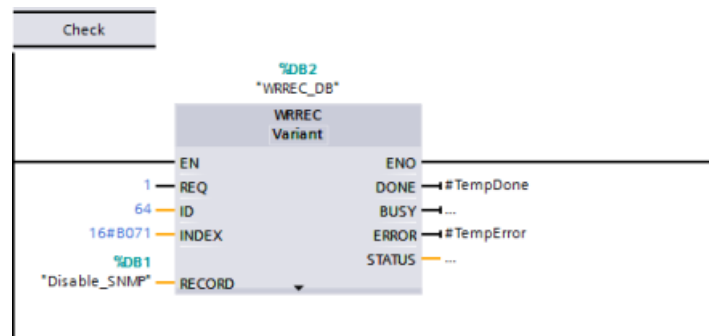
- Klicken Sie auf die Schaltfläche "OK". Daraufhin wird eine Meldung angezeigt, die Ihnen rät, Ihr Programm neu zu übersetzen. Übersetzen Sie Ihr Programm zu diesem Zeitpunkt neu.
- Erstellen Sie in der klassischen DB-Bausteinschnittstelle die folgenden statischen Variablen mit den gezeigten Werten. Diese Variablen verwenden Sie in Ihrem Programm, um die interne SNMP-Implementierung zu deaktivieren:



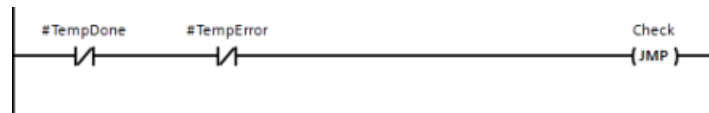
- Fügen Sie im Anlauf-OB (OB100) die temporären Variablen wie gezeigt hinzu:



7. Fügen Sie im KOP-Editor im Anlauf-OB (OB100) in Netzwerk 1 eine Anweisung Label (Sprungmarke) (im nachstehenden Beispiel heißt die Sprungmarke "Check") und eine Anweisung WRREC (Datensatz schreiben) mit den gezeigten Eingängen und Ausgängen ein:



8. Fügen Sie die folgende Schleife und den folgenden Prüfcode mit dem Ausgang "Zu Sprungmarke springen" (JMP) ein. Durch diesen Code wird gewährleistet, dass der Aufruf abgeschlossen wird und dass Sie SNMP vor dem Verlassen des Anlauf-OBs deaktivieren:



11.5.18 Diagnose

Informationen dazu, wie Sie Organisationsbausteine (OBs) für die Diagnose mit diesen Kommunikationsnetzwerken einsetzen, finden Sie unter "Organisationsbausteine (OBs)" (Seite 75).

11.5.19 Anweisungen für die dezentrale Peripherie

Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 371).

11.5.20 Diagnoseanweisungen

Weitere Informationen zur Verwendung dieser Anweisungen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseanweisungen" (Seite 420).

11.5.21 Diagnoseereignisse für die dezentrale Peripherie

Weitere Informationen zur Verwendung dieser Diagnoseinformationen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseereignisse für die dezentrale Peripherie" (Seite 469).

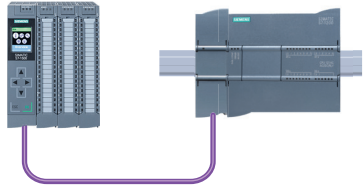
11.6 PROFIBUS

Ein PROFIBUS-System nutzt einen Bus-Master, um Slavegeräte abzufragen, die an mehreren Stellen auf einem seriellen RS485-Bus verteilt sind. Ein PROFIBUS-Slave ist ein beliebiges Peripheriegerät (E/A-Wandler, Ventil, Motorantrieb oder Messgerät), das Daten verarbeitet und die Ausgabe an den Master sendet. Der Slave stellt eine passive Station im Netzwerk dar, weil er keine Buszugriffsrechte besitzt. Er kann lediglich empfangene Meldungen quittieren oder auf Anforderung Antwortmeldungen an den Master senden. Alle PROFIBUS-Slaves haben die gleiche Priorität und die gesamte Netzwerkkommunikation stammt vom Master.

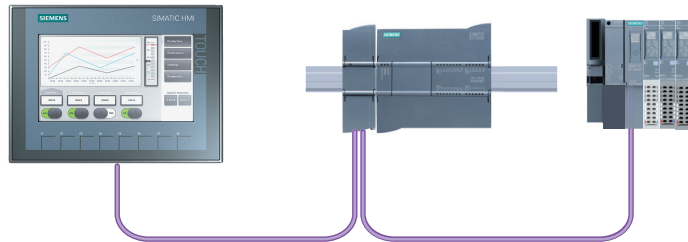
Ein PROFIBUS-Master stellt eine "aktive Station" im Netzwerk dar. PROFIBUS DP definiert zwei Klassen von Mastern. Ein Master der Klasse 1 (normalerweise eine zentrale programmierbare Steuerung (PLC) oder ein PC mit Spezialsoftware) bearbeitet die normale Kommunikation bzw. den Datenaustausch mit den ihm zugewiesenen Slaves. Ein Master der Klasse 2 (üblicherweise ein Konfigurationsgerät, z. B ein Laptop oder eine Programmierkonsole für Inbetriebnahme, Wartung und Diagnose) ist ein Sondergerät, das hauptsächlich für die Inbetriebnahme von Slaves und zu Diagnosezwecken eingesetzt wird.

Die S7-1200 ist mit dem Kommunikationsmodul CM 1242-5 als DP-Slave an ein PROFIBUS-Netzwerk angeschlossen. Das CM 1242-5 (DP-Slave) kann der Kommunikationspartner von DP-Mastern V0/V1 sein. Wenn Sie das Modul in einem Fremdsystem konfigurieren möchten, finden Sie auf der im Lieferumfang des Moduls enthaltenen CD sowie auf den Seiten des Siemens Automation-Kundensupports (<https://support.industry.siemens.com/cs/ww/de/ps/6GK7242-5DX30-0XE0>) im Internet eine GSD-Datei für das CM 1242-5 (DP-Slave).

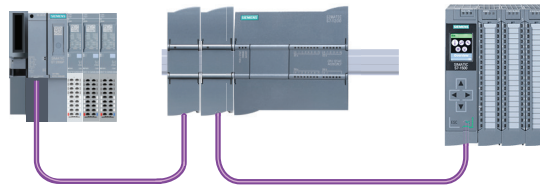
In der folgenden Abbildung fungiert die S7-1200 als DP-Slave einer S7-300 Steuerung:



Die S7-1200 ist mit dem Kommunikationsmodul CM 1243-5 als DP-Master an ein PROFIBUS-Netzwerk angeschlossen. Das CM 1243-5 (DP-Master) kann der Kommunikationspartner von DP-Slaves V0/V1 sein. In der folgenden Abbildung ist die S7-1200 ein Master und steuert einen ET200SP DP-Slave.



Wenn ein CM 1242-5 und ein CM 1243-5 zusammen installiert sind, kann eine S7-1200 gleichzeitig sowohl als Slave eines übergeordneten DP-Mastersystems als auch als Master eines untergeordneten DP-Slavesystems fungieren.



In V3.0 können Sie maximal drei PROFIBUS-CMs je Station konfigurieren, wobei es sich um jede beliebige Kombination aus DP-Master-CMs oder DP-Slave-CMs handeln kann. DP-Master in einer Implementierung mit einer CPU ab Firmware-Version 3.0 können jeweils maximal 32 Slaves steuern.

Die Konfigurationsdaten der PROFIBUS-CMs werden in der lokalen CPU gespeichert. Dadurch ist bei Bedarf der einfache Austausch dieser Kommunikationsmodule möglich.

Um PROFIBUS mit S7-1200 CPUs ab Version 4.0 einzusetzen, müssen Sie die PROFIBUS Master CM-Firmware auf mindestens V1.3 hochrüsten.

Hinweis

Halten Sie die PROFIBUS CM-Firmware stets auf dem neuesten Stand (<http://support.automation.siemens.com/WW/view/de/42131407>). Ein Firmware-Update können Sie auf eine der folgenden Arten durchführen:

- Mit den Online- und Diagnosewerkzeugen von STEP 7 (Seite 1212)
 - Mit einer SIMATIC Memory Card (Seite 131)
 - Über die Standard-Webseite "Modulinformationen" des Webservers (Seite 872)
 - Mit dem SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/de/view/98161300>)
-

11.6.1 Kommunikationsdienste der PROFIBUS-CMs

Die PROFIBUS-CMs nutzen das PROFIBUS DP-V1-Protokoll.

Kommunikationstypen für DP-V1

Für DP-V1 stehen die folgenden Kommunikationstypen zur Verfügung:

- Zyklische Kommunikation (CM 1242-5 und CM 1243-5)
Beide PROFIBUS-Module unterstützen die zyklische Kommunikation für die Prozessdatenübertragung zwischen DP-Slave und DP-Master.
Die zyklische Kommunikation wird vom Betriebssystem der CPU bearbeitet. Hierfür sind keine Softwarebausteine erforderlich. Die E/A-Daten werden direkt aus dem/in das Prozessabbild der CPU gelesen bzw. geschrieben.
- Azyklische Kommunikation (nur CM 1243-5)
Der DP-Master unterstützt auch die azyklische Kommunikation mit Softwarebausteinen:
 - Die Anweisung "RALRM" ist für die Alarmbearbeitung verfügbar.
 - Die Anweisungen "RDREC" und "WRREC" sind für die Übertragung von Konfigurations- und Diagnosedaten verfügbar.

Vom CM 1243-5 nicht unterstützte Funktionen: SYNC/FREEZE und Get_Master_Diag

Weitere Kommunikationsdienste des CM 1243-5

Der CM 1243-5 DP-Master unterstützt die folgenden zusätzlichen Kommunikationsdienste:

- S7-Kommunikation
 - PUT/GET-Dienste
Der DP-Master fungiert als Client und Server für Abfragen anderer S7-Steuerungen oder PCs über PROFIBUS.
 - PG/OP-Kommunikation
Mit den PG-Funktionen können Konfigurationsdaten und Anwenderprogramme von einem PG heruntergeladen und Diagnosedaten in ein PG geladen werden.
Mögliche Kommunikationspartner für die OP-Kommunikation sind HMI-Panels, SIMATIC Panel-PCs mit WinCC flexible oder SCADA-Systeme, die die S7-Kommunikation unterstützen.

11.6.2 Verweis auf die Benutzerhandbücher für PROFIBUS-CMs

Weitere Informationen

Ausführliche Informationen über die PROFIBUS-CMs finden Sie in den Handbüchern zu den jeweiligen Geräten. Diese stehen im Internet auf den Seiten des Kundensupports von Siemens Industrial Automation unter den folgenden IDs zur Verfügung:

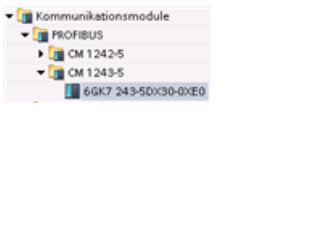
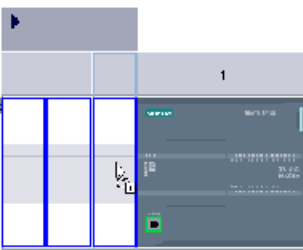
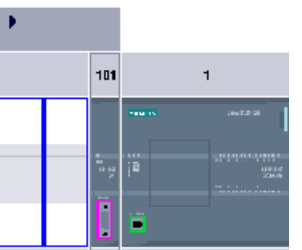
- CM 1242-5 (<https://support.industry.siemens.com/cs/ww/de/ps/15667>)
- CM 1243-5 (<https://support.industry.siemens.com/cs/ww/de/ps/15669>)

11.6.3 DP-Master und -Slave konfigurieren

11.6.3.1 CM 1243-5 (DP-Master) und DP-Slave hinzufügen

Verwenden Sie im Portal "Geräte & Netze" den Hardwarekatalog, um PROFIBUS-Module zur CPU hinzuzufügen. Diese Module werden an der linken Seite der CPU angeschlossen. Um ein Modul in die Hardwarekonfiguration einzufügen, selektieren Sie das Modul im Hardwarekatalog und doppelklicken, oder Sie ziehen es in den markierten Steckplatz.

Tabelle 11-62 PROFIBUS CM 1243-5 (DP-Master) zur Gerätekonfiguration hinzufügen



Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-5 (DP-Master)			

Im Hardwarekatalog können Sie auch die DP-Slaves hinzufügen. Um beispielsweise einen ET200 SP DP-Slave hinzuzufügen, erweitern Sie im Hardwarekatalog die folgenden Ordner:

- Dezentrale E/A
- ET 200SP
- Schnittstellenmodule
- PROFIBUS

Wählen Sie dann "6ES7 155-6BU00-0CNO" (IM155-6 DP HF) in der Liste der Bestellnummern aus und fügen Sie den ET200SP DP-Slave wie in der folgenden Abbildung gezeigt ein.

Tabelle 11-63 ET200SP DP-Slave zur Gerätekonfiguration hinzufügen

DP-Slave einfügen	Ergebnis
	

11.6.3.2 Konfigurieren der logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten

Nachdem Sie das Modul CM 1243-5 (DP-Master) konfiguriert haben, können Sie nun Ihre Netzwerkverbindungen konfigurieren.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Um eine PROFIBUS-Verbindung anzulegen, aktivieren Sie das violettfarbene Feld (PROFIBUS) am

ersten Gerät. Ziehen Sie mit der Maus eine Linie zum PROFIBUS-Feld am zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFIBUS-Verbindung hergestellt.

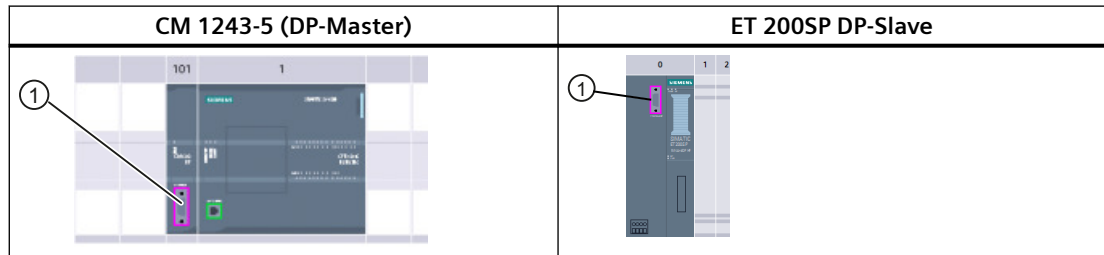
Ausführliche Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 600).

11.6.3.3 PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen

Konfigurieren der PROFIBUS-Schnittstelle

Nachdem Sie die logischen Netzwerkverbindungen zwischen zwei PROFIBUS-Geräten konfiguriert haben, können Sie die Parameter für die PROFIBUS-Schnittstellen konfigurieren. Hierfür klicken Sie auf das violettfarbene PROFIBUS-Feld am Modul CM 1243-5. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die PROFIBUS-Schnittstelle an. Die PROFIBUS-Schnittstelle des DP-Slaves wird auf dieselbe Weise konfiguriert.

Tabelle 11-64 PROFIBUS-Schnittstellen von CM 1243-5 (DP-Master) und ET200SP DP-Slave konfigurieren



① PROFIBUS-Port

PROFIBUS-Adresse zuweisen

In einem PROFIBUS-Netzwerk wird jedem Gerät eine PROFIBUS-Adresse zugewiesen. Diese Adresse kann im Bereich von 0 bis 127 liegen, mit folgenden Ausnahmen:

- Adresse 0: Reserviert für die Netzwerkkonfiguration und/oder an den Bus angeschlossene Programmierwerkzeuge
- Adresse 1: Reserviert von Siemens für den ersten Master
- Adresse 126: Reserviert für Geräte im Werk, die keine Schaltereinstellung haben und über das Netzwerk neu adressiert werden müssen
- Adresse 127: Reserviert für Broadcast-Meldungen an alle Geräte im Netzwerk; darf keinen betriebsfähigen Geräten zugewiesen werden

Deshalb liegen die Adressen, die für betriebsfähige PROFIBUS-Geräte verwendet werden können, im Bereich von 2 bis 125.

Wählen Sie im Eigenschaftsfenster den Eintrag "PROFIBUS-Adresse". STEP 7 zeigt den Konfigurationsdialog für die PROFIBUS-Adresse an, in dem Sie die PROFIBUS-Adresse des Geräts zuweisen.

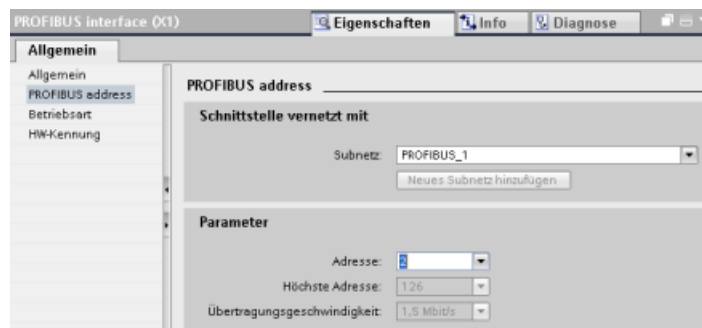


Tabelle 11-65 Parameter für die PROFIBUS-Adresse

Parameter	Beschreibung	
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist. Um ein neues Subnetz anzulegen, klicken Sie auf Schaltfläche "Neues Subnetz hinzufügen". Die Voreinstellung ist "nicht verbunden". Zwei Arten von Verbindungen sind möglich: <ul style="list-style-type: none"> Die Voreinstellung "nicht verbunden" stellt eine lokale Verbindung her. Wenn Ihr Netzwerk über zwei oder mehr Geräte verfügt, ist ein Subnetz erforderlich. 	
Parameter	Adresse	Dem Gerät zugewiesene PROFIBUS-Adresse
	Höchste Adresse	Die höchste PROFIBUS-Adresse basiert auf den aktiven Stationen auf dem PROFIBUS (z. B. DP-Master). Passive DP-Slaves haben unabhängige PROFIBUS-Adressen von 1 bis 125, auch wenn beispielsweise für die höchste PROFIBUS-Adresse 15 eingestellt ist. Die höchste PROFIBUS-Adresse ist für die Token-Weiterleitung (Weiterleitung der Senderechte) von Bedeutung, der Token wird nur an aktive Stationen weitergeleitet. Durch die Angabe der höchsten PROFIBUS-Adresse wird der Bus optimiert.
	Übertragungsgeschwindigkeit	Übertragungsgeschwindigkeit im konfigurierten PROFIBUS-Netzwerk: Die PROFIBUS-Übertragungsgeschwindigkeiten liegen zwischen 9,6 kBit/s und 12 MBit/s. Die Übertragungsgeschwindigkeit ist von den Eigenschaften der verwendeten PROFIBUS-Teilnehmer abhängig. Die Übertragungsgeschwindigkeit darf nicht größer sein als die vom langsamsten Teilnehmer unterstützte Geschwindigkeit. Die Übertragungsgeschwindigkeit wird normalerweise für den Master im PROFIBUS-Netzwerk festgelegt, wobei alle DP-Slaves automatisch die gleiche Übertragungsgeschwindigkeit nutzen (Auto-Baud).

11.6.4 Anweisungen für die dezentrale Peripherie

Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 371).

11.6.5 Diagnoseanweisungen

Weitere Informationen zur Verwendung dieser Anweisungen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseanweisungen" (Seite 420).

11.6.6 Diagnoseereignisse für die dezentrale Peripherie

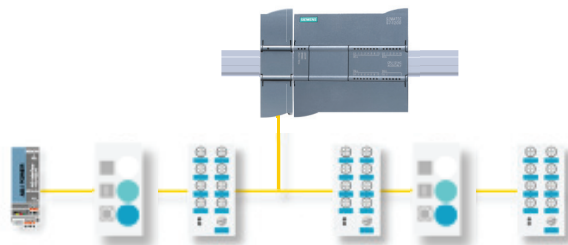
Weitere Informationen zur Verwendung dieser Diagnoseinformationen mit diesen Kommunikationsnetzwerken finden Sie unter "Diagnose (PROFINET oder PROFIBUS): Diagnoseereignisse für die dezentrale Peripherie" (Seite 469).

11.7 ASi

Das S7-1200 AS-i Master CM 1243-2 ermöglicht die Anbindung eines AS-i-Netzwerks an eine S7-1200 CPU.

Die Aktor-/Sensorschnittstelle bzw. AS-i ist ein Netzwerkverbindingssystem für einen Master auf der niedrigsten Stufe in einem Automatisierungssystem. Das CM 1243-2 dient als AS-i-Master im Netzwerk. Mit einem einfachen AS-i-Kabel können Sensoren und Aktoren (AS-i-Slaves) über das CM 1243-2 mit der CPU verbunden werden. Das CM 1243-2 übernimmt die gesamte AS-i-Netzwerkkoordination und leitet Daten und Statusinformationen von Aktoren und Sensoren über die dem CM 1243-2 zugewiesenen E/A-Adressen an die CPU weiter. Je nach Slave-Typ können Sie auf binäre oder analoge Werte zugreifen. Die AS-i-Slaves sind die Eingangs- und Ausgangskanäle des AS-i-Systems und sie sind nur aktiv, wenn sie vom CM 1243-2 aufgerufen werden.

In der folgenden Abbildung ist die S7-1200 ein AS-i-Master, der digitale/analoge Slavegeräte von AS-i-E/A-Modulen steuert.



Um AS-i mit S7-1200 CPUs der Version 4.0 einzusetzen, müssen Sie die AS-i Master CM-Firmware auf V1.1 hochrüsten.

Sie können diese Aktualisierung über den Webserver oder mit einer SIMATIC Memory Card vornehmen.

Hinweis

Bei S7-1200 CPUs der Version 4.0 müssen Sie bei Verwendung des Webservers oder einer SIMATIC Memory Card für das Upgrade der AS-i-Firmware von V1.0 auf V1.1 die AS-i-Firmware im AS-i Master CM 1243-2 wie folgt aktualisieren:

1. Laden Sie das Firmware-Upgrade ins AS-i Master CM 1243-2.
2. Schalten Sie nach beendetem Ladevorgang die S7-1200 CPU aus und wieder ein, um das Firmware-Upgrade im AS-i Master CM 1243-2 vollständig abzuschließen.
3. Wiederholen Sie die Schritte 1 und 2 für jedes zusätzliche AS-i Master CM 1243-2. Der S7-1200 PLC erlaubt maximal drei AS-i Master CM 1243-2.

Hinweis

Es ist empfehlenswert, die AS-i CM-Firmware stets auf dem neuesten Stand zu halten (<http://support.automation.siemens.com/WW/view/de/43416171>). Die aktuelle Firmware finden Sie auf der Siemens-Website "Service & Support".

11.7.1 AS-i-Master und -Slavegeräte konfigurieren




Der AS-i-Master CM 1243-2 ist als Kommunikationsmodul in das S7-1200 Automatisierungssystem integriert.

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-i Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<https://support.industry.siemens.com/cs/ww/de/ps/15750/man>).

11.7.1.1 AS-i-Mastermodul CM 1243-2 und AS-i-Slave hinzufügen

Im Hardwarekatalog können Sie AS-i-Mastermodule CM1243-2 zur CPU hinzufügen. Diese Module werden an der linken Seite der CPU angeschlossen. Es können maximal drei AS-i-Mastermodule CM1243-2 verwendet werden. Um ein Modul in die Hardwarekonfiguration einzufügen, selektieren Sie das Modul im Hardwarekatalog und doppelklicken, oder Sie ziehen es in den markierten Steckplatz.

Tabelle 11-66 AS-i-Mastermodul CM1243-2 zur Gerätekonfiguration hinzufügen

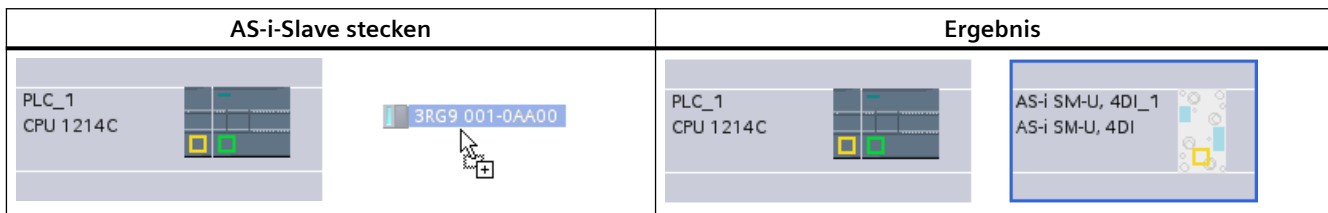
Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-2 AS-i-Master			

Im Hardwarekatalog können Sie auch die AS-i-Slaves hinzufügen. Um beispielsweise einen Slave mit den Merkmalen "E/A-Modul, kompakt, digital, Eingang" hinzuzufügen, erweitern Sie im Hardwarekatalog die folgenden Behälter:

- Feldgeräte
- AS-Schnittstellen-Slaves

Wählen Sie anschließend in der Liste der Bestellnummern "3RG9 001-0AA00" (AS-i SM-U, 4DI) aus und fügen Sie den Slave mit den Merkmalen "E/A-Modul, kompakt, digital, Eingang" wie in der folgenden Abbildung gezeigt ein.

Tabelle 11-67 AS-i-Slave zur Gerätekonfiguration hinzufügen



11.7.1.2 Logische Netzwerkverbindungen zwischen zwei AS-i-Geräten konfigurieren

Nachdem Sie den AS-i-Master CM1243-2 konfiguriert haben, können Sie nun Ihre Netzwerkverbindungen konfigurieren.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Um eine AS-i-Verbindung anzulegen, aktivieren Sie das gelbe Feld (AS-i) am ersten Gerät. Ziehen Sie mit der Maus eine Linie zum AS-i-Feld am zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre AS-i-Verbindung hergestellt.

Weitere Informationen finden Sie unter "Gerätekonfiguration: Erstellen einer Netzwerkverbindung" (Seite 600).

11.7.1.3 Eigenschaften des AS-i-Masters CM1243-2 konfigurieren

Um Parameter für die AS-i-Schnittstelle zu konfigurieren, klicken Sie auf das gelbe AS-i-Feld am AS-i-Mastermodul CM1243-2. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die AS-i-Schnittstelle an.

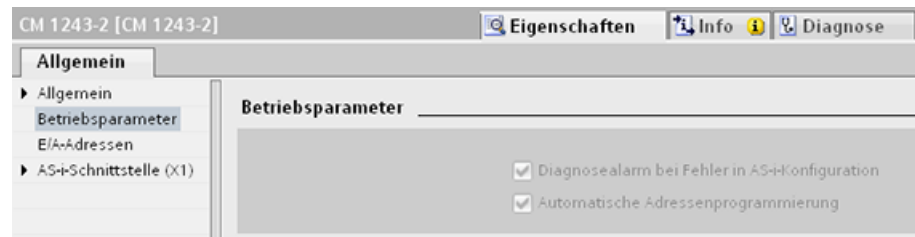
Im STEP 7-Inspektorfenster können Sie allgemeine Informationen, Adressen und Betriebsparameter anzeigen, konfigurieren und ändern:

Tabelle 11-68 Eigenschaften des AS-i-Mastermoduls CM1243-2

Eigenschaft	Beschreibung
Allgemein	Name des AS-i-Masters CM 1243-2
Betriebsparameter	Parameter für die Antwort des AS-i-Masters
E/A-Adressen	Adressbereich für die Slave-E/A-Adressen
AS-i-Schnittstelle (X1)	Zugewiesenes AS-i-Netzwerk

Hinweis

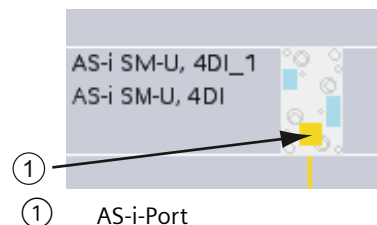
"Diagnosealarm für Fehler in der AS-i-Konfiguration" und "Automatische Adressenprogrammierung" sind stets aktiv und deshalb grau dargestellt.



11.7.1.4 Einem AS-i-Slave eine AS-i-Adresse zuweisen

AS-i-Slaveschnittstelle konfigurieren

Um Parameter für die AS-i-Schnittstelle zu konfigurieren, klicken Sie auf das gelbe AS-i-Feld am AS-i-Slave. Daraufhin zeigt das Register "Eigenschaften" im Inspektorfenster die AS-i-Schnittstelle an.



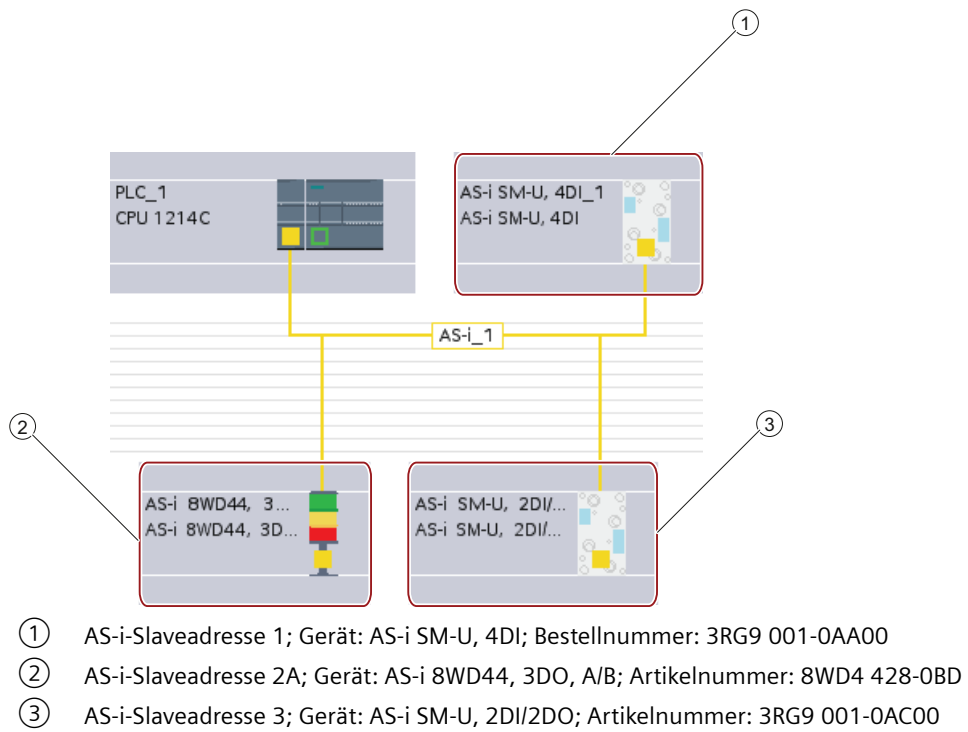
AS-i-Slaveadresse zuweisen

In einem AS-i-Netzwerk wird jedem Gerät eine AS-i-Slaveadresse zugewiesen. Diese Adresse kann im Bereich von 0 bis 31 liegen. Die Adresse 0 ist jedoch ausschließlich für neue Slavegeräte reserviert. Die Slaveadressen sind 1(A oder B) bis 31(A oder B) für insgesamt maximal 62 Slavegeräte.

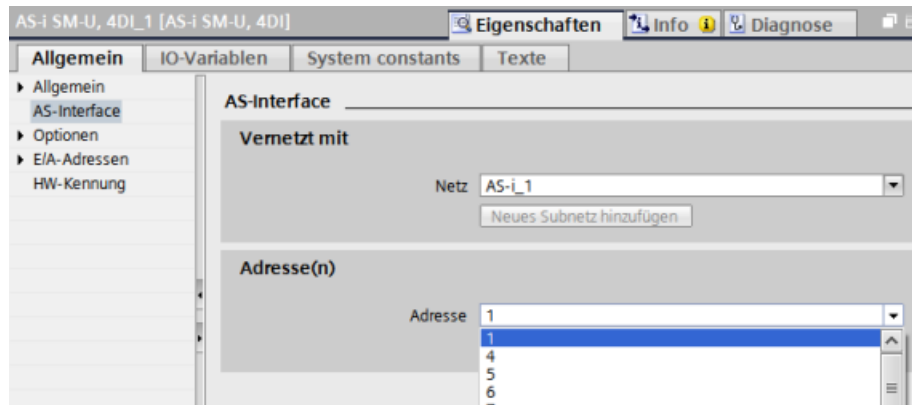
"Standard"-AS-i-Geräte verwenden die gesamte Adresse und haben eine Zahlenadresse ohne den Bezeichner A oder B. AS-i-Geräte vom Typ "A/B-Knoten" verwenden jeweils den A- oder B-Anteil der Adresse, so dass jede der 31 Adressen zweimal verwendet werden kann. Der Adressbereich umfasst 1A bis 31A plus 1B bis 31B.

Jede Adresse im Bereich von 1 bis 31 kann einem AS-i-Slavegerät zugewiesen werden. D. h. mit anderen Worten: Es ist nicht wichtig, ob die Slaves mit Adresse 21 beginnen oder ob der erste Slave tatsächlich die Adresse 1 erhält.

Im folgenden Beispiel wurden drei AS-i-Geräten die Adressen "1" (Standardgerät), "2A" (Gerät vom Typ A/B-Knoten) und "3" (Standardgerät) zugewiesen:



Geben Sie hier die AS-i-Slaveadresse ein:



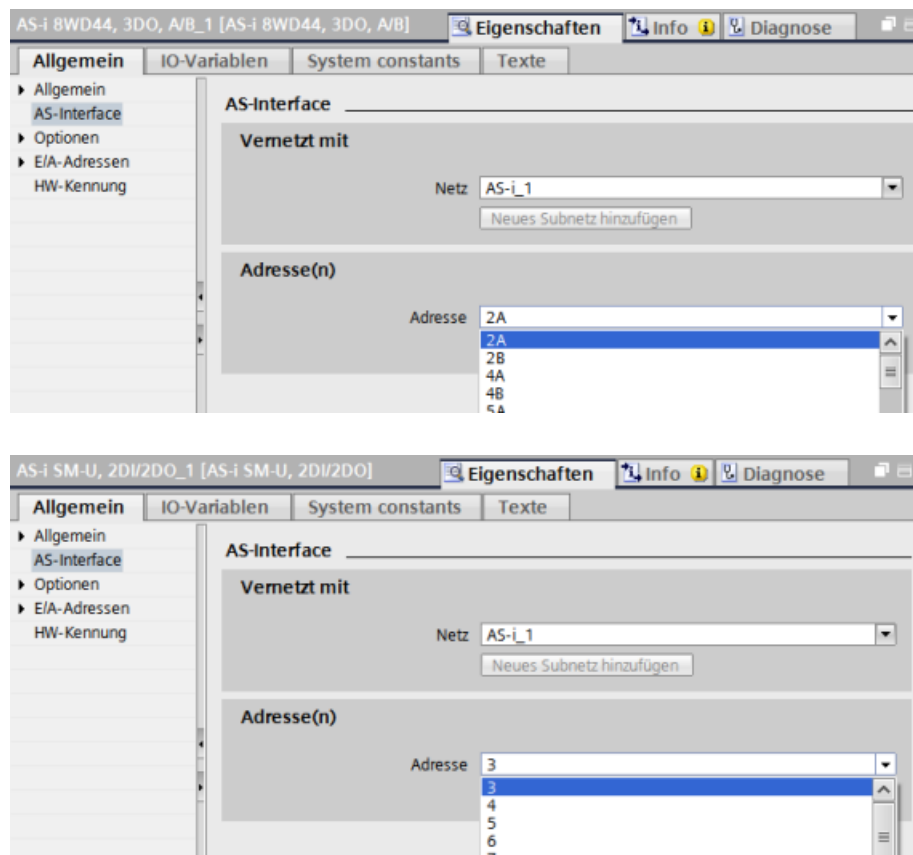


Tabelle 11-69 Parameter für die AS-i-Schnittstelle

Parameter	Beschreibung
Netzwerk	Name des Netzwerks, mit dem das Gerät verbunden ist.
Adresse(n)	Zugewiesene AS-i-Adresse für das Slavegerät im Bereich von 1(A oder B) bis 31(A oder B) für insgesamt bis zu 62 Slavegeräte

11.7.2 Datenaustausch zwischen dem Anwenderprogramm und AS-i-Slaves

11.7.2.1 STEP 7 Basic konfigurieren

Der AS-i-Master reserviert einen 62 Byte großen Datenbereich im E/A-Bereich der CPU. Der Zugriff auf die Digitaldaten erfolgt hier in Byte. Für jeden Slave gibt es ein Byte Eingangs- und ein Byte Ausgangsdaten.

Die Zuweisung der AS-i-Verbindungen der digitalen AS-i-Slaves zu den Datenbits des zugewiesenen Bytes wird im Inspektorfenster des AS-i-Mastermoduls CM 1243-2 angezeigt.

E-Adresse	A-Adresse	AS-i Adresse	HWID
		0	335
2	2	1A	336
33	33	1B	337
3	3	2A	338
34	34	2B	339
4	4	3A	340
35	35	3B	341
5	5	4A	342
36	36	4B	343
6	6	5A	344
37	37	5B	345
7	7	6A	346

Sie können auf die Daten der AS-i-Slaves im Anwenderprogramm zugreifen, indem Sie die angezeigten E/A-Adressen mit den entsprechenden Bitverknüpfungsoperationen (z. B. "UND") oder Bitzuweisungen verwenden.

Hinweis

Wenn Sie die AS-i-Slaves nicht mit STEP 7 konfigurieren, wird die "Systemzuweisung" automatisch aktiviert.

Wenn Sie keine Slaves konfigurieren, müssen Sie das AS-i-Master CM1243-2 über die tatsächliche Buskonfiguration informieren. Hierfür verwenden Sie die Online-Funktion "IST > SOLL".

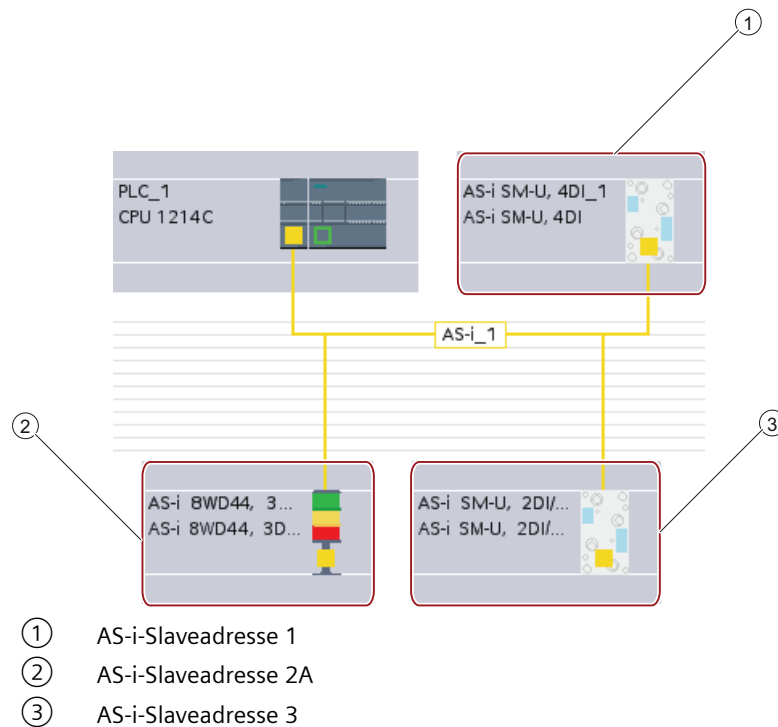
Weitere Informationen

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-I Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<https://support.industry.siemens.com/cs/ww/de/ps/15750/man>).

11.7.2.2 Slaves mit STEP 7 konfigurieren

Digitale AS-i-Werte übertragen

Die CPU greift im zyklischen Betrieb über den AS-i-Master CM1243-2 auf die digitalen Eingänge und Ausgänge der AS-i-Slaves zu. Der Zugriff auf die Daten erfolgt über E/A-Adressen oder mittels Datensatzübertragung.

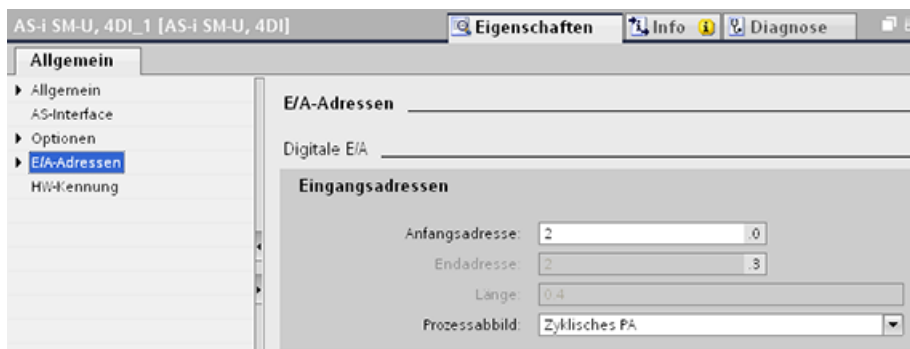


Der Zugriff auf die Digitaldaten erfolgt hier in Bytes (mit anderen Worten: jedem digitalen AS-i-Slave ist ein Byte zugeordnet). Wenn Sie die AS-i-Slaves in STEP 7 konfigurieren, wird für den jeweiligen AS-i-Slave die E/A-Adresse für den Zugriff auf die Daten aus dem Anwenderprogramm im Inspektorenfenster angezeigt.

Dem Digitaleingabemodul (AS-i SM-U, 4DI) im AS-i-Netzwerk oben wurde die Slaveadresse 1 zugewiesen. Wenn Sie auf das Digitaleingabemodul klicken, wird in den "Eigenschaften" des Geräts im Register "AS-Schnittstelle" die Slaveadresse wie im Folgenden angezeigt:

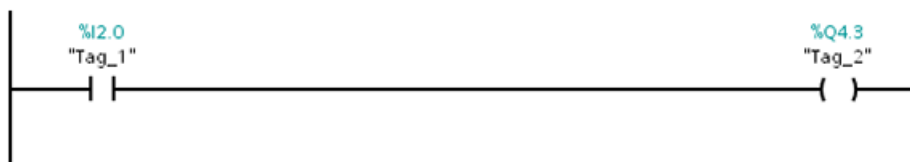


Dem Digitaleingabemodul (AS-i SM-U, 4DI) im AS-i-Netzwerk oben wurde die E/A-Adresse 2 zugewiesen. Wenn Sie auf das Digitaleingabemodul klicken, wird in den "Eigenschaften" des Geräts im Register "E/A-Adressen" die E/A-Adresse wie im Folgenden angezeigt:



Sie können auf die Daten der AS-i-Slaves im Anwenderprogramm zugreifen, indem Sie die deren E/A-Adressen mit den entsprechenden Bitverknüpfungsoperationen (z. B. "UND") oder Bitzuweisungen verwenden. Das folgende einfache Programm zeigt die Funktionsweise der Zuweisung:

In diesem Programm wird Eingang 2.0 abgefragt. Im AS-i-System gehört dieser Eingang zu Slave 1 (Eingangsbyte 2, Bit 0). Ausgang 4.3, der dann gesetzt wird, entspricht dem AS-i-Slave 3 (Ausgangsbyte 4, Bit 3).



Analoge AS-i-Werte übertragen

Wenn Sie einen AS-i-Slave in STEP 7 als analogen Slave konfiguriert haben, können Sie über das Prozessabbild der CPU auf die analogen Daten dieses AS-i-Slaves zugreifen.

Wenn Sie den analogen Slave nicht in STEP 7 konfiguriert haben, können Sie nur über die azyklischen Funktionen (Datenschnittstelle) auf die Daten des AS-i-Slaves zugreifen. Im Anwenderprogramm der CPU werden AS-i-Aufrufe über die Anweisungen für die dezentrale

Peripherie, nämlich RDREC (Datensatz lesen) und WRREC (Datensatz schreiben), gelesen und geschrieben.

Hinweis

Eine über STEP 7 angegebene und in die S7-Station geladene Konfiguration der AS-i-Slaves wird beim Anlauf der S7-Station von der CPU in den AS-i-Master CM1243-2 geladen. Eine ggf. bereits vorhandene Konfiguration, die über die Online-Funktion der "Systemzuweisung" (Seite 793) ("IST -> SOLL") bestimmt wurde, wird überschrieben.

Weitere Informationen

Ausführliche Informationen zum AS-i-Master CM 1243-2 finden Sie im Handbuch "AS-I Master CM 1243-2 und AS-i-Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200" (<http://support.automation.siemens.com/WW/view/en/50414115/133300>).

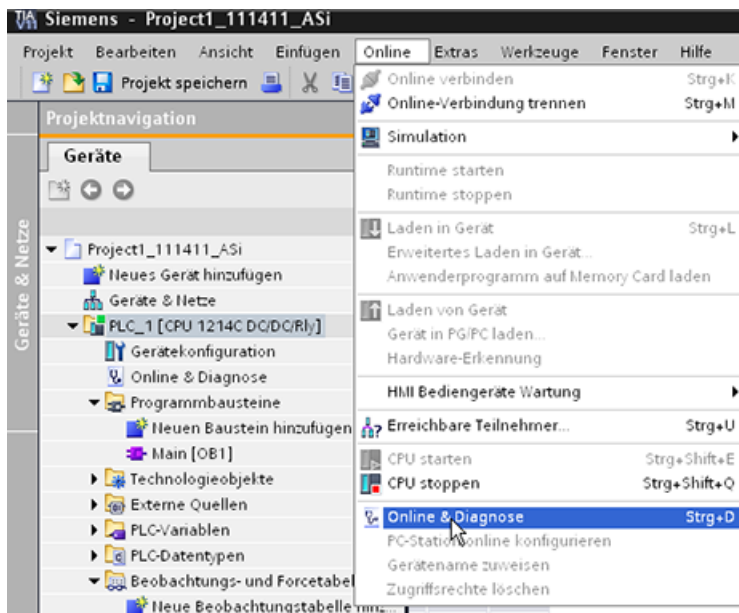
11.7.3 Anweisungen für die dezentrale Peripherie

Weitere Informationen zur Verwendung der Anweisungen für die dezentrale Peripherie mit diesen Kommunikationsnetzwerken finden Sie unter "Dezentrale Peripherie (PROFINET, PROFIBUS oder AS-i)" (Seite 371).

11.7.4 Mit AS-i-Online-Werkzeugen arbeiten**Betriebsarten der AS-i online ändern**

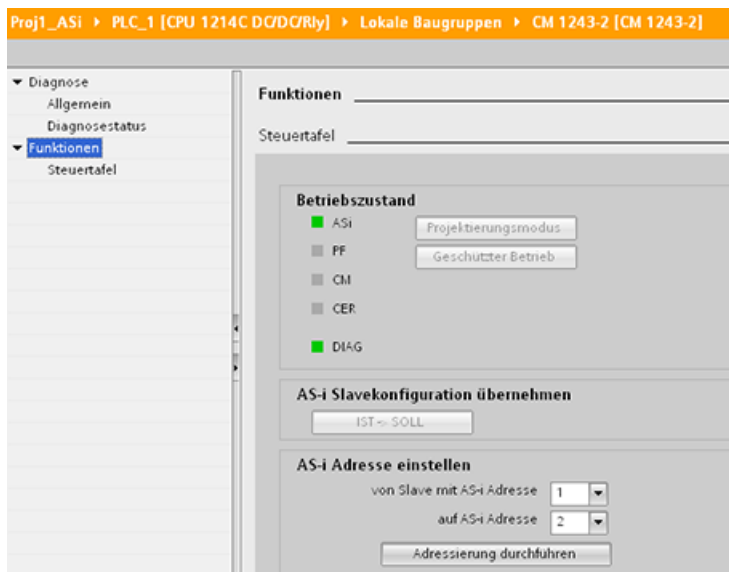
Sie müssen online gehen, um die Betriebsarten der AS-i anzuzeigen und zu ändern.

Um online zu gehen, müssen Sie zunächst das AS-i-Mastermodul CM1243-2 auswählen und die "Gerätekonfiguration" aufrufen. Klicken Sie dann in der Symbolleiste auf die Schaltfläche "Online gehen". Wählen Sie dann im Menü "Online" den Befehl "Online und Diagnose".



Es gibt zwei AS-i-Betriebsarten:

- Schutzmodus:
 - Das AS-i-Slavegerät und die E/A-Adressen der CPU können nicht geändert werden.
 - Die grüne LED "CM" ist ausgeschaltet.
- Konfigurationsmodus:
 - Sie können in Ihrem AS-i-Slavegerät und an den E/A-Adressen der CPU erforderliche Änderungen vornehmen.
 - Die grüne LED "CM" ist eingeschaltet.



Die AS-i-Slaveadresse können Sie im Feld "AS-i-Adresse festlegen" ändern. Ein neuer Slave, dem noch keine Adresse zugewiesen wurde, hat immer die Adresse 0. Er wird vom Master als neuer

Slave ohne Adresszuweisung erkannt und wird erst nach Zuweisung einer Adresse in die normale Kommunikation einbezogen.

Konfigurationsfehler

Wenn die gelbe LED "CER" leuchtet, liegt ein Fehler in der Gerätekonfiguration des AS-i-Slaves vor. Wählen Sie die Schaltfläche "IST > SOLL", um die Slave-Gerätekonfiguration des AS-i-Mastermoduls CM1243-2 durch die Slave-Gerätekonfiguration des AS-i-Feldnetzwerks zu überschreiben.

11.8 S7-Kommunikation

11.8.1 GET und PUT (Aus remoter CPU auslesen und schreiben)

Mittels der Anweisungen GET und PUT können Sie mit S7-CPU's über PROFINET- und PROFIBUS-Verbindungen kommunizieren. Dies ist nur möglich, wenn die Funktion "Zugriff über PUT/GET-Kommunikation erlauben" für die Partner-CPU in der Eigenschaft "Schutz" der lokalen CPU-Eigenschaften aktiviert ist:

- Zugriff auf Daten in einer dezentralen CPU: Eine S7-1200 CPU kann im Eingabefeld ADDR_x nur absolute Adressen verwenden, um Variablen dezentraler CPUs (S7-200/300/400/1200) anzusprechen.
- Zugriff auf Daten in einem Standard-DB: Eine S7-1200 CPU kann im Eingabefeld ADDR_x nur absolute Adressen verwenden, um DB-Variablen in einem Standard-DB einer dezentralen S7-CPU anzusprechen.
- Zugriff auf Daten in einem optimierten DB: Eine S7-1200 CPU kann nicht auf DB-Variablen in einem optimierten DB einer dezentralen S7-1200 CPU zugreifen.
- Zugriff auf Daten in einer lokalen CPU: Eine S7-1200 CPU kann entweder absolute Adressen oder symbolische Adressen als Eingabe im Eingabefeld RD_x oder SD_x der Anweisung GET oder PUT verwenden.

Hinweis

GET/PUT-Funktion im Programm einer V4.0-CPU nicht automatisch aktiviert

Die GET/PUT-Funktion des Programms einer V3.0 CPU ist in einer V4.0-CPU automatisch aktiviert.

Die GET/PUT-Funktion des Programms einer V4.0 CPU jedoch ist in einer V4.0-CPU nicht automatisch aktiviert. Sie müssen den GET/PUT-Zugriff (Seite 163) in der Gerätekonfiguration der CPU im Register "Eigenschaften" des Inspektorfensters unter der Eigenschaft "Schutz" aktivieren.

Tabelle 11-70 Anweisungen GET und PUT

KOP/FUP	SCL	Beschreibung
	<pre>"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_]);</pre>	<p>Mit der Anweisung GET lesen Sie Daten aus einer entfernten S7-CPU aus. Die entfernte CPU kann sich dabei im Betriebszustand RUN oder STOP befinden.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>
	<pre>"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_]);</pre>	<p>Mit der Anweisung PUT schreiben Sie Daten in eine entfernte S7-CPU. Die entfernte CPU kann sich dabei im Betriebszustand RUN oder STOP befinden.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>

Tabelle 11-71 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	Input	Bool
ID	Input	CONN_PRG (Word)
NDR (GET)	Output	Bool
DONE (PUT)	Output	Bool

Parameter und Datentyp		Datentyp	Beschreibung
ERROR STATUS	Output Output	Bool Word	<ul style="list-style-type: none"> • ERROR=0 STATUS-Wert: <ul style="list-style-type: none"> – 0000H: Weder Warnung noch Fehler. – <> 0000H: Warnung, STATUS liefert ausführliche Informationen. • ERROR=1 Fehler. STATUS liefert ausführliche Informationen über die Natur des Fehlers.
ADDR_1	InOut	Entfernt	Pointer auf die Speicherbereiche in der entfernten CPU, die die zu lesenden (GET) oder zu sendenden (PUT) Daten speichert.
ADDR_2	InOut	Entfernt	
ADDR_3	InOut	Entfernt	
ADDR_4	InOut	Entfernt	
RD_1 (GET) SD_1 (PUT)	InOut	Variante	Pointer auf die Speicherbereiche in der lokalen CPU, die die zu lesenden (GET) oder zu sendenden (PUT) Daten speichert.
RD_2 (GET) SD_2 (PUT)	InOut	Variante	Zulässige Datentypen: Bool (nur ein Bit zulässig), Byte, Char, Word, Int, DWord, DInt oder Real.
RD_3 (GET) SD_3 (PUT)	InOut	Variante	Hinweis: Wenn der Pointer auf einen DB zugreift, müssen Sie die absolute Adresse angeben, z. B.:
RD_4 (GET) SD_4 (PUT)	InOut	Variante	P# DB10.DBX5.0 Byte 10 In diesem Fall stellt der Wert 10 die Anzahl von Bytes für die Anweisung GET oder PUT dar.

Sie müssen sicherstellen, dass Länge (Anzahl der Bytes) und Datentypen der Parameter ADDR_x (entfernte CPU) und RD_x oder SD_x (lokale CPU) übereinstimmen. Die Zahl nach der Kennung "Byte" ist die Anzahl von Bytes, die vom Parameter ADDR_x, RD_x oder SD_x angegeben wird.

Hinweis

Die Gesamtzahl der von einer Anweisung GET empfangenen Bytes bzw. die Gesamtzahl der von einer Anweisung PUT gesendeten Bytes ist begrenzt. Die Begrenzung richtet sich danach, wie viele der vier möglichen Adress- und Speicherbereiche Sie verwenden:

- Wenn Sie nur ADDR_1 und RD_1/SD_1 verwenden, kann eine Anweisung GET 222 Bytes abrufen und eine Anweisung PUT 212 Bytes senden.
- Wenn Sie ADDR_1, RD_1/SD_1, ADDR_2 und RD_2/SD_2 verwenden, kann eine Anweisung GET insgesamt 218 Bytes abrufen und eine Anweisung PUT insgesamt 196 Bytes senden.
- Wenn Sie ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3 und RD_3/SD_3 verwenden, kann eine Anweisung GET insgesamt 214 Bytes abrufen und eine Anweisung PUT insgesamt 180 Bytes senden.
- Wenn Sie ADDR_1, RD_1/SD_1, ADDR_2, RD_2/SD_2, ADDR_3, RD_3/SD_3, ADDR_4, RD_4/SD_4 verwenden, kann eine Anweisung GET insgesamt 210 Bytes abrufen und eine Anweisung PUT insgesamt 164 Bytes senden.

Die Summe der Byteanzahlen von jeder Ihrer Adress- und Speicherbereichsparameter muss kleiner oder gleich den festgelegten Grenzwerten sein. Wenn Sie diese Grenzwerte überschreiten, gibt die Anweisung GET oder PUT einen Fehler aus.

Bei der steigenden Flanke am Parameter REQ lädt die Leseanweisung (GET) oder die Schreibanweisung (PUT) die Parameter ID, ADDR_1 und RD_1 (GET) oder SD_1 (PUT).

- Bei GET: Die entfernte CPU gibt die angeforderten Daten an die Empfangsbereiche aus (RD_x). Sie beginnt damit im nächsten Zyklus. Wenn die Leseanweisung fehlerfrei beendet wurde, wird der Parameter NDR auf 1 gesetzt. Eine neue Anweisung kann erst gestartet werden, wenn die vorherige Anweisung beendet ist.
- Bei PUT: Die lokale CPU beginnt, die Daten (SD_x) an die Adresse (ADDR_x) im Speicher der entfernten CPU zu senden. Wenn die Schreibanweisung fehlerfrei beendet wurde, gibt die entfernte CPU eine Ausführungsquittierung aus. Der Parameter DONE der Anweisung PUT wird dann auf 1 gesetzt. Eine neue Schreibanweisung kann erst gestartet werden, wenn die vorherige Anweisung beendet ist.

Hinweis

Um Datenkonsistenz sicherzustellen, prüfen Sie stets, ob die Anweisung beendet wurde (NDR = 1 bei GET bzw. DONE = 1 bei PUT), bevor Sie auf die Daten zugreifen oder eine weitere Lese- oder Schreibanweisung ausführen.

Die Parameter ERROR und STATUS liefern Informationen zum Zustand der Lese- (GET) oder Schreibanweisung (PUT).

Tabelle 11-72 Fehlerinformation

ERROR	STATUS (dezimal)	Beschreibung
0	11	<ul style="list-style-type: none"> • Der neue Auftrag kann nicht wirksam werden, weil der vorherige Auftrag noch nicht beendet ist. • Der Auftrag wird nun in einer Prioritätsklasse mit geringerer Priorität verarbeitet.
0	25	Kommunikation gestartet. Auftrag wird bearbeitet.
1	1	Kommunikationsprobleme, zum Beispiel: <ul style="list-style-type: none"> • Verbindungsbeschreibung nicht geladen (lokal oder entfernt) • Verbindung unterbrochen (Beispiel: Kabel, CPU ausgeschaltet oder CM/CB/CP ist in STOP) • Verbindung zum Partner noch nicht aufgebaut
1	2	Negative Quittierung vom Partnergerät. Die Aufgabe kann nicht ausgeführt werden.
1	4	Fehler in den Sendebereichs-Pointern (RD_x bei GET, SD_x bei PUT) bezüglich Datenlänge oder Datentyp.
1	8	Zugriffsfehler auf der Partner-CPU
1	10	Zugriff auf den lokalen Anwenderspeicher nicht möglich (Beispiel: Zugriffsversuch auf gelöschten DB)
1	12	Als der SFB aufgerufen wurde: <ul style="list-style-type: none"> • Wurde ein Instanz-DB angegeben, der nicht zu GET oder PUT gehört. • Wurde kein Instanz-DB angegeben, sondern ein gemeinsam genutzter DB • Wurde kein Instanz-DB gefunden (Laden eines neuen Instanz-DB).
1	20	<ul style="list-style-type: none"> • Maximale Anzahl paralleler Aufträge/Instanzen überschritten • Die Instanzen wurden bei CPU-RUN überladen Dieser Zustand ist während der ersten Ausführung der Anweisung GET oder PUT möglich.
1	27	Es gibt keine entsprechende Anweisung GET oder PUT in der CPU.

11.8.2 S7-Verbindung erstellen

Verbindungsmechanismen

Um mit Hilfe der Anweisungen PUT/GET mit entfernten Teilnehmern zu kommunizieren, benötigt der Benutzer entsprechende Rechte.

Die Option "Zugriff über PUT/GET-Kommunikation erlauben" ist standardmäßig nicht aktiviert. Der Lese- und Schreibzugriff auf CPU-Daten ist dann nur mit Kommunikationsverbindungen möglich, die für die lokale CPU und für den entfernten Teilnehmer konfiguriert bzw. programmiert werden müssen. Beispielsweise ist der Zugriff mit Hilfe der Anweisungen BSEND/BRCV möglich.

Verbindungen, für die die lokale CPU nur als Server dient (d. h., die Kommunikation mit dem Teilnehmer ist in der lokalen CPU nicht konfiguriert/programmiert), sind während des Betriebs der CPU dann nicht möglich. Dies gilt z. B. für:

- PUT/GET-, FETCH/WRITE- oder FTP-Zugriff über Kommunikationsmodule
- PUT/GET-Zugriff von anderen S7 CPUs
- HMI-Zugriff über PUT/GET-Kommunikation

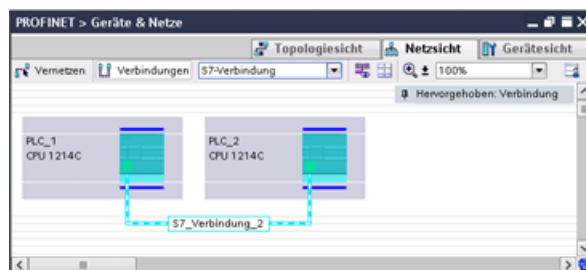
Soll auch der clientseitige Zugriff auf CPU-Daten möglich sein, d. h., sollen die Kommunikationsdienste der CPU nicht beschränkt werden, können Sie den Zugriffsschutz für die S7-1200 CPU (Seite 163) für diese Sicherheitsstufe konfigurieren.

Verbindungsarten

Die ausgewählte Verbindungsart baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine S7-Verbindung). Klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen Sie eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Erstellen einer Netzwerkverbindung" (Seite 600).



Klicken Sie auf die Schaltfläche "Hervorgehoben: Verbindung", um den Konfigurationsdialog "Eigenschaften" der Kommunikationsanweisung aufzurufen.

11.8.3 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

Allgemeine Parameter konfigurieren

Sie legen die Kommunikationsparameter im Dialog "Eigenschaften" der Kommunikationsanweisung fest. Dieser Dialog wird am unteren Ende der Seite angezeigt, wenn Sie einen Teil der Anweisung selektiert haben.

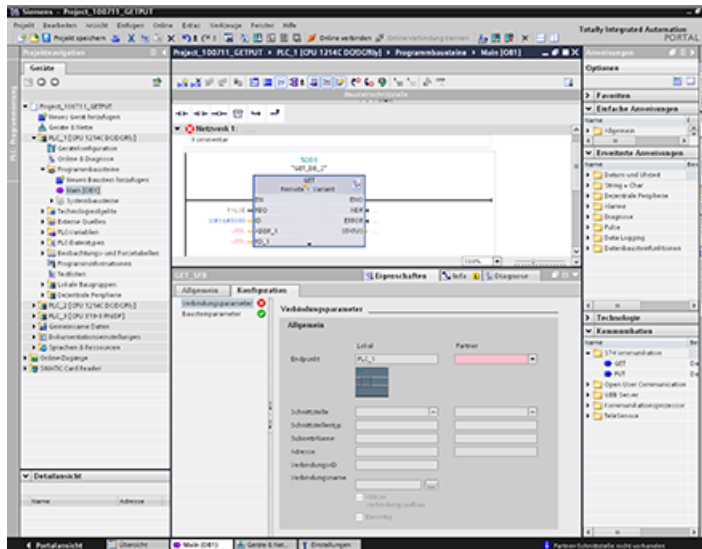
Weitere Informationen finden Sie unter "Gerätekonfiguration: Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren (Seite 600)".

Die zu verwendenden TSAPs oder Ports definieren Sie im Dialog "Verbindungsparameter" unter "Adresdetails". Der TSAP oder Port einer Verbindung in der CPU wird im Feld "Lokaler TSAP" eingegeben. Der TSAP oder Port für die Verbindung in der Partner-CPU wird im Feld "Partner-TSAP" eingegeben.

11.8.4 Zuweisung von Verbindungsparametern für GET/PUT

Die Zuweisung der Verbindungsparameter für die GET/PUT-Anweisungen ist eine Hilfe für den Anwender bei der Konfiguration der S7-Verbindungen für die S7-Kommunikation von CPU zu CPU.

Nach dem Einfügen eines GET- oder PUT-Bausteins zeigt STEP 7 den Zuweisungsdialog für die Verbindungsparameter der GET/PUT-Anweisungen an:



Im Inspektorfenster werden die Eigenschaften der Verbindung angezeigt, wenn Sie einen Teil der Anweisung auswählen. Sie können die Kommunikationsparameter über das Register "Konfiguration" im Dialog "Eigenschaften" der Kommunikationsanweisung konfigurieren.

Hinweis

Die GET/PUT-Funktion ist im Programm einer CPU ab V4.1 nicht automatisch aktiviert

Die GET/PUT-Funktion im Programm einer CPU V3.0 ist in einer CPU ab V4.1 automatisch aktiviert.

Die GET/PUT-Funktion im Programm einer CPU ab V4.1 jedoch ist in einer CPU ab V4.1 nicht automatisch aktiviert. Sie müssen den GET/PUT-Zugriff (Seite 163) in der Gerätekonfiguration der CPU im Register "Eigenschaften" des Inspektorfensters unter der Eigenschaft "Schutz" aktivieren.

11.8.4.1 Verbindungsparameter

Auf der Seite "Verbindungsparameter" können Sie die erforderliche S7-Verbindung und den Parameter "Verbindungs-ID" konfigurieren, auf den vom Parameter "ID" des GET/PUT-Bausteins verwiesen wird. Die Seite enthält Informationen zum lokalen Endpunkt und Sie können hier auch die lokale Schnittstelle festlegen. Ferner können Sie den Partner-Endpunkt definieren.

Auf der Seite "Bausteinparameter" können Sie weitere Bausteinparameter konfigurieren.

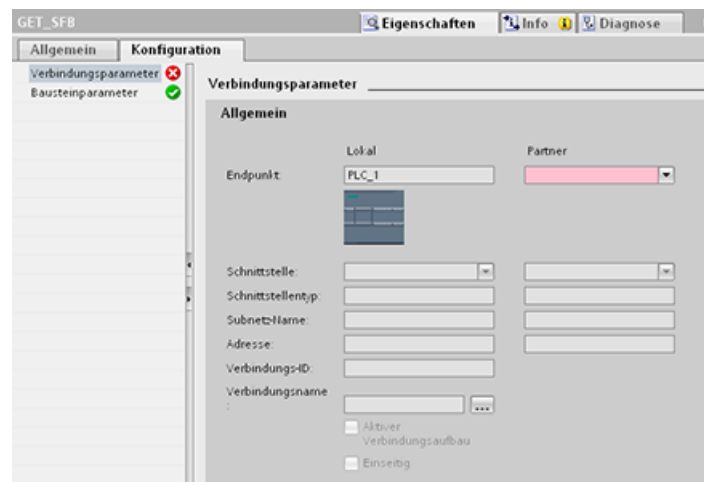


Tabelle 11-73 Verbindungsparameter: Allgemeine Definitionen

Parameter		Definition
Verbindungsparameter: Allgemein	Endpunkt	"Lokaler Endpunkt": Name der lokalen CPU "Partner-Endpunkt": Name der Partner-CPU (dezentral) Hinweis: In der Klappliste "Partner-Endpunkt" zeigt das System alle potenziellen S7-Verbindungspartner des aktuellen Projekts sowie die Option "Nicht spezifiziert" an. Ein nicht spezifizierter Partner ist ein Kommunikationspartner, der sich gegenwärtig nicht im STEP 7-Projekt befindet (z. B. ein Kommunikationspartner eines Fremdherstellers).
	Schnittstelle	Name der Schnittstellen Hinweis: Sie können die Verbindung ändern, indem Sie die lokale und die Partnerschnittstelle ändern.
	Schnittstellentyp	Typ der Schnittstelle
	Subnetzname	Name der Subnetze
	Adresse	Zugewiesene IP-Adressen Hinweis: Für einen "nicht spezifizierten" Kommunikationspartner können Sie die entfernte Adresse eines Fremdgeräts angeben.
	Verbindungs-ID	ID-Nummer: Wird von der Parameterzuweisung der GET/PUT-Verbindung automatisch generiert
	Verbindungsname	Datenspeicheradresse für die lokale CPU und die Partner-CPU: Wird von der Parameterzuweisung der GET/PUT-Verbindung automatisch generiert
	Aktiver Verbindungsaufbau	Kontrollkästchen zur Auswahl der lokalen CPU als aktive Verbindung
	Unidirektional	Kontrollkästchen, um eine unidirektionale oder eine bidirektionale Verbindung anzugeben; schreibgeschützt Hinweis: In einer GET/PUT-Verbindung mit PROFINET können sowohl das lokale Gerät als auch das Partnergerät als Server oder Client fungieren. Dies ermöglicht eine bidirektionale Verbindung, das Kontrollkästchen "Unidirektional" ist dabei nicht aktiviert. Bei einer GET/PUT-Verbindung mit PROFIBUS kann das Partnergerät in einigen Fällen nur als Server fungieren (z. B. bei einer S7-300) und das Kontrollkästchen "Unidirektional" ist aktiviert.

Parameter Verbindungs-ID

Es gibt drei Möglichkeiten, die systemdefinierten Verbindungs-IDs zu ändern:

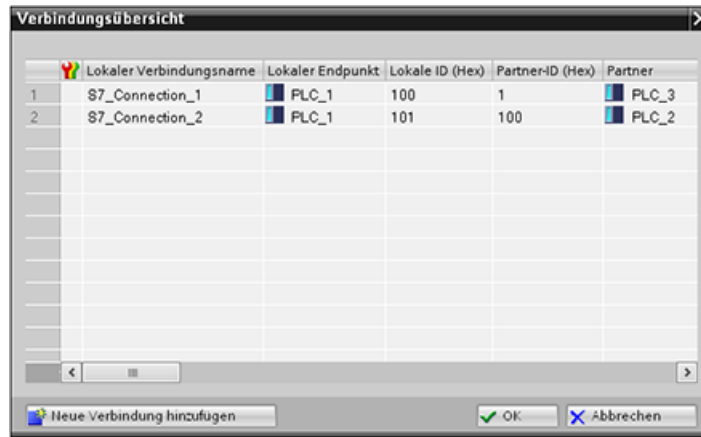
1. Sie können die aktuelle ID direkt im GET/PUT-Baustein ändern. Wenn die neue ID zu einer bereits vorhandenen Verbindung gehört, wird die Verbindung geändert.
2. Sie können die aktuelle ID direkt im GET/PUT-Baustein ändern, doch die neue ID ist noch nicht vorhanden. Das System erstellt eine neue S7-Verbindung.
3. Sie können die aktuelle ID über den Dialog "Verbindungsübersicht" ändern: Ihre Eingabe wird mit dem ID-Parameter des entsprechenden GET/PUT-Bausteins synchronisiert.

Hinweis

Der Parameter "ID" des GET/PUT-Bausteins ist kein Verbindungsname, sondern ein numerischer Ausdruck, der wie im folgenden Beispiel geschrieben wird: W#16#1

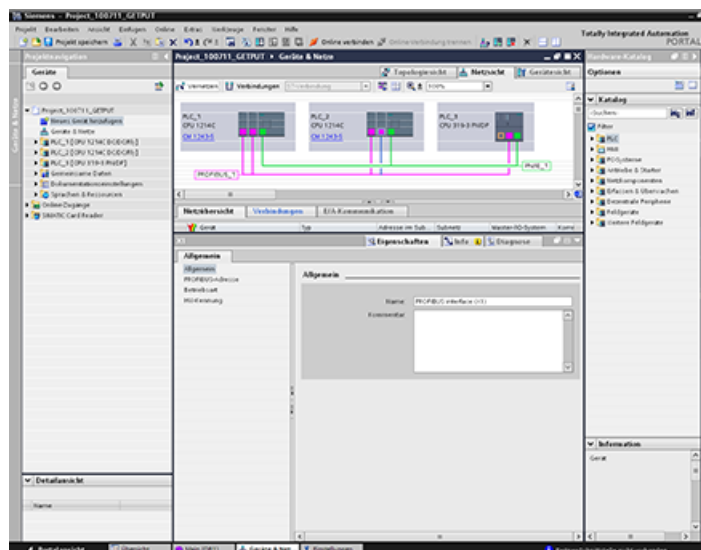
Parameter Verbindungsname

Der Verbindungsname lässt sich über ein besonderes Bedienelement, den Dialog "Verbindungsübersicht", bearbeiten. In diesem Dialog werden alle verfügbaren S7-Verbindungen angezeigt, die als Alternative für die aktuelle GET/PUT-Kommunikation ausgewählt werden können. Sie können in dieser Tabelle eine vollständig neue Verbindung anlegen. Klicken Sie auf die Schaltfläche rechts vom Feld "Verbindungsname", um den Dialog "Verbindungsübersicht" aufzurufen.



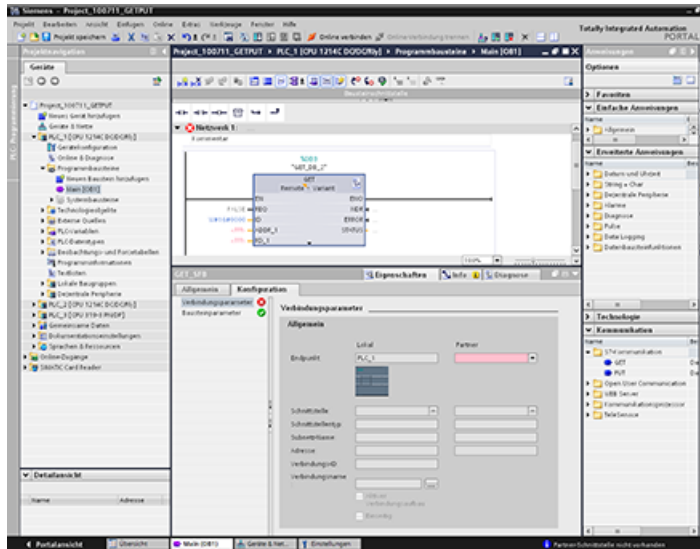
11.8.4.2 S7-Verbindung von CPU zu CPU konfigurieren

Bei einer Konfiguration von PLC_1, PLC_2 und PLC_3, wie in der folgenden Abbildung gezeigt, fügen Sie GET- oder PUT-Bausteine für "PLC_1" ein.



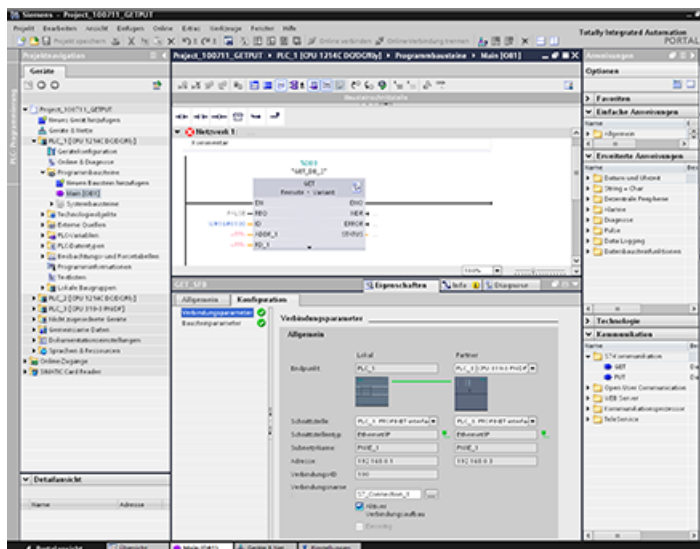
Für die Anweisung GET oder PUT wird im Inspektorfenster automatisch das Register "Eigenschaften" mit den folgenden Menüoptionen angezeigt:

- "Konfiguration"
- "Verbindungsparameter"



PROFINET S7-Verbindung konfigurieren

Wählen Sie für den "Partner-Endpunkt" die Option "PLC_3" aus.



Das System reagiert mit den folgenden Änderungen:

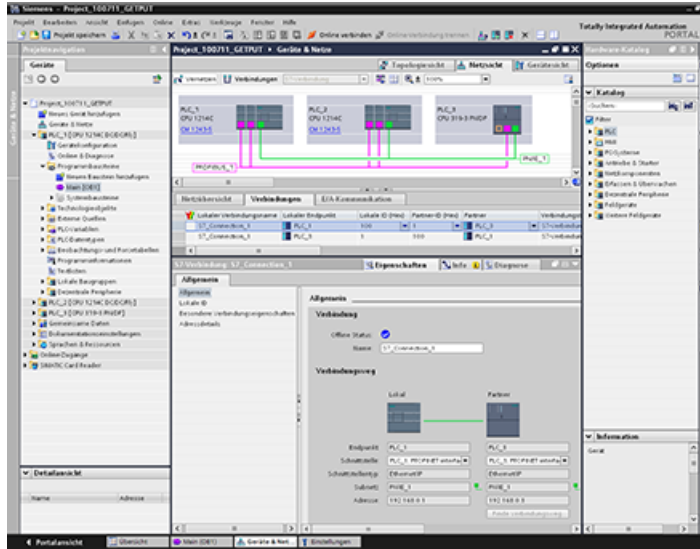
Tabelle 11-74 Verbindungsparameter: Allgemeine Werte

Parameter		Definition
Verbindungsparameter: Allgemein	Endpunkt	"Lokaler Endpunkt" enthält "PLC_1" (schreibgeschützt). Das Feld "Partner-Endpunkt" enthält "PLC_3[CPU319-3PN/DP]": <ul style="list-style-type: none"> Die Farbe wechselt von Rot nach Weiß. Das "Partner"-Gerätebild wird angezeigt. Eine Verbindungslinie erscheint zwischen den PLC_1- und PLC_3-Gerätebildern (grüne Ethernet-Linie).
	Schnittstelle	"Lokale Schnittstelle" enthält "CPU1214C DC/DC/DC, PROFINET-Schnittstelle (R0/S1)". "Partner-Schnittstelle" enthält: "CPU319-3PN/DP, PROFINET-Schnittstelle (R0/S2)".
	Schnittstellentyp	"Lokaler Schnittstellentyp" enthält "Ethernet/IP"; die Bedienung ist schreibgeschützt. "Partner-Schnittstellentyp" enthält "Ethernet/IP"; die Bedienung ist schreibgeschützt. Die Schnittstellentypbilder werden rechts neben dem lokalen und dem Partner-Schnittstellentyp angezeigt (grünes Ethernet-Symbol).
	Subnetzname	"Lokaler Subnetzname" enthält "PN/IE_1"; die Bedienung ist schreibgeschützt. "Partner-Subnetzname" enthält "PN/IE_1"; die Bedienung ist schreibgeschützt.
	Adresse	"Lokale Adresse" enthält die lokale IP-Adresse; die Bedienung ist schreibgeschützt. "Partneradresse" enthält die Partner-IP-Adresse; die Bedienung ist schreibgeschützt.
	Verbindungs-ID	"Verbindungs-ID" enthält "100". Im Hauptprogramm [OB1] des Programmiereditors enthält die "Verbindungs-ID" des GET/PUT-Bausteins ebenfalls den Wert "100".
	Verbindungsname	"Verbindungsname" enthält den Standardverbindungsnamen (Beispiel: "S7_Verbindung_1"); die Bedienung ist aktiviert.
	Aktiver Verbindungsaufbau	Aktiviert, um die lokale CPU als aktive Verbindung auszuwählen.
	Unidirektional	Schreibgeschützt und nicht aktiviert. Hinweis: "PLC_1" (eine S7-1200 CPU 1214CDC/DC/Relais) und "PLC_3" (eine S7-300 CPU 319-3PN/DP) können in einer PROFINET GET/PUT-Verbindung beide als Server und als Client fungieren, so dass eine bidirektionale Verbindung möglich ist.

Das GET/PUT-Symbol in der Eigenschaftsansicht wechselt von Rot nach Grün.

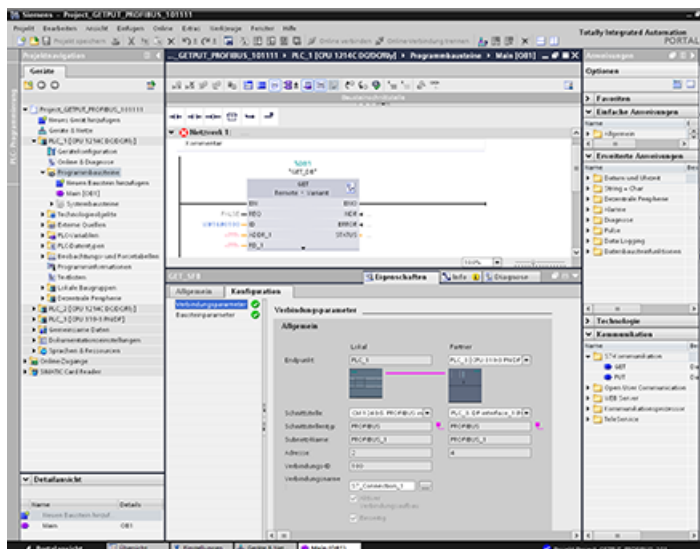
Vollständige PROFINET S7-Verbindung

In der "Netzansicht" wird in der Tabelle "Verbindungen" zwischen "PLC_1" und "PLC_3" eine bidirektionale S7-Verbindung angezeigt.



PROFIBUS S7-Verbindung konfigurieren

Wählen Sie für den "Partner-Endpunkt" die Option "PLC_3" aus.



Das System reagiert mit den folgenden Änderungen:

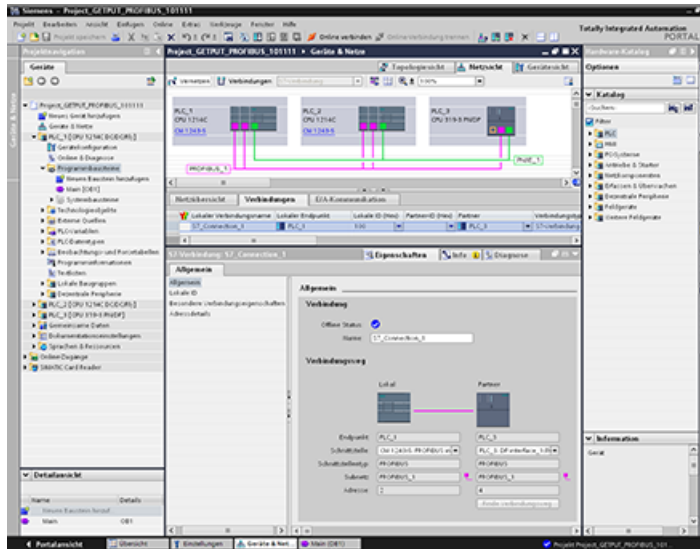
Tabelle 11-75 Verbindungsparameter: Allgemeine Werte

Parameter		Definition
Verbindungsparameter: Allgemein	Endpunkt	"Lokaler Endpunkt" enthält "PLC_1" (schreibgeschützt). Das Feld "Partner-Endpunkt" enthält "PLC_3[CPU319-3PN/DP]": <ul style="list-style-type: none"> Die Farbe wechselt von Rot nach Weiß. Das "Partner"-Gerätebild wird angezeigt. Eine Verbindungslinie erscheint zwischen den PLC_1- und PLC_3-Gerätebildern (violettfarbene PROFIBUS-Linie).
	Schnittstelle	"Lokale Schnittstelle" enthält "CPU1214C DC/DC/DC, PROFIBUS-Schnittstelle (R0/S1)". "Partner-Schnittstelle" enthält: "CPU319-3PN/DP, PROFIBUS-Schnittstelle (R0/S2)".
	Schnittstellentyp	"Lokaler Schnittstellentyp" enthält "PROFIBUS"; die Bedienung ist schreibgeschützt. "Partner-Schnittstellentyp" enthält "PROFIBUS"; die Bedienung ist schreibgeschützt. Die Schnittstellentypbilder werden rechts neben dem lokalen und dem Partner-Schnittstellentyp angezeigt (violettfarbenedes PROFIBUS-Symbol).
	Subnetzname	"Lokaler Subnetzname" enthält "PROFIBUS_1"; die Bedienung ist schreibgeschützt. "Partner-Subnetzname" enthält "PROFIBUS_1"; die Bedienung ist schreibgeschützt.
	Adresse	"Lokale Adresse" enthält die lokale IP-Adresse; die Bedienung ist schreibgeschützt. "Partneradresse" enthält die Partner-IP-Adresse; die Bedienung ist schreibgeschützt.
	Verbindungs-ID	"Verbindungs-ID" enthält "100". Im Hauptprogramm [OB1] des Programmiereditors enthält die "Verbindungs-ID" des GET/PUT-Bausteins ebenfalls den Wert "100".
	Verbindungsname	"Verbindungsname" enthält den Standardverbindungsnamen (Beispiel: "S7_Verbindung_1"); die Bedienung ist aktiviert.
	Aktiver Verbindungsaufbau	Schreibgeschützt und aktiviert, um die lokale CPU als aktive Verbindung auszuwählen.
	Unidirektional	Schreibgeschützt und aktiviert. Hinweis: "PLC_3" (eine S7-300 CPU319-3PN/DP) kann in einer GET/PUT-Verbindung mit PROFIBUS nur als Server fungieren (und nicht auch als Client), so dass nur eine unidirektionale Verbindung möglich ist.

Das GET/PUT-Symbol in der Eigenschaftsansicht wechselt von Rot nach Grün.

Vollständige PROFIBUS S7-Verbindung

In der "Netzansicht" wird in der Tabelle "Verbindungen" zwischen "PLC_1" und "PLC_3" eine unidirektionale S7-Verbindung angezeigt.



11.9 Was tun, wenn kein Zugriff auf die CPU über die IP-Adresse möglich ist

Wenn Sie die CPU nicht über die IP-Adresse erreichen, können Sie eine (temporäre) Not-IP-Adresse für die CPU einstellen. Über die Not-IP-Adresse können Sie die Verbindung mit der CPU wiederherstellen, um eine Gerätekonfiguration mit einer gültigen IP-Adresse zu laden.

Gründe, warum eine Not-IP-Adresse benötigt werden kann

Wenn jemand ein Projekt mit einem der folgenden Probleme heruntergeladen hat, kann Ihre CPU unerreichbar werden:

- Die IP-Adresse der PROFINET-Schnittstelle der CPU ist eine Kopie eines anderen Geräts im Netzwerk.
- Das Subnetz ist für die CPU falsch.
- Die Subnetzmaske macht die CPU unerreichbar.

In diesen Fällen ist die CPU nicht mehr für STEP 7 erreichbar.

Eine Not-IP-Adresse zuweisen

Sie können unter folgenden Umständen eine Not-IP-Adresse zuweisen:

- In der Gerätekonfiguration in STEP 7 ist "IP-Adresse im Projekt einstellen" für das IP-Protokoll ausgewählt.
- Die CPU befindet sich im Betriebszustand STOP.

In diesen Fällen können Sie für die IP-Adresse des Geräts mit einem DCP-Tool eine Not-IP-Adresse einstellen. Beispielsweise verfügt das SIMATIC Automation Tool über einen DCP-Befehl "Set IP address". Eine Not-IP-Adresse können Sie unabhängig von der Schutzstufe (Seite 163) der CPU einstellen. Nach dem Einstellen einer temporären IP-Adresse mit einem DCP-Tool leuchtet die Wartungs-LED an der CPU. Auch der Diagnosepuffer zeigt dann einen Eintrag, der anzeigt, dass eine Notadresse für eine Ethernet-Schnittstelle aktiviert wurde.

IP-Adresse nach Zuweisung einer Not-IP-Adresse wiederherstellen

Der Diagnosepuffer informiert Sie, wenn eine Not-IP-Adresse aktiviert oder deaktiviert wurde. Sie können die Not-IP-Adresse durch Aus- und Einschalten der CPU zurücksetzen.

Nach dem Zuweisen einer Not-IP-Adresse können Sie ein STEP 7-Projekt mit einer gültigen IP-Adresse für die CPU laden. Nach dem Laden des Projekts schalten Sie die CPU aus und wieder ein.

11.10 OPC UA-Server

S7-1200 CPUs unterstützen das OPC UA Micro-Embedded Profile. Außerdem unterstützen S7-1200 CPUs die OPC UA Benutzerauthentifizierung, Kommunikationssicherheit, Subscriptions und Lesen und Schreiben von Programmvariablen. OPC UA-Funktionalität, die nicht im OPC UA Micro Embedded-Profil enthalten oder anderweitig referenziert wird, wird nicht unbedingt unterstützt. Fähigkeiten und Einschränkungen werden in den folgenden Abschnitten beschrieben.

Siehe auch

OPC Foundation (<https://opcfoundation.org/>)

11.10.1 Konfiguration des OPC UA-Servers

Sie können den OPC UA-Server im TIA Portal in den Eigenschaften der CPU-Hardware konfigurieren.

Um den OPC UA-Server zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Gerätekonfiguration das Register "Allgemein" aus.

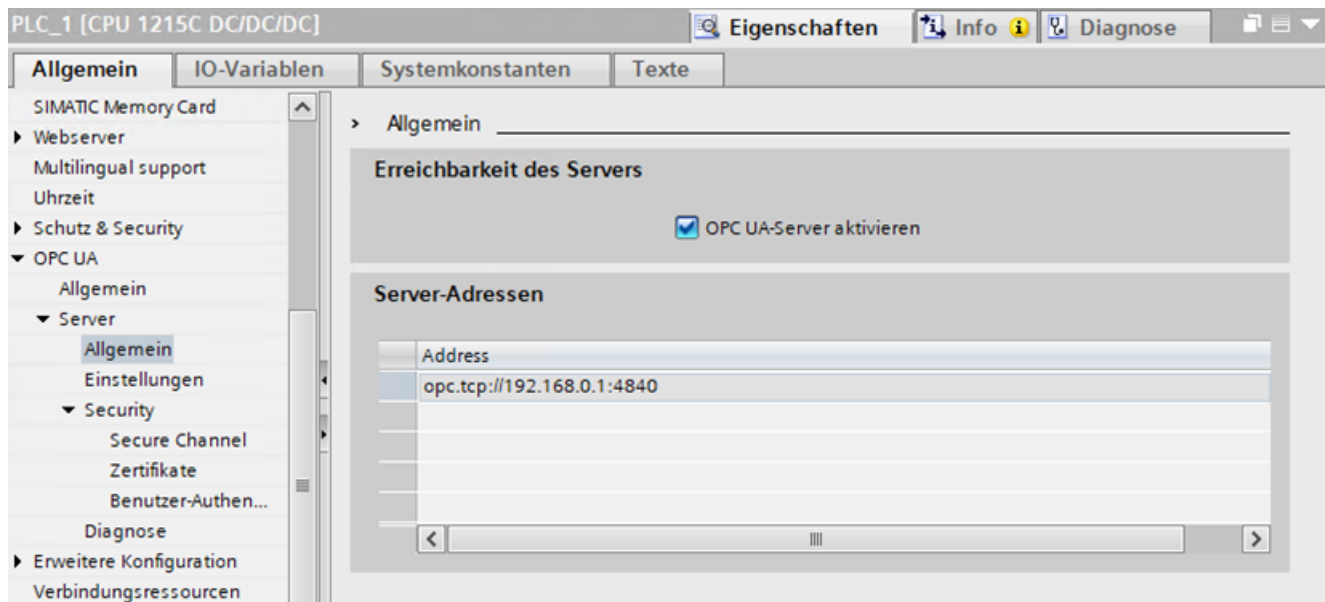
- Wählen Sie im Fenster "Allgemein" "OPC UA" aus.
 - Im Fenster "OPC UA > Server" können Sie Folgendes einstellen:
 - Allgemeine Einstellungen
 - Optionen
 - Security
 - Diagnose

11.10.1.1 Aktivierung des OPC UA-Servers

Der OPC UA-Server ist standardmäßig nicht aktiviert. Sie müssen den OPC UA-Server aktivieren, indem Sie in den Hardware-Eigenschaften der CPU das Kontrollkästchen "OPC UA-Server aktivieren" auswählen.

Um den OPC UA-Server zu aktivieren, gehen Sie folgendermaßen vor:

1. Wählen Sie in der Gerätekonfiguration das Register "Allgemein" aus.
2. Wählen Sie im Fenster "Allgemein" "OPC UA" aus.
3. Aktivieren Sie im Fenster "OPC UA > Server > Allgemein" das Kontrollkästchen "OPC UA-Server aktivieren".



11.10.1.2 Verhalten des OPC UA-Servers im Betrieb

Verhalten des OPC UA-Servers im Betrieb

Der OPC UA-Server der S7-1200 CPU startet, wenn Sie den Server aktivieren und das Projekt in die CPU laden.

Verhalten im Betriebszustand STOP der CPU

Ein aktivierter OPC UA-Server bleibt in Betrieb, selbst wenn die CPU in den Betriebszustand "STOP" geht. Der OPC UA-Server reagiert weiterhin auf Anfragen von OPC UA-Clients.

Serverantwort im Detail:

- Wenn Sie die Werte von PLC-Variablen anfordern, erhalten Sie die Werte, die aktuell waren, bevor die CPU in STOP gegangen ist.
- Wenn Sie Werte in den OPC UA-Server schreiben, akzeptiert der OPC UA-Server diese Werte. Die CPU verarbeitet diese Werte jedoch nicht, weil das Anwenderprogramm im Betriebszustand STOP nicht ausgeführt wird. Ein OPC UA-Client kann trotzdem die Werte lesen, die im Betriebszustand STOP vom OPC UA-Server in die CPU geschrieben wurden.

CPU bei laufendem OPC UA-Server laden

Wenn Sie eine CPU bei laufendem OPC UA-Server laden, müssen Sie möglicherweise abhängig von den geladenen Objekten den Server stoppen und neu starten. In diesem Fall werden aktive Verbindungen unterbrochen und müssen beim Neustart des Servers neu aufgebaut werden.

Die Dauer des Neustarts hängt hauptsächlich von den folgenden Parametern ab:

- Umfang der Datenstruktur
- Anzahl der im OPC UA-Adressraum sichtbaren Variablen
- Einstellung für abwärtskompatible Datentypdefinition entsprechend OPC UA-Spezifikation V1.03 (Typwörterbuch aktiviert)
- Einstellungen für die Kommunikationslast und die Mindestzykluszeit (Seite 153)

Bei Firmware-Versionen der CPU vor V4.5 wurde der OPC UA-Server bei jedem Laden in die CPU gestoppt und dann neu gestartet.

Ab Firmware-Version V4.5 wurde das Verhalten des OPC UA-Servers wie folgt optimiert:

- Werden Objekte im Betriebszustand STOP der CPU geladen, stoppt der OPC UA-Server immer noch und startet dann neu. STEP 7 zeigt in diesem Fall keine Warnung an.
- Werden Objekte im Betriebszustand RUN der CPU geladen, stoppt der OPC UA-Server nur dann, wenn die geladenen Objekte OPC UA-relevant sind oder sein könnten. Der OPC UA-Server startet aufgrund der geänderten OPC UA-Daten nach der Neuinitialisierung neu. Bevor OPC UA-relevante Objekte in die CPU geladen werden und den OPC UA-Server stoppen, zeigt STEP 7 im Vorschau-Dialog zum Laden eine Warnung an. Sie können dann entscheiden, ob ein Server-Neustart für den laufenden Prozess kompatibel ist oder ob Sie das Laden abbrechen. Diese Warnungen werden nur angezeigt, wenn der OPC UA-Server läuft. Wenn der OPC UA-Server nicht aktiviert ist, haben geänderte OPC UA-Daten keinen Einfluss auf den Ladevorgang.

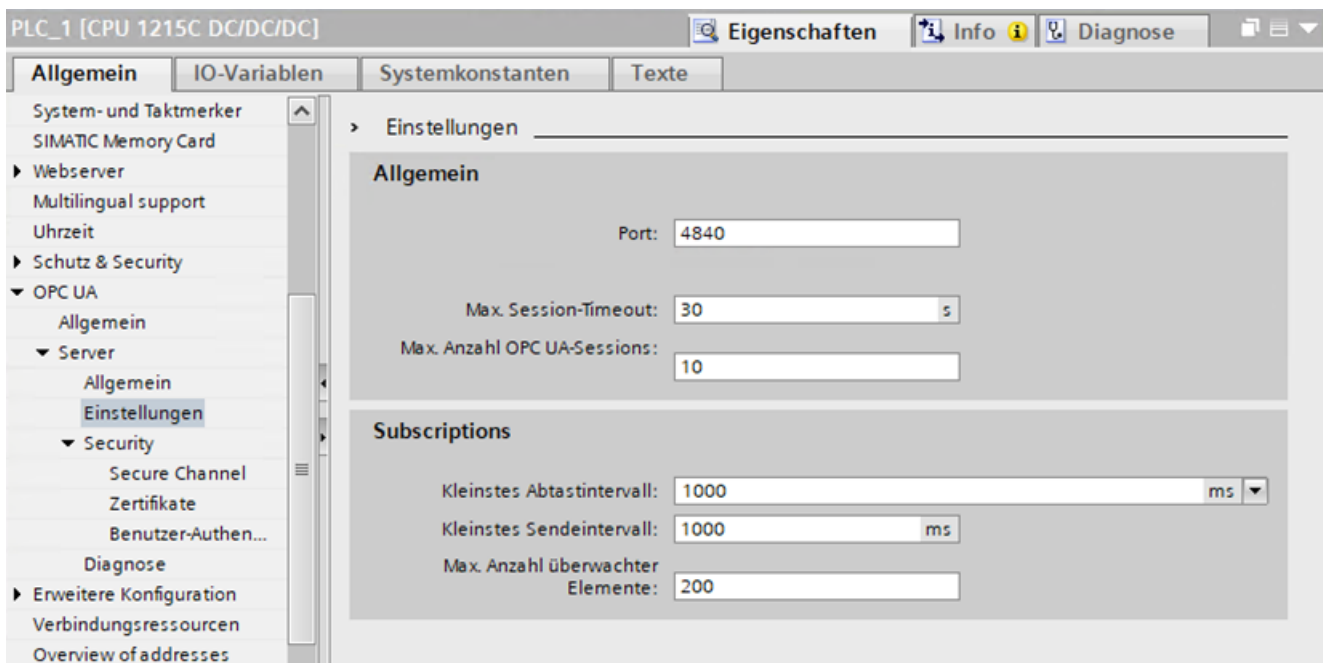
Beispiele

- Sie möchten nur einen weiteren Codebaustein im Programm einfügen.
Weder Datenbausteine noch Eingänge, Ausgänge, Merker, Zeiten oder Zähler werden dadurch beeinflusst.
Reaktion beim Laden: Ein laufender OPC UA-Server wird nicht unterbrochen.
- Sie möchten ein neues Datenmodul laden und haben das Datenmodul als nicht OPC UA-relevant gekennzeichnet.
Reaktion beim Laden: Ein laufender OPC UA-Server wird nicht unterbrochen.
- Sie möchten ein Datenmodul überschreiben.
Reaktion beim Laden: Eine Warnung erscheint, dass der Server neu gestartet wird.
Hintergrund: STEP 7 kann nicht ermitteln, ob sich die Änderungen auf OPC UA-relevante Daten beziehen oder nicht.

Betriebszustand der CPU über den OPC UA-Server auslesen

11.10.1.3 Einstellungen für den OPC UA-Server

Im Dialog "Optionen" werden die OPC UA-Einstellungen angezeigt, die Sie einstellen können.



In der Tabelle "Einstellungen für den OPC UA-Server" finden Sie weitere Informationen zu den Einstellungen für die Konfiguration des OPC UA-Servers. In der Tabelle werden auch die Einschränkungen des Servers für die S7-1200 ab Firmware V4.5 beschrieben.

Der Standardwert für jede Spalte funktioniert mit der Standard-Kommunikationslast der CPU von 20 %. Werden die maximalen Lastparameter angegeben, muss der Prozentwert für die CPU-

Kommunikationslast erhöht werden. Das ist abhängig von der Last durch andere Kommunikationsaktivitäten (beispielsweise PROFINET).

Tabelle 11-76 Einstellungen für den OPC UA-Server

Servereinstellungen	Min.	Max.	Standardeinstellung	Vom Anwender einstellbar?
Kommunikationsanschlüsse	1024	49151	4840	Ja
Max. Sitzungs-Timeout	1 s	600.000 s	30 s	Ja
Gleichzeitige Sitzungen	1	10	10	Ja
Max. Teilnehmer	100	2000		
Teilnehmer an Serverschnittstelle ¹		1000		Nein
Namensräume insgesamt ²		18		Nein
Servermethoden		10		Nein
Max. Anzahl Instanz-DBs für Servermethoden ³		20	20	Nein
Max. Anzahl Eingabeargumente für Servermethoden		20	20	Nein
Max. Anzahl Ausgabeargumente für Servermethoden		20	20	Nein
Subscription-Einschränkungen				
Min. Abtastintervall	100 ms, 250 ms, 500 ms, 1000 ms, 5000 ms, 10000 ms		1000 ms	Ja
Min. Veröffentlichungsintervall	200 ms	20.000.000 ms	1000 ms	Ja
Gleichzeitige Subscriptions		50	50	Nein
Max. Anzahl Subscriptions pro Sitzung		5		Nein
Max. überwachte Teilnehmer	100	1000	200	Ja

¹ Dies ist die maximale Anzahl benutzerdefinierter Teilnehmer. Darin enthalten sind die Teilnehmer, die implizit an der Serverschnittstelle definiert sind. Die maximale Anzahl enthält nicht die Teilnehmer, die mit der SIMATIC-Serverschnittstelle definiert sind.

² Dies ist die Gesamtzahl der definierten Namensräume für alle Serverschnittstellen und Referenznamensräume.

³ Dies ist die Gesamtzahl der zulässigen Servermethoden-Instanz-DBs für alle Serverschnittstellen und alle zugehörigen Servermethoden. Diese Einschränkung wird vom ES (beim Übersetzen) und vom AS (beim Laden) kontrolliert.

11.10.1.4 S7-1200 als OPC UA-Server verwenden

Konfiguration der OPC UA-Server-Schnittstelle

Wenn Sie die OPC UA-Server-Schnittstellen verwenden, müssen Sie entsprechend der Leistungsfähigkeit der S7-1200 CPU für die folgenden Objekte bestimmte Einschränkungen einhalten.

Hinweise zu Konfigurationsgrenzen bei der Verwendung von Serverschnittstellen

- Anzahl der Serverschnittstellen
- Anzahl der OPC UA-Teilnehmer
- Wenn Sie Methoden implementiert haben: Anzahl der Servermethoden oder Servermethoden-Instanzen

Konfigurationsgrenzen für die OPC UA-Server-Schnittstellen und -Servermethoden

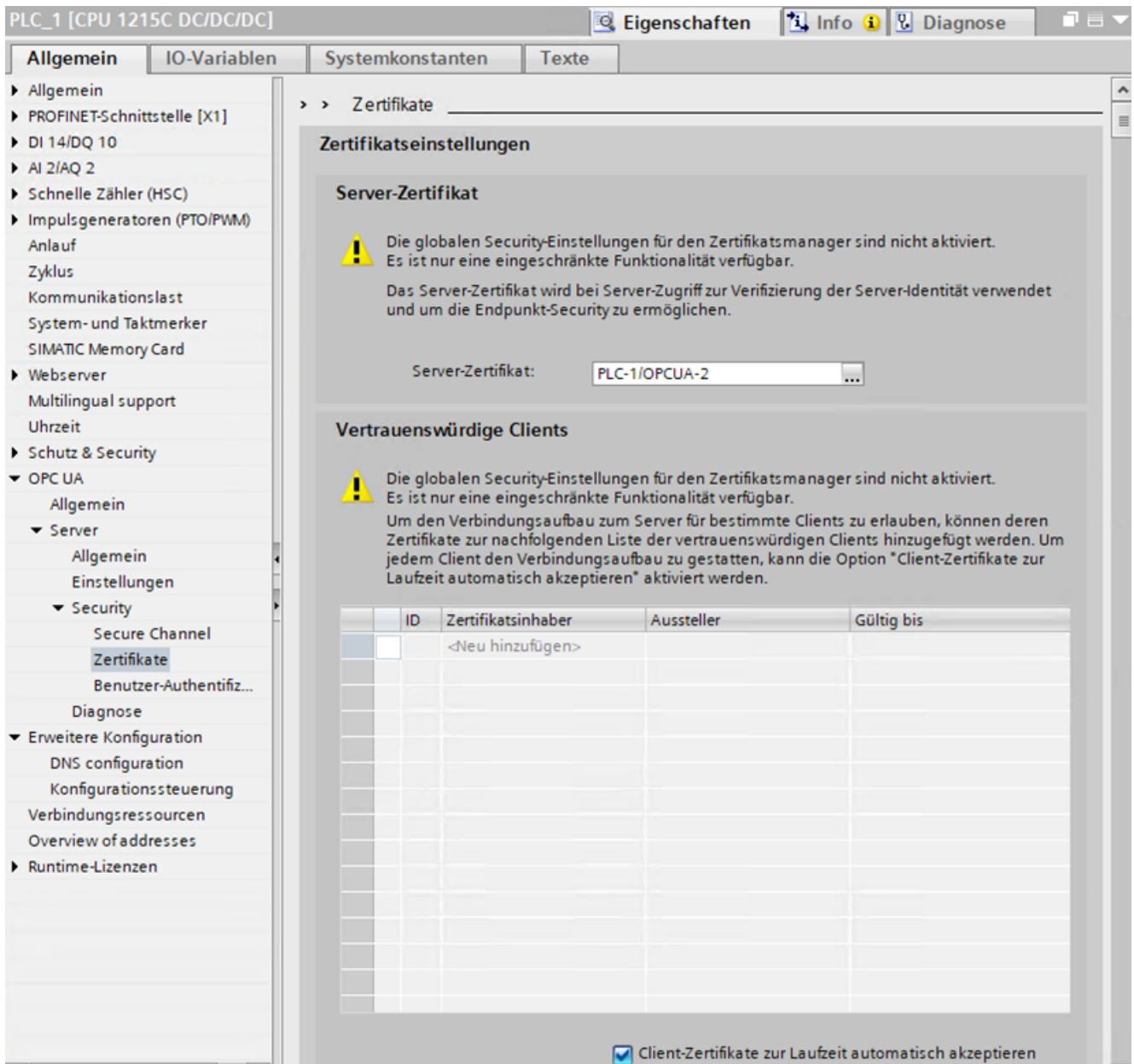
In der nachfolgenden Tabelle finden Sie die Konfigurationsgrenzen für S7-1200 CPUs. Diese müssen auch beim Übersetzen und Laden einer Konfiguration berücksichtigt werden. Eine Verletzung der Konfigurationsgrenzen führt zu einer Fehlermeldung.

Wert der technischen Daten	
S7-1200 CPU	CPU 1211C DC/DC/DC 6ES7211-1AE40-0XB0
	CPU 1211C AC/DC/Relais 6ES7211-1BE40-0XB0
	CPU 1211C DC/DC/Relais 6ES7211-1HE40-0XB
	CPU 1212C DC/DC/DC 6ES7212-1AE40-0XB0
	CPU 1212C AC/DC/Relais 6ES7212-1BE40-0XB0
	CPU 1212C DC/DC/Relais 6ES7212-1HE40-0XB
	CPU 1214C DC/DC/DC 6ES7214-1AG40-0XB0
	CPU 1214C AC/DC/Relais 6ES7214-1BG40-0XB0
	CPU 1214C DC/DC/Relais 6ES7214-1HG40-0XB0
	CPU 1215C DC/DC/DC 6ES7215-1AG40-0XB0
	CPU 1215C AC/DC/Relais 6ES7215-1BG40-0XB0
	CPU 1215C DC/DC/Relais 6ES7215-1HG40-0XB0
Fehlersichere S7-1200 CPU	CPU 1217C DC/DC/DC 6ES7217-1AG40-0XB0
	CPU 1212FC DC/DC/DC 6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/Relais 6ES7212-1HF40-0XB0
	CPU 1214FC DC/DC/DC 6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/Relais 6ES7214-1HF40-0XB0
	CPU 1215FC DC/DC/DC 6ES7215-1AF40-0XB0
CPU 1215FC DC/DC/Relais 6ES7215-1HF40-0XB0	
Verwendung importierter Begleitspezifikationen (Infomodelle)	
Max. Anzahl OPC UA-Server-Schnittstellen:	
• Typ "Begleitspezifikation"	2
• Typ "Referenz-Namensraum"	10
• Typ "Serverschnittstelle"	2
Max. Anzahl OPC UA-Teilnehmer in benutzerdefinierten Serverschnittstellen	2000
Bereitstellung von Methoden	
Max. Anzahl nutzbarer Servermethoden oder max. Anzahl Servermethoden-Instanzen (Anweisungen OPC_UA_Server- MethodPre, OPC_UA_Server- MethodPost)	20

11.10.2 Sicherheit des OPC UA-Servers

Serversicherheit ist lediglich eine der Sicherheitsmethoden zum Sichern der Technik. Andere Methoden zum Sichern der Technik sind u.a. TIAP-Sicherheit und PLC-Sicherheit.

Der OPC UA-Server benötigt ein Zertifikat für die Aktivierung. Das TIA Portal generiert automatisch ein Zertifikat, wenn Sie den Server aktivieren. Sie können dieses Zertifikat in den PLC-Eigenschaften ändern.



Hinweis

Begrenzte Anzahl Zertifikate bei der S7-1200

Die S7-1200 hat einen vom System vorgegebenen Grenzwert von 64 Zertifikaten.

In diese Anzahl werden alle Zertifikate eingerechnet (z.B. Webzertifikate, OPC UA-Zertifikate und OUC-Zertifikate).

Wenn Sie mehr als 64 Zertifikate haben, zeigt das TIA Portal in einer Fehlermeldung an, dass Sie die maximale Anzahl von 64 Zertifikaten überschritten haben. Sie müssen dann einige Zertifikate aus der PLC-Konfiguration entfernen.

11.10.2.1 Unterstützte Sicherheitsrichtlinien

Sie können eine der von Ihrem Server unterstützten Sicherheitsrichtlinien wählen. Die von Ihnen zur Laufzeit ausgewählte Sicherheitsrichtlinie bestimmt die Sicherheit der Kommunikation zwischen Client und Server.

Zum Auswählen Ihrer Sicherheitsrichtlinie für OPC UA gehen Sie wie folgt vor:

1. Wählen Sie in der Gerätekonfiguration das Register "Allgemein" aus.
2. Wählen Sie im Fenster "Allgemein" "OPC UA" aus.
3. Wählen Sie im Fenster "OPC UA > Server > Sicherheit" die Option "Sicherer Kanal".
4. Fügen Sie im Feld "Serverzertifikat" die unterstützte Sicherheitsrichtlinie hinzu.

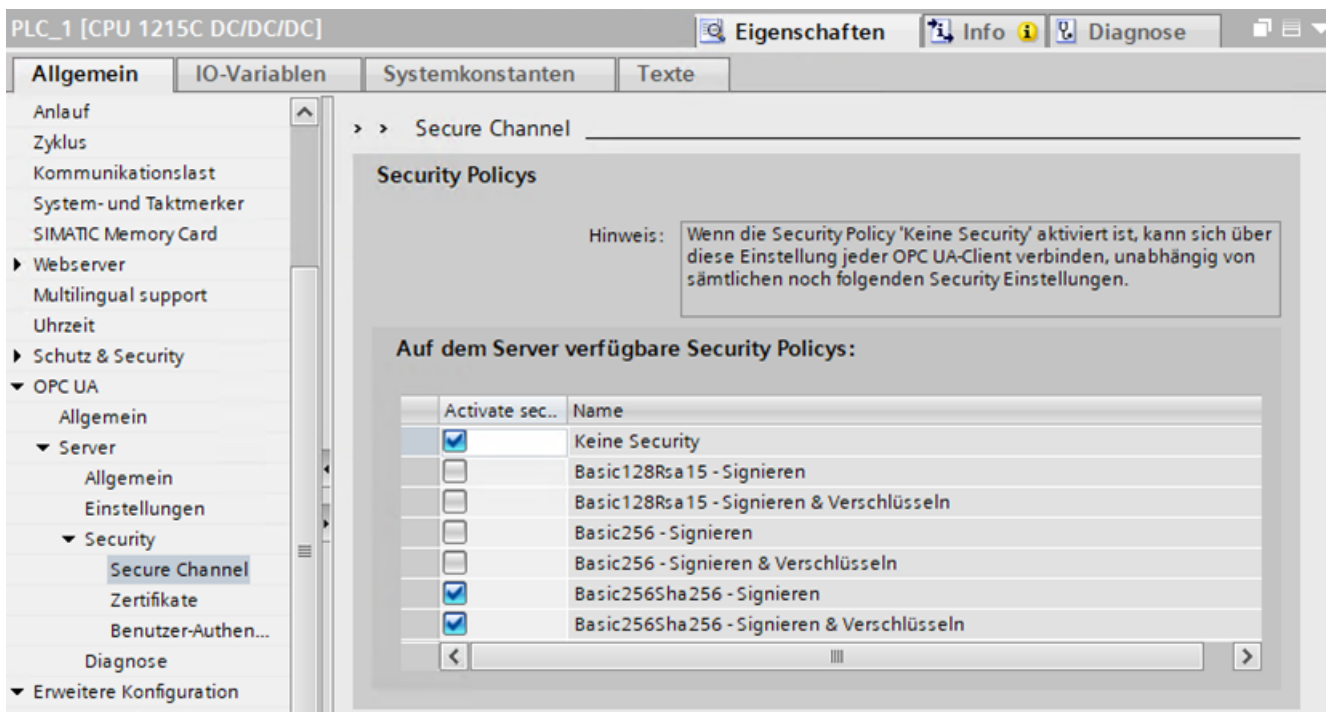


Tabelle 11-77 Von der S7-1200 unterstützte OPC UA-Sicherheitsrichtlinien

Sicherheitsrichtlinie	Standardmäßig aktiviert?
Keine Sicherheit	Ja
Basic128Rsa15 – Sign	Nein
Basic128Rsa15 – Sign & Encrypt	Nein
Basic256 – Sign	Nein
Basic256 – Sign & Encrypt	Nein
Basic256Sha256 – Sign	Ja
Basic256Sha256 – Sign & Encrypt	Ja

Hinweis

Eine sichere OPC UA-Verbindung herstellen

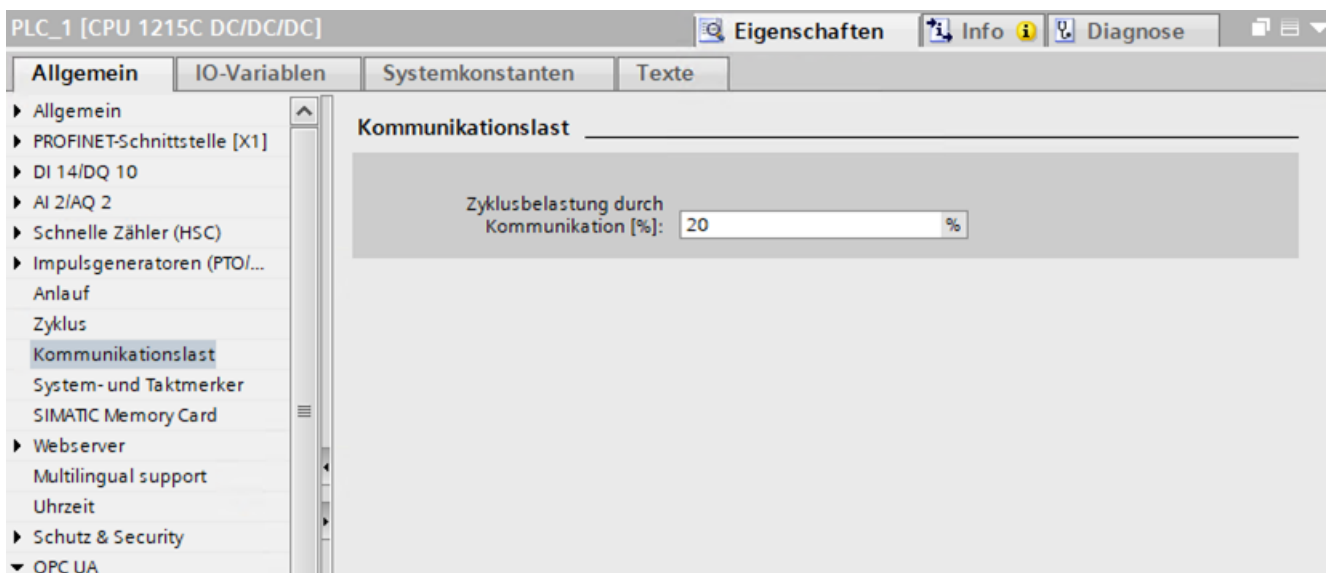
Ausgereifere OPC UA-Sicherheitsrichtlinien können erfordern, dass der Prozentsatz für die Kommunikationslast vom angegebenen Standardwert heraufgesetzt wird, um Online-Verbindungen mit externen Geräten wie TIA Portal, HMI-Geräten usw. aufrechtzuerhalten.

Beim Herstellen einer Verbindung von einem OPC UA-Client zum S7-1200 OPC UA-Server beispielsweise kann eine bestehende Online-Verbindung zum TIA Portal abgebrochen werden. Das beruht auf den anfänglichen zeitintensiven Berechnungen, die zum Herstellen der sicheren Verbindung erforderlich sind und die dazu führen können, dass es bei der Online-Verbindung zum TIA Portal zu einem Timeout kommt.

Beachten Sie, dass die höhere Sicherheit sich über die anfängliche Herstellung der Verbindung hinaus nicht nachteilig auf die OPC UA Client/Server-Kommunikation auswirkt.

Eine Lösung für dieses Problem ist es, die Online-Verbindung einfach erneut herzustellen. Eine andere Lösung ist es, den Standardprozentsatz für die Kommunikationslast wie nachfolgend gezeigt heraufzusetzen.

Beachten Sie jedoch, dass sich durch das Erhöhen des Standardprozentsatzes für die Kommunikationslast die Zykluszeit proportional dazu verlängert.

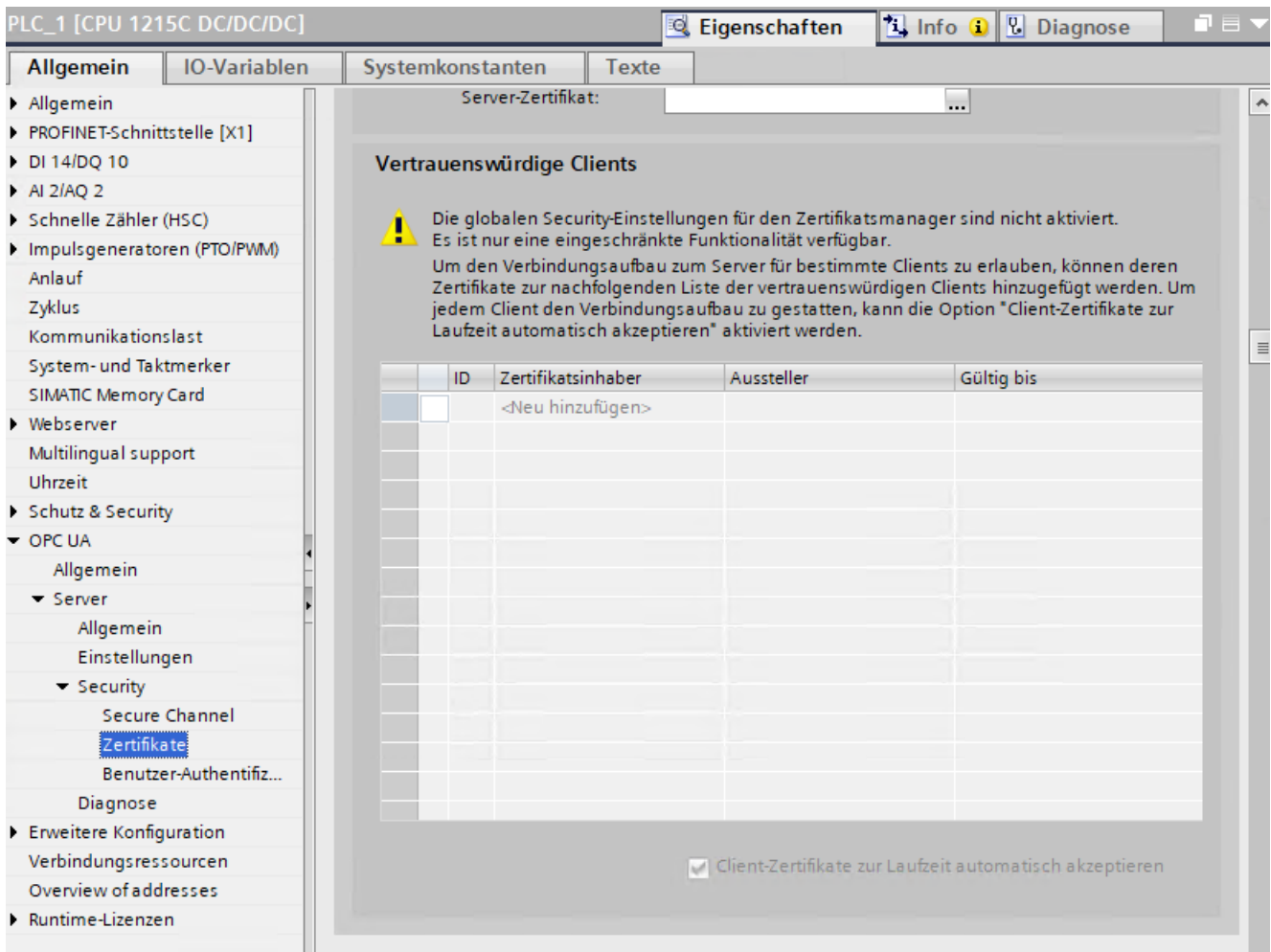


11.10.2.2 Vertrauenswürdige Clients

Der OPC UA-Server kann so konfiguriert werden, dass er nur Verbindungen von vertrauenswürdigen Clients gestattet. Standardmäßig ist der Server so konfiguriert, dass er Client-Zertifikate automatisch akzeptiert.

Sie definieren die Liste der vertrauenswürdigen Clients, die anhand ihrer Zertifikate erkannt werden. Wenn ausgewählt, wird es zur Laufzeit nur Clients mit vertrauenswürdigen Zertifikaten gestattet, eine Verbindung zum Server herzustellen.

Um vertrauenswürdige Clients anzugeben, fügen Sie deren Zertifikate der Liste "Vertrauenswürdige Clients" im TIA Portal unter "Hardware-Eigenschaften > OPC UA-Sicherheit > Sicherer Kanal > Vertrauenswürdige Clients" hinzu.



11.10.2.3 Benutzer-Authentifizierung

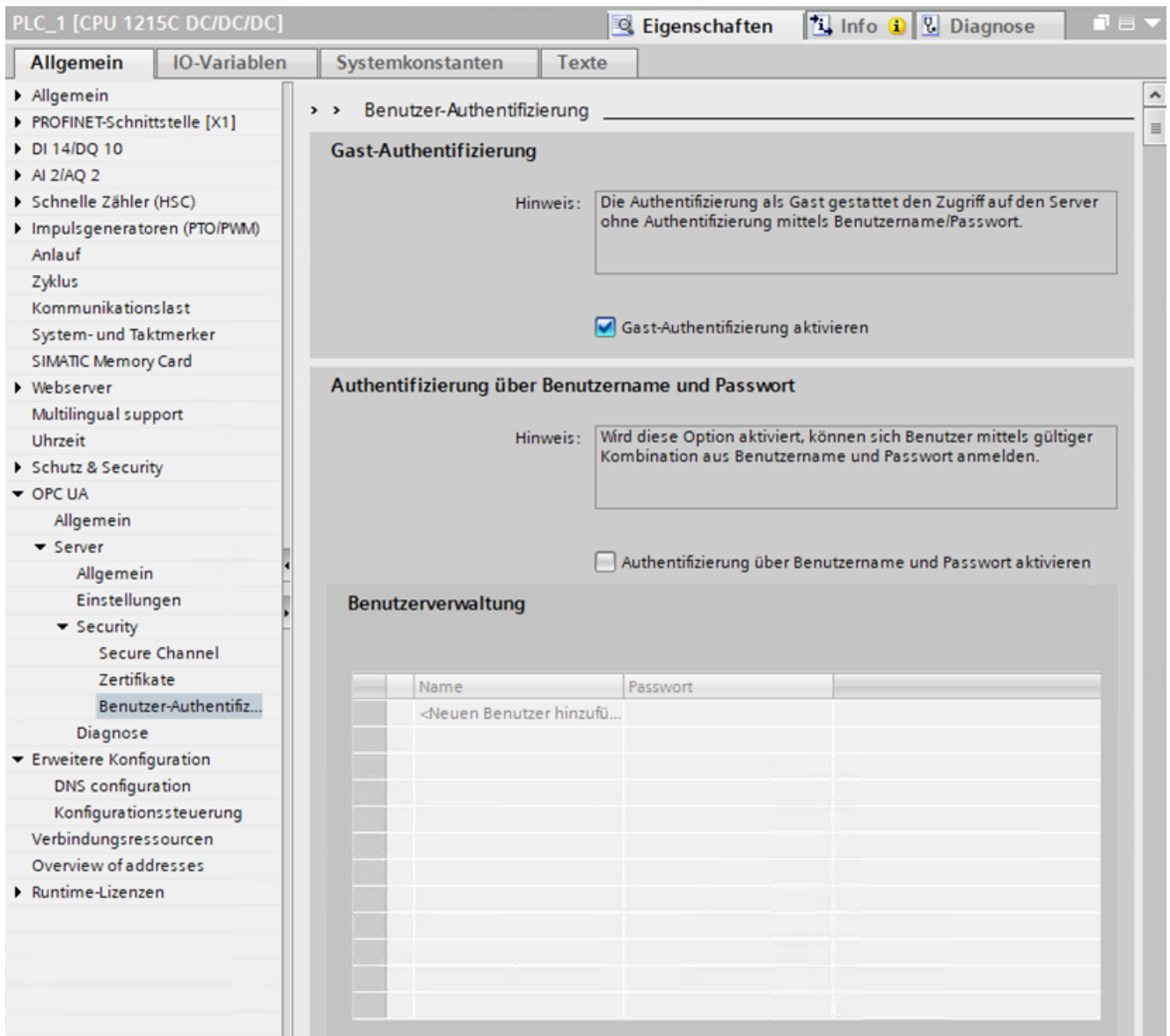
Die S7-1200 unterstützt sowohl die Gastbenutzer-Authentifizierung als auch die Authentifizierung mittels Benutzername und Passwort. Die Voreinstellung ist die Gastbenutzer-Authentifizierung.

Wenn die Gastbenutzer-Authentifizierung aktiviert ist, wird der Client nicht aufgefordert, beim Herstellen der Verbindung einen Benutzernamen und ein Passwort einzugeben.

Wenn die Gastbenutzer-Authentifizierung deaktiviert ist, dürfen nur Clients, die einen zuvor angegebenen Benutzernamen mit Passwort bereitstellen, eine Verbindung aufbauen.

Sie können gültige Benutzer für den OPC UA-Server auf zwei Arten definieren:

- Sie definieren gültige Benutzer in den Hardware-Eigenschaften der CPU.
- Wenn Sie globale Sicherheit für Ihr Projekt aktiviert haben, werden gültige Benutzer über die Rollen bestimmt, die Sie für einzelne Benutzer von TIA Portal-Projekten angeben. Sie können maximale 21 Benutzer konfigurieren.



11.10.3 Schnittstelle des OPC UA-Servers

Der S7-1200 OPC UA-Server unterstützt die standardmäßige SIMATIC-Serverschnittstelle. Er unterstützt jedoch nicht das automatische "Veröffentlichen" von CPU- und DB-Variablen über diese Auswahl. Stattdessen müssen Sie die Struktur und den Inhalt der Serverschnittstelle im TIA Portal definieren und diese dann in die PLC laden.

Um eine Serverschnittstelle hinzuzufügen, führen Sie die folgenden Schritte durch:

1. Klicken Sie in der Projektnavigation auf den Namen der PLC.
2. Wählen Sie "OPC UA-Kommunikation" aus.
3. Wählen Sie "Serverschnittstellen" aus.
4. Wählen Sie "Neue Serverschnittstelle hinzufügen" aus.
5. Geben Sie die relevanten Informationen für Ihren Server ein.
6. Laden Sie den Server in Ihre PLC.

Wenn Sie eine Serverschnittstelle hinzufügen, beachten Sie, dass alle verfügbaren Variablen in den OPC UA-Elementen des Bildschirms aufgeführt werden. Sie können die Elemente aus dem Fenster "OPC UA-Elemente" in das Fenster "OPC UA-Serverschnittstelle" ziehen. Eine Konsistenzprüfung verifiziert den Inhalt der Serverschnittstelle. Die Schnittstelle kann auch in eine XML-Datei exportiert werden.

11.10.3.1 Unterstützte Datentypen

Die OPC Foundation (<https://opcfoundation.org/>) definiert einen Satz unterstützter Datentypen, die die Struktur des Wertattributs von Variablen und ihren Variablentypen beschreiben. Die S7-1200 V4.5 unterstützt eine Teilmenge dieser Datentypen (Seite 105) sowie andere definierte Typen, die von diesen Datentypen abgeleitet sind.

In der folgenden Tabelle werden die von der S7-1200 V4.5 unterstützten Datentypen aufgeführt:

SIMATIC-Typ	OPC UA-Typname	Teilnehmer-Kennung
Bool	Boolescher Wert	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	Struktur	-/-

Beachten Sie, dass diese Liste die unterstützten grundlegenden Teilnehmertypen darstellt. Sie stellt keine vollständige Liste aller unterstützten Teilnehmertypen dar, weil sich viele SIMATIC-Datentypen den grundlegenden Teilnehmertypen zuordnen lassen. Jeder SIMATIC-Datentyp, der einem grundlegenden Teilnehmertyp zugeordnet werden kann, ist ebenfalls ein unterstützter Teilnehmertyp.

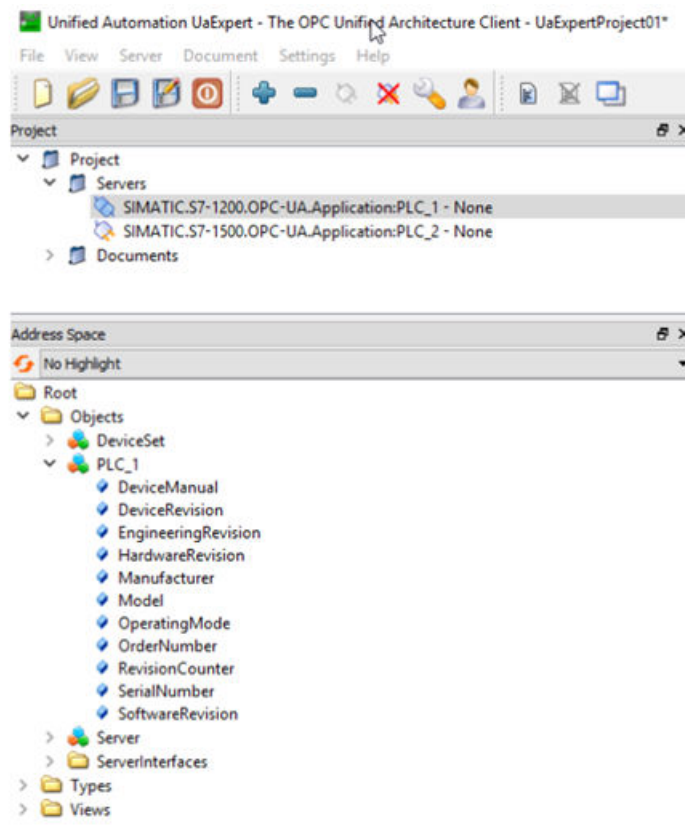
Die S7-1200 CPUs V4.5 unterstützen Servermethoden sowie strukturierte Datentypen (Strukturen und Arrays).

Vereinigungen werden von S7-1200 CPUs nicht unterstützt.

Die S7-1200 akzeptiert den Download eines Servers mit nicht unterstützten Typen. Die PLC gibt jedoch einen Fehler zurück, wenn der Client versucht, einen Teilnehmer mit einem nicht unterstützten Typen zu lesen oder zu schreiben.

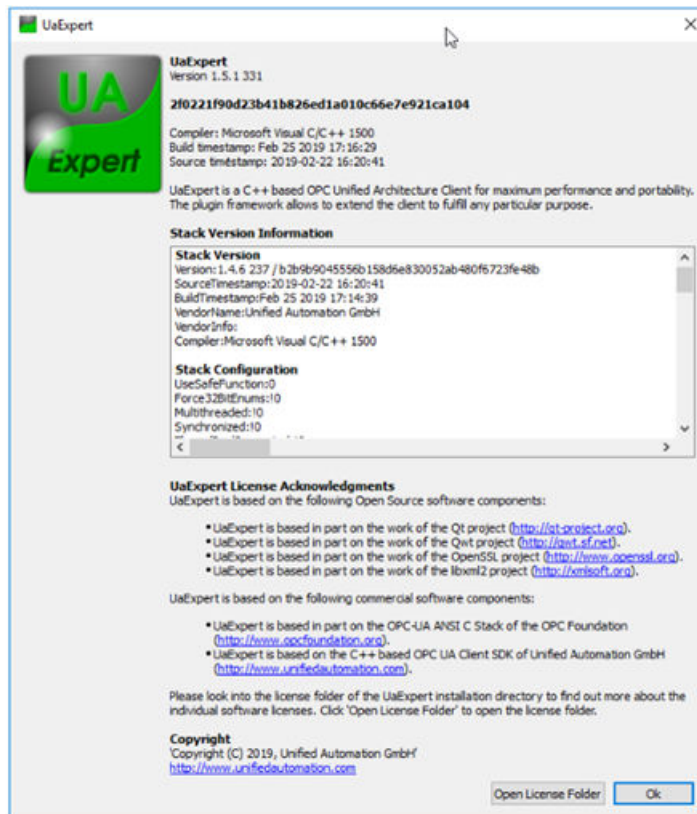
11.10.3.2 PLC-Darstellung

Die OPC UA-Serverschnittstelle stellt Knoten bereit, die die PLC darstellen, mit Eigenschaften, die die PLC beschreiben. Diese sind immer dann verfügbar, wenn der OPC UA-Server aktiviert wird.



Die folgenden Informationen werden von der standardmäßigen SIMATIC-Serverschnittstelle bereitgestellt:

Attribut	Wert		
Nodeld	Ns=SI;s=PLC		
BrowseName	SI:<PLC> Dabei ist <PLC> der Name, den der Benutzer der PLC im TIA Portal-Projekt zugeordnet hat.		
Referenzen	NodeClass	BrowseName	Kommentar
ComponentOf des Objekts DeviceSet			
OrganizedBy des Ordners Objects			
HasTypeDefinition	ObjectType	SimaticDeviceType	Abgeleitet vom Gerätetyp
HasProperty	Variable	DeviceManual	
HasProperty	Variable	DeviceRevision	
HasProperty	Variable	EngineeringRevision	
HasProperty	Variable	HardwareRevision	
HasProperty	Variable	Symbol	
HasProperty	Variable	Manufacturer	
HasProperty	Variable	Model	
HasProperty	Variable	OperatingMode	
HasProperty	Variable	OrderNumber	
HasProperty	Variable	RevisionCounter	
HasProperty	Variable	SerialNumber	
HasProperty	Variable	SoftwareRevision	



11.10.3.3 Ladbare Serverschnittstellen

Sie erstellen und bearbeiten Komponenten der OPC UA-Serverschnittstelle im TIA Portal. Sie können OPC UA-Serverschnittstellen auf zwei Arten definieren:

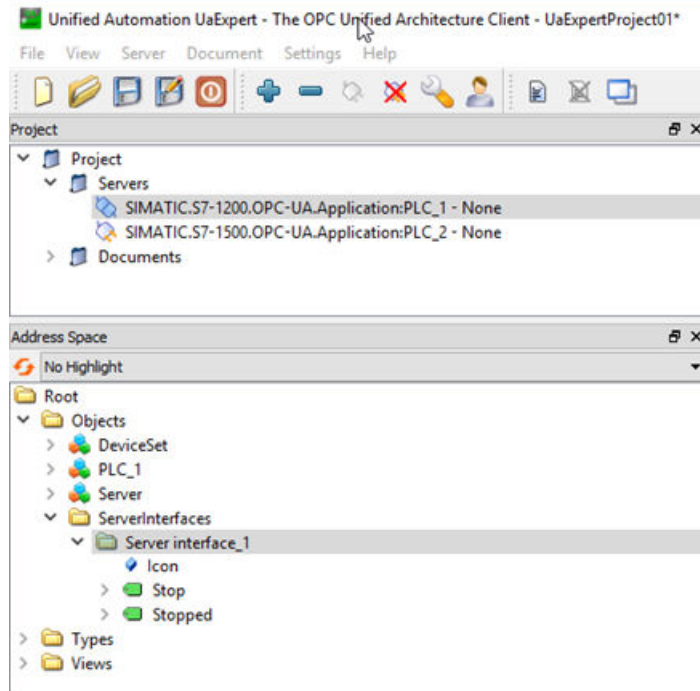
- Mit Hilfe von extern erstellten XML-Dateien (die Begleitspezifikationen genannt werden)
- Mit Hilfe einer direkt im TIA Portal definierten Serverschnittstelle, die auf Datenbausteinelementen und globalen Variablen basiert, die Sie in Ihr Programm aufnehmen können

Beim Laden in das Zielsystem wird auf diese Weise die Serverschnittstelle festgelegt, die gegenüber einem OPC UA-Client sichtbar ist.

Sie müssen spezifische TAG-Attribute festlegen, damit das TAG von OPC UA gelesen und geschrieben werden kann.

Im Dialog für die Servererstellung können Sie die Serviceschnittstelle hinzufügen oder definieren. So definieren Sie die OPC UA-Serverschnittstelle im TIA Portal:

- Wählen Sie "Serverschnittstelle", Typ: Schnittstelle
- Wählen Sie "Begleitspezifikation", Typ: Begleitspezifikation
- Wählen Sie "Begleitspezifikation", Typ: Referenz-Namensraum



		OPC UA-Server-Schnittstelle			
		Browse Name	Knotentyp	Zugriffsstufe	Lokaldaten
▼ PLC_1 [CPU 1214C DC/DC/DC]	Gerätekonfiguration	1	▼ Server-Schnittstelle_1	Schnittstelle	---
	Online & Diagnose	2	■ Stop	Bool	RD/WR
	Programmbausteine	3	■ Stopped	Bool	RD/WR
	Technologieobjekte	4	 <Neu hinzufügen>		
	Externe Quellen				
	PLC-Variablen				
	PLC-Datentypen				
	Beobachtungs- und Forcetabell...				
	Online-Sicherungen				
	Traces				
	OPC UA-Kommunikation				
	Server-Schnittstellen				
	Neue Server-Schnittstelle				
	Server-Schnittstelle_1				

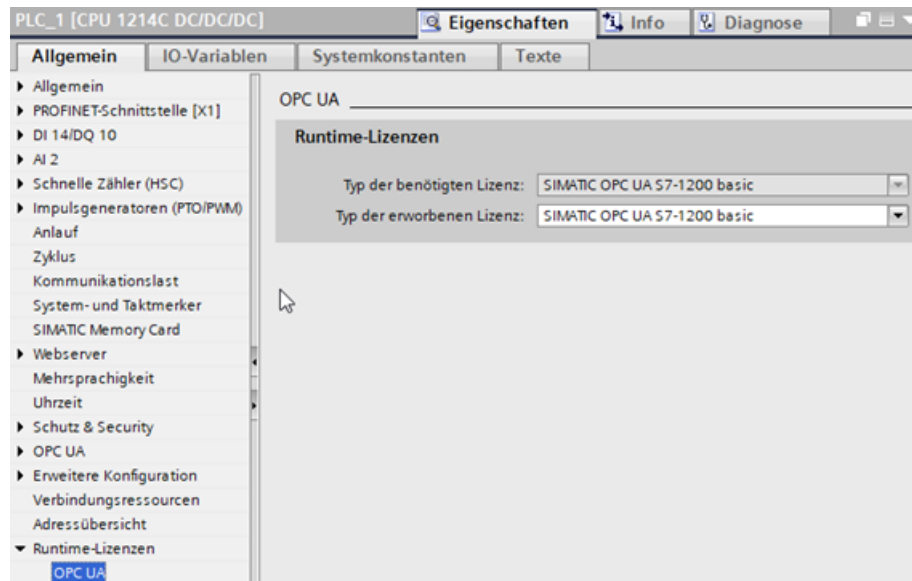
Runtime-Lizenz erforderlich

Eine Runtime-Lizenz ist erforderlich, um den OPC UA-Server für die S7-1200 CPU auszuführen. Die folgenden Lizenzen sind verfügbar:

- SIMATIC OPC UA S7-1200 Basic DVD 6ES7823-0BA00-2BA0
- SIMATIC OPC UA S7-1200 Basic DL 6ES7823-0BE00-2BA0

Der erforderliche Lizenztyp wird unter "Eigenschaften > Allgemein > Runtime-Lizenzen > OPC-UA > Typ der erforderlichen Lizenz" angezeigt. Um den Erwerb der erforderlichen Lizenz zu bestätigen, gehen Sie wie folgt vor:

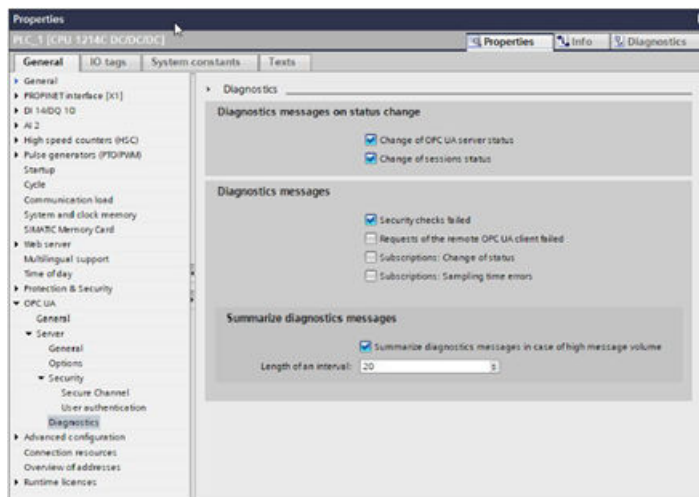
1. Klicken Sie in den Eigenschaften der CPU auf "Runtime-Lizenzen > OPC UA".
2. Wählen Sie in der Klappliste "Typ der gekauften Lizenz" die erforderliche Lizenz aus.



11.10.4 OPC UA-Diagnosepuffer

Bei S7-1200 V4.5 können Sie mit dem OPC UA-Server Ereignisse für eine Komponente generieren, die alle OPC UA-Diagnosemeldungen verwaltet und die Meldungen dem Diagnosepuffer der CPU S7-1200 hinzufügt. Immer wenn eine Unterkomponente von OPC UA ein Ereignis an diese Komponente ausgibt, prüft die zentrale Komponente, ob die Diagnosemeldung in der Konfiguration eingeschaltet ist und dem Diagnosepuffer hinzugefügt werden muss.

Sie konfigurieren die OPC UA-Diagnose im TIA Portal V17 und laden die Einstellungen in die PLC. Stoßen Sie dann Diagnosemeldungen mit einem OPC-Client an und prüfen Sie die Einträge im Diagnosepuffer über den OPC-Client.



Wie gezeigt haben Sie folgende Auswahlmöglichkeiten:

- Diagnosemeldungen bei Statusänderung
- Diagnosemeldungen
- Zusammengefasste Diagnosemeldungen

Diagnosemeldungen bei Statusänderung

Für die Arten von Diagnosemeldungen, die über die zwei Kontrollkästchen "Änderung des OPC UA-Serverstatus" und "Änderung des Sitzungsstatus" gesteuert werden, gibt es keine zusammengefassten Meldungen. Es wird nicht erwartet, dass diese Arten von Meldungen häufig auftreten. Jede dieser Meldungen hat eine große Bedeutung für die Nutzung von OPC UA. Bedeutung der Kontrollkästchen:

"Änderung des OPC UA-Serverstatus" umfasst die Meldungsarten:

- "OPC UA-Serverzustand geändert in – Grund:"
- "OPC-UA-Server hat niedrigste Sicherheitsrichtlinie aktiviert"

"Änderung des Sitzungsstatus" umfasst die Meldungsarten:

- "Sitzungsstatus von OPC UA-Server geändert in – SessionID:"

Diagnosemeldungen

Alle Meldungsarten dieser Kategorie unterstützen den Erfassungsmechanismus. Bedeutung der Kontrollkästchen:

- **"Sicherheitsprüfungen fehlgeschlagen"** umfasst die Meldungsart:
"Sicherheitsprüfungen von OPC UA-Server fehlgeschlagen."
- **"Requests des entfernten OPC UA-Client fehlgeschlagen"** umfasst:
"OPC UA-Server: Falsche Verwendung des Services
"OPC UA-Server: Servicestörung mit Status
"OPC UA-Server: Grenzwert überschritten von
"OPC UA-Server: Fehler aufgetreten im Serverzustand
 - **"Subscriptions: Änderung des Status"** umfasst "Subscription-Zustand von OPC UA-Server geändert in Subscription-ID:
 - **"Subscriptions: Abtastzeitfehler"** umfasst "Abtastrate von OPC UA-Server konnte nicht erreicht werden. Überlast von Subscription-ID"

Zusammengefasste Diagnosemeldungen

Diese Parametrierung gilt nur für Diagnosemeldungen des OPC UA-Servers. Unterschiedliche Einstellungen können für die CPU-globalen Sicherheitsereignisse gewählt werden.

Das Kontrollkästchen "Diagnosemeldungen bei hohem Meldeaufkommen zusammenfassen" legt fest, ob der Erfassungsmechanismus für alle Diagnosemeldungen des OPC UA-Servers aktiviert wird oder nicht. Die Länge des Erfassungsintervalls kann in Sekunden angegeben werden. Dies unterscheidet sich vom Dialog der CPU-globalen Sicherheitsereignisse, wo auch Minuten und Stunden unterstützt werden. Bei spezifischen Meldungen werden nur Sekunden als Zeiteinheit unterstützt. Der Wertebereich umfasst 1 s bis 7200 s.

Hinweis

OPC UA-Server aktiviert/deaktiviert

Die Einstellungen im OPC UA-Diagnosemenü können nur geändert werden, wenn der OPC UA-Server aktiviert ist.

Wenn der OPC UA-Server deaktiviert ist, sind alle Einstellungen gegraut und können nur gelesen werden.

11.10.4.1 OPC UA-Grenzwerte erreicht

Sie können jetzt informiert werden, wenn eine Systemgrenze erreicht wird. Sie erfahren, dass Daten nicht bereitgestellt werden können oder dass ein Server/Client nicht wie erwartet verwendet werden kann.

Wenn spezifische Systemgrenzen erreicht werden, wird eine Meldung im Diagnosepuffer eingetragen. Folgende Grenzwerte gibt es:

- Die maximale Anzahl Subscriptions / überwachte Elemente
- Die maximale Anzahl registrierte Elemente
- Die Abtastrate von Subscriptions kann nicht erreicht werden
- Die Anzahl verbundener Clients

Ein neuer Eintrag im Diagnosepuffer wird erstellt, wenn ein bestimmter Grenzwert erreicht wird. Der Text der Meldung lautet:

OPC UA-Server: Grenzwert von <Name des Grenzwerts> überschritten.

Für die folgenden Grenzwerte gibt es vordefinierte Texte:

- Anzahl der Sitzungen
- Anzahl überwachter Elemente
- Anzahl Subscriptions
- Anzahl registrierter Teilnehmer
- Anzahl Servermethoden
- Anzahl überwachter Elemente pro Aufruf
- Anzahl Teilnehmer pro Browse
- Anzahl Teilnehmer pro Read

- Anzahl Teilnehmer pro RegisterNodes
- Anzahl Teilnehmer pro TranslateBrowsePathsToNodeIds
- Anzahl Teilnehmer pro Write
- Anzahl Teilnehmer pro MethodCall
- Speicherverbrauch

Hinweis

Servicerequest überschreitet einen konfigurierten Grenzwert oder eine Systemgrenze

Wenn ein Servicerequest einen konfigurierten Grenzwert oder eine Systemgrenze überschreitet, reagiert der Server in den meisten Fällen mit einer Servicestörung.

Die folgenden Anwendungsfälle werden nachstehend gezeigt:

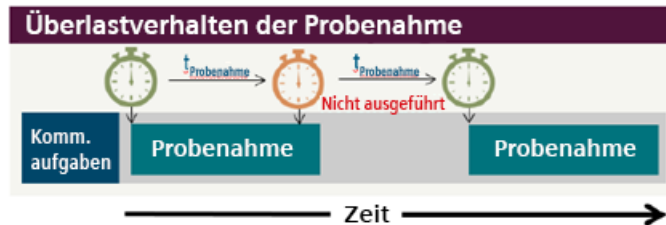
Tabelle 11-78 Anwendungsfall für "OPC UA-Grenzwerte erreicht"

Benutzeraktion	Erwartetes Verhalten	Grenzwert
Client erstellt mehr Sitzungen als von HWCN zugelassen	Diagnosemeldung mit Grenzwert von <Grenzwert> wird dem Diagnosepuffer hinzugefügt.	Sitzungen
Client erstellt mehr überwachte Elemente als von HWCN zugelassen	Diagnosemeldung mit Grenzwert von <Grenzwert> wird dem Diagnosepuffer hinzugefügt.	Überwachte Elemente
Client erstellt mehr Subscriptions als zulässig (abhängig vom PLC-Typ)	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	Client versucht, eine Verbindung zu einem nicht unterstützten Identitäts-Token herzustellen
Client registriert mehr Teilnehmer als von HWCN zugelassen	Diagnosemeldung mit Grenzwert von <Grenzwert> wird dem Diagnosepuffer hinzugefügt.	Registrierte Teilnehmer
Client löst eine Antwort BadOutOfMemory vom Server durch eine hohe Anzahl von Sitzungen / Subscriptions / registrierten Teilnehmern aus	Diagnosemeldung mit Grenzwert von <Grenzwert> wird dem Diagnosepuffer hinzugefügt.	Speicherverbrauch
Client überschreitet eine Betriebsgrenze eines Service (Anzahl Operationen in einem einzigen Request überschreitet den vom Server vorgegebenen Grenzwert).	Diagnosemeldung mit Grenzwert von <Grenzwert> wird dem Diagnosepuffer hinzugefügt.	Teilnehmer pro Browse Teilnehmer pro Read Teilnehmer pro Write Teilnehmer pro MethodCall Teilnehmer pro RegisterNodes Teilnehmer pro Translate

Diagnosemeldung für "Überlastverhalten von Subscriptions"

Bei einer OPC UA-Subscription von verschiedenen Elementen (Variablen) prüft der OPC UA-Server der SIMATIC die Elemente in vordefinierten Intervallen (Abtastintervalle) auf Wertänderungen. Diese Prüfung (Abtastung) erfordert eine bestimmte Zeit, die von der Anzahl

und vom Datentyp der Elemente unabhängig ist. Nach der Abtastung gibt es einen Veröffentlichungsauftrag und der Server sendet die Elemente an den Client. Wenn die Warteschlange zu viele Elemente enthält, kann es zu einer "Überlast" des Kommunikationsstapels kommen. Die CPU kann nicht alle Elemente im angegebenen Abtastintervall prüfen und muss somit zum nächsten Abtastauftrag springen. In diesem Fall sendet die CPU den Statuscode "GoodOverload" (0x002F0000) für jedes Element, selbst wenn die Elemente nicht geprüft wurden. Die Bedeutung des Statuscodes nach IEC 61131-3 lautet: "Abtastung aufgrund von Ressourceneinschränkungen verlangsamt". Das Verhalten wird nachstehend beschrieben.



Wenn eine spezifische Abtastrate nicht erreicht werden kann, empfängt der Client einen Wert mit dem Status "GoodOverload". Es wird ebenfalls eine Meldung "Überlast Abtastung" im Diagnosepuffer eingetragen:

"Abtastrate von OPC UA-Server konnte nicht erreicht werden. Überlast von Subscription-ID"

Die Meldung wird nur ausgelöst, wenn der Status des überwachten Elements von "Good" nach "GoodOverload" wechselt.

11.10.4.2 Entferntes Lesen des Diagnosepuffers bei OPC UA

Sie können jetzt mit OPC UA den Diagnosepuffer aus der Ferne auslesen.

Der OPC UA-Server bietet im Bereich "Diagnose" ein Zeichenkettenfeld, in dem nur der neueste Diagnosepuffereintrag verfügbar ist. Sie können eine Subscription für dieses Feld erstellen und auf diese Art alle Meldungen nachverfolgen.

Sie konfigurieren die OPC UA-Diagnose im TIA Portal und laden die Einstellungen in die PLC. Stoßen Sie dann Diagnosemeldungen mit einem OPC-Client an und prüfen Sie die Einträge im Diagnosepuffer über den OPC-Client.

11.10.4.3 OPC UA-Sicherheitsereignisse

Sie können sich jetzt bei jedem Sicherheitsereignis im OPC UA-Server informieren lassen, so dass Sie reagieren können, wenn der OPC UA-Server verletzlich ist oder Angriffe erkannt werden. Werden bestimmte OPC UA-Sicherheitsereignisse im OPC UA Server/Client-System ausgelöst, werden Meldungen in den Diagnosepuffer eingetragen.

Ein Sicherheitsereignis könnte beispielsweise sein, dass Sitzungen oder Verbindungsversuche aufgrund von falschen Authentifizierungsdaten abgelehnt werden.

Sicherheitsprobleme in Bezug auf den OPC UA-Server werden von folgendem Meldungstyp gemeldet:

"OPC UA-Server: Sicherheitsprüfungen fehlgeschlagen"

Die Meldungen werden generiert, wenn eine Sicherheitsprüfung fehlschlägt und ein Vorgang nicht wie gefordert ausgeführt wird.

In den meisten Fällen wird der Statuscode, der von anderen Sicherheitskomponenten wie OpenSSL, UMAC 711 oder OPC UA Stack zurückgegeben wird, in die Diagnosemeldung geschrieben.

Immer wenn der Anwender eine negative Antwort erhält, wird eine Meldung generiert. Die häufigsten Anwendungsfälle sind nachstehend aufgeführt:

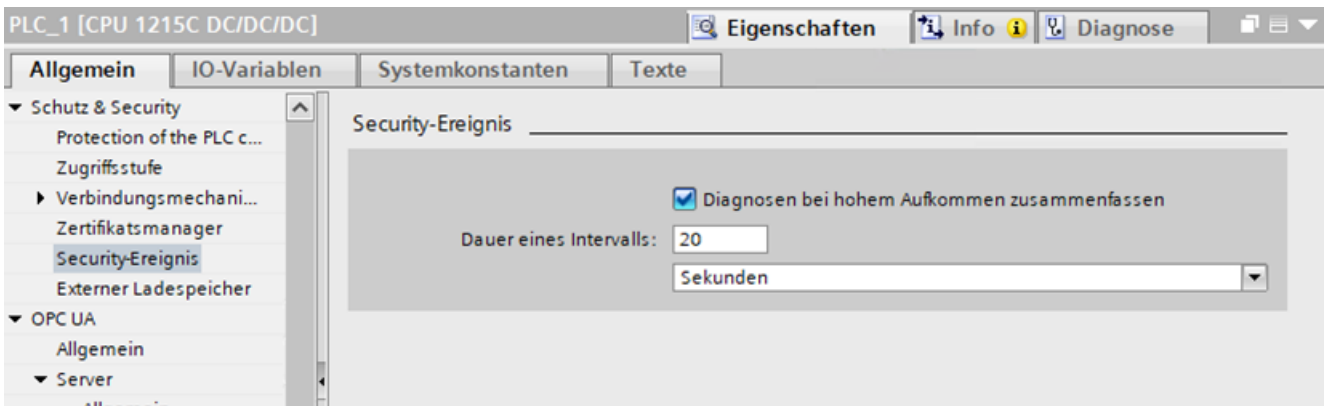
Benutzeraktion	Erwartetes Verhalten	Statuscode
Client versucht, eine Verbindung zu einem nicht unterstützten Sicherheitsmodell herzustellen	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	16#8054_0000 (BadSecurityModeRejected)
Client versucht, eine Verbindung zu einer nicht unterstützten Sicherheitsrichtlinie herzustellen	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	16#8055_0000 (BadSecurityPolicyRejected)
Client versucht, eine Verbindung zu einem nicht unterstützten Identitäts-Token herzustellen	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	Client versucht, eine Verbindung zu einem nicht unterstützten Identitäts-Token herzustellen
Client versucht, eine Verbindung zu einer falschen Authentifizierung herzustellen (Benutzername oder Passwort falsch)	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	16#8021_0000 (BadIdentityTokenRejected)
Client versucht, eine Verbindung zu einem ungültigen Zertifikat herzustellen (es gibt verschiedene mögliche Gründe für die Ungültigkeit von Zertifikaten)	Diagnosemeldung mit Statuscode <Statuscode> wird dem Diagnosepuffer hinzugefügt.	Abhängig vom Grund der Ungültigkeit, z.B. 16#8014_0000 (BadCertificateTimeInvalid) 16#801A_0000 (BadCertificateUntrusted)

Arten von Sicherheitsmeldungen

Derzeit gibt es zwei Arten von Sicherheitsmeldungen: CPU-weite Sicherheitsmeldungen und Sicherheit von OPC UA-Meldungen.

CPU-weite Sicherheitsmeldungen

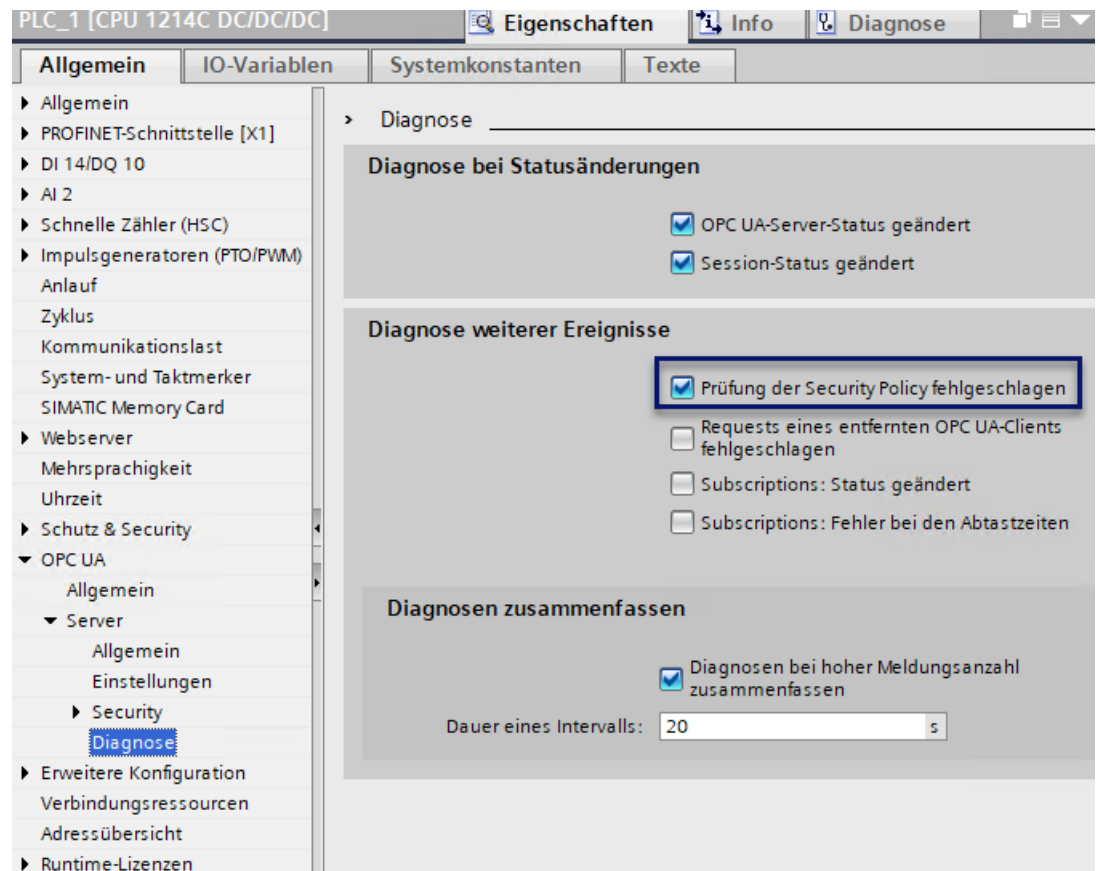
CPU-weite Sicherheitsmeldungen sind Diagnosemeldungen, die in Szenarien generiert werden, die vom Sicherheitsbeauftragten als CPU-weit kritisch eingestuft sind. Solche Meldungen verwenden eine spezielle Alarmdomäne. Dadurch können HMI-Geräte diese Meldungen filtern und unterschiedlich behandeln. Solche Sicherheitsmeldungen sind Authentifizierungsfehler, die auftreten können, wenn ein Anwender versucht, sich am Webserver der CPU oder am OPC UA-Server anzumelden. Im TIA Portal gibt es einen Konfigurationsdialog unter dem Element "Schutz & Sicherheit".



Sicherheit von OPC UA-Meldungen

Die Meldungen von OPC UA in Bezug auf die Sicherheit des OPC UA-Servers werden vom OPC UA-Server generiert und verwenden die Standard-Alarmdomäne oder Prüfungen während der Zertifikateverifizierung.

Sie können die OPC UA-bezogene Sicherheitsprüfung im Eigenschaftsfenster im TIA Portal aktivieren oder deaktivieren (siehe unten).



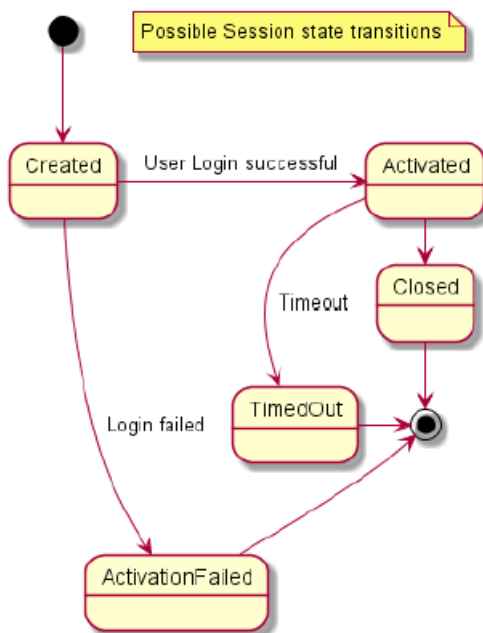
Verbindungsinformation des OPC UA-Servers

Mit dem OPC UA-Diagnosepuffer können Sie jetzt Informationen über verbundene Clients abrufen, um den Status Ihres OPC UA-Server anzuzeigen. Sie können:

- Sehen, wann ein Client eine Verbindung zu Ihrem OPC UA-Server herstellt
- Eine Meldung erhalten, wenn der Client die Verbindung zu Ihrem OPC UA-Server trennt

Eine Diagnosemeldung wird in den Diagnosepuffer geschrieben, wenn sich der Zustand einer Sitzung ändert. Folgende Zustandswechsel von Sitzungen sind möglich:

- Created
- Activated
- Closed
- TimedOut
- ActivationFailed



Die folgenden Anwendungsfälle sind implementiert:

Tabelle 11-79 Implementierte Anwendungsfälle bei aktivierter Sicherheit des OPC UA-Servers

Benutzeraktion	Erwartetes Verhalten	Sicherheitsrichtlinien
OPC UA-Server wurde gestartet (z.B. Aus- und Wiedereinschalten)	Diagnosemeldung mit niedrigster Sicherheitsrichtlinie <Sicherheitsrichtlinie> wird dem Diagnosepuffer hinzugefügt.	Keine, Basic128Rsa15, Basic256, Basic256Sha256,

Tabelle 11-80 Implementierte Anwendungsfälle für Zustandsänderungen von OPC UA-Sitzungen

Voraussetzung	Benutzeraktion	Erwartetes Verhalten	Zustände
OPC UA-Server läuft, kein Client verbunden	Client stellt eine Verbindung zum OPC UA-Server her und übergibt korrekte Anmeldedaten	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Created, Activated
OPC UA-Server läuft, kein Client verbunden	Client stellt eine Verbindung zum OPC UA-Server her und übergibt falsche Anmeldedaten	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Created, Activation failed
OPC UA-Server läuft, Client ist verbunden	Client schließt Sitzung korrekt	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Closed
OPC UA-Server läuft, Client ist verbunden	Client sendet keine weiteren Meldungen an den Server, bis die Sitzung abläuft	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	TimedOut

OPC UA-Server gestartet/gestoppt

Im OPC UA-Diagnosepuffer können Sie jetzt sehen, wenn ein Server gestartet oder gestoppt wird, damit Sie den globalen Status des OPC UA-Servers kennen. Immer wenn der OPC UA-Server gestartet oder gestoppt wird, wird ein Diagnosepuffereintrag angezeigt. Sie können auch zusätzliche Informationen anzeigen, etwa ob die generische Schnittstelle eingeschaltet ist oder wie viele Serverschnittstellen oder Namensräume aktiv sind.

OPC UA-Serverzustände sind:

- Running
- Failed
- NoConfiguration
- Suspended
- Shutdown
- Test
- CommunicationFault
- Unknown
- Starting
- Restarting

Zu den Gründen für eine Zustandsänderung gehören: Ladevorgang/Aus- und Wiedereinschalten, Anweisung vom Anwenderprogramm aufgerufen, entfernter Request. Die möglichen Serverzustandswechsel werden nachstehend gezeigt.

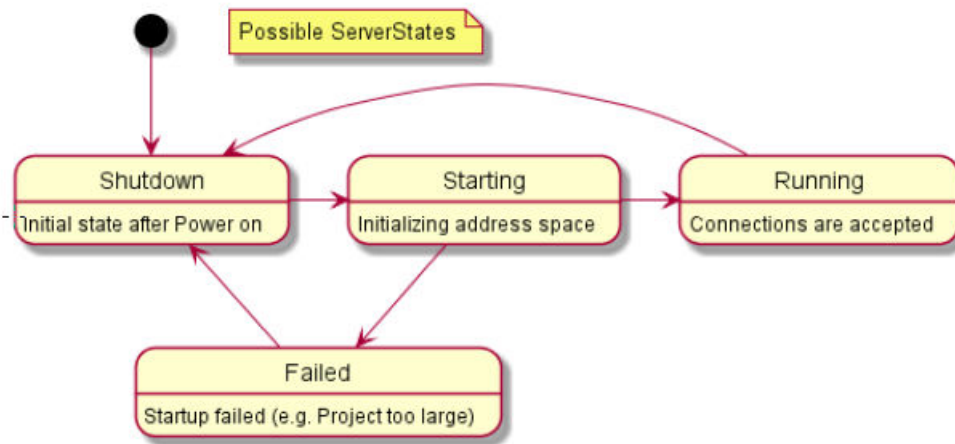


Tabelle 11-81 Implementierte Anwendungsfälle für Zustandsänderungen von OPC UA-Sitzungen

Voraussetzung	Benutzeraktion	Erwartetes Verhalten	Zustände
OPC UA-Server läuft	HW-Download mit aktiviertem OPC UA-Server	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt. Grund: Laden / Aus- und Wiedereinschalten	Shutdown, Starting, Running
OPC UA-Server gestoppt	HW-Download mit aktiviertem OPC UA-Server	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Created, Activated
OPC UA-Server läuft	HW-Download mit deaktiviertem OPC UA-Server	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Created, Activation failed
OPC UA-Server läuft	HW-Download mit aktiviertem OPC UA-Server und zu großem Typwörterbuch (zu viele Strukturen)	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	Closed
OPC UA-Server gestoppt	HW-Download mit aktiviertem OPC UA-Server und zu großem Typwörterbuch	Diagnosemeldungen mit Zustandsänderungen (siehe Spalte "Zustände") werden dem Diagnosepuffer hinzugefügt.	TimedOut

Hinweis

Software-Download

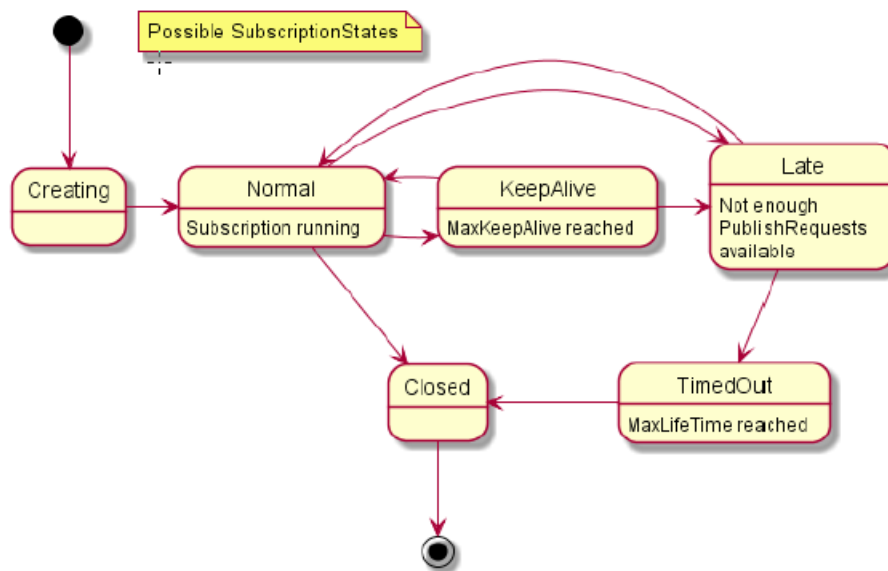
Ein Software-Download führt auch zu einem Neustart des OPC UA-Servers.

Timeout der OPC UA-Sitzung/Subscription

Sie können jetzt sehen, wenn eine OPC UA-Sitzung oder -Subscription abgelaufen ist, so dass Sie erfahren, welche Sitzungen oder Subscriptions noch aktiv sind.

Wenn sich der Zustand einer OPC UA-Sitzung oder Subscription ändert, werden die Informationen zu diesen Ereignissen in den OPC UA-Diagnosepuffer geschrieben. Die möglichen Subscription-Zustände werden nachstehend aufgeführt:

- Created
- Closed
- Normal
- Late
- KeepAlive
- TimedOut



Eine laufende Subscription wechselt häufig zwischen den Zuständen "Normal" und "KeepAlive" (wenn die überwachten Werte sich nur gelegentlich ändern). Für den Zustand "KeepAlive" werden keine Meldungen ausgelöst, weil dadurch dem Diagnosepuffer zu viele Meldungen hinzugefügt würden. Die folgenden Anwendungsfälle sind implementiert:

Tabelle 11-82 Implementierte Anwendungsfälle bei aktivierter Sicherheit des OPC UA-Servers

Benutzeraktion	Erwartetes Verhalten	Sicherheitsrichtlinien
OPC UA-Server wurde gestartet (z.B. Aus- und Wiedereinschalten)	Diagnosemeldung mit niedrigster Sicherheitsrichtlinie <Sicherheitsrichtlinie> wird dem Diagnosepuffer hinzugefügt.	Keine, Basic128Rsa15, Basic256, Basic256Sha256,

11.10.4.4 Falsche OPC UA-Verwendung

Sie können jetzt über die falsche Verwendung des OPC UA-Server durch den Anwender oder den Client informiert werden.

"Falsche Verwendung" wird in Fällen erkannt, in denen der Client Daten oder Funktionalität fordert, die für diese Art der Verwendung nicht vorgesehen sind. Wenn ein Client vor dem

Senden des Request erkennt, dass ein Service oder Request ungültig ist, sollte er den Request nicht senden. Ansonsten wird diese Diagnosemeldung ausgelöst.

Die Anforderung einer ungültigen NodeID kann durch Durchsuchen erkannt werden – also löst das Anfordern eines ungültigen Teilnehmers eine Meldung "Falsche Verwendung" aus. Beim Lesen optionaler Attribute, Überschreiten von Grenzwerten für Subscriptions oder Sitzungen des Servers kann der Client dieses Problem nur durch "Versuchen" erkennen, es wird also keine Meldung über falsche Verwendung ausgelöst.

Hinweis: Der Namensraum "http://opcfoundation.org/UA/" (ns=0) ist ein besonderer Namensraum von der OPC Foundation (bzw. dem SDK) und Diagnosefunktionen sind sehr eingeschränkt. Nicht jede "falsche Verwendung" in diesem Namensraum löst eine Meldung aus (beispielsweise das Registrieren eines unbekanntem Teilnehmers).

Die Diagnosemeldung "*OPC UA-Server: Falsche Verwendung des Services <Servicename> in Sitzungs-ID <Sitzungs-ID>*" wird in den Diagnosepuffer geschrieben, wenn die falsche Verwendung einer der folgenden Services erkannt wurde. Es werden nur Services berücksichtigt, die vom S7-1200 OPC UA-Server unterstützt werden.

- FindServers
- GetEndpoints
- FindServersOnNetwork
- CreateSession
- ActivateSession
- CloseSession
- Cancel
- Browse
- BrowseNext
- TranslateBrowsePathsToNodeIds
- RegisterNodes
- UnregisterNodes
- Write
- Read
- Call
- DeleteMonitoredItems
- ModifyMonitoredItems
- DeleteMonitoredItems
- SetMonitoringMode
- SetTriggering
- DeleteSubscription
- ModifySubscription
- DeleteSubscription
- Publish

- Republish
- SetPublishingMode
- OpenSecureChannel
- CloseSecureChannel

11.10.4.5 Zusammenfassende Meldungen für OPC UA

Beispiel für Diagnosemeldungen im OPC UA-Diagnosepuffer: Falsche Verwendung des Service. Siehe Beispiel unten.

Tabelle 11-83 Zusammenfassung von Meldungen für OPC UA für die falsche Verwendung des Service

Mel- dung	Einfaches Ereignis	Zusammenfassendes Ereignis
Falsche OPC UA-Verwendung		
Definiti- on	'@2W%t#7W@: Falsche Verwendung des Service @4W%t#7W@ in Sitzungs-ID @5X%u@ln	'@2W%t#7W@: Falsche Verwendung des Service Lesen
Beispiel	Beispiel OPC UA-Server: Falsche Verwendung des Service Lesen in Sitzungs-ID 12345678	OPC UA-Server: Falsche Verwendung eines Service. – Zusam- menfassende Meldung. - Zusammenfassende Meldung für drei Vorkommen während der letzten 20 Sekunden

11.10.5 OPC UA-Methodenaufrufe

11.10.5.1 Nützliche Informationen über Servermethoden

Anwenderprogramm für Servermethoden bereitstellen

Auf dem OPC UA-Server einer S7-1200 CPU (ab Firmware V4.5) haben Sie die Möglichkeit, Methoden über Ihr Anwenderprogramm bereitzustellen. Diese Methoden können von OPC UA-Clients verwendet werden, beispielsweise um einen Fertigungsauftrag über den Methodenaufruf der S7-1200 CPU zu starten.

OPC UA-Methoden, eine Implementierung von "Remote Procedure Calls", bieten einen effizienten Mechanismus für Interaktionen zwischen unterschiedlichen Kommunikationsteilnehmern. Der Mechanismus bietet sowohl Auftragsbestätigung als auch Rückführwerte, weshalb Sie keine Handshake-Mechanismen mehr programmieren müssen.

Mit OPC UA-Methoden können Sie Daten konsistent beispielsweise ohne Triggerbits/ Handshaking übertragen oder spezifische Aktionen auf dem Controller anstoßen.

Wie funktioniert eine OPC UA-Methode?

Eine OPC UA-Methode funktioniert im Prinzip wie ein Know-how-geschützter Funktionsbaustein, der von einem externen OPC UA-Client während der Laufzeit aufgerufen wird.

Der OPC UA-Client "sieht" nur die definierten Eingänge und Ausgänge. Der Inhalt des Funktionsbausteins, die Methode oder der Algorithmus, bleibt gegenüber dem externen OPC UA-Client verborgen. Der OPC UA-Client empfängt Rückmeldungen bei erfolgreicher Ausführung und vom Funktionsbaustein (Methode) zurückgegebene Werte oder aber eine Fehlermeldung, wenn die Ausführung nicht erfolgreich war.

Als Programmierer haben Sie die volle Kontrolle über und die Verantwortung für den Programmkontext, in dem die OPC UA-Methode ausgeführt wird.

Regeln zum Programmieren einer Methode und Laufzeitverhalten

- Stellen Sie sicher, dass die von der OPC UA-Methode zurückgegebenen Werte mit den vom OPC UA-Client bereitgestellten Eingangswerten konsistent sind.
- Befolgen Sie die Regeln für die Zuweisung von Name und Struktur von Parametern und den zulässigen Datentypen (siehe Beschreibung der OPC UA-Server-Anweisungen).
- Verhalten während der Laufzeit: Der OPC UA-Server akzeptiert einen Aufruf pro Instanz. Die Methodeninstanz ist für andere OPC UA-Clients erst verfügbar, wenn der Aufruf vom Anwenderprogramm verarbeitet wurde oder die Zeit dafür abgelaufen ist.

Die grundlegende Vorgehensweise zum Implementieren eines Anwenderprogramms als Servermethode wird nachfolgend beschrieben.

Servermethode implementieren

Ein Programm (Funktionsbaustein) zum Implementieren einer Servermethode ist wie folgt strukturiert:

1. Aufruf der Servermethode mit OPC_UA_ServerMethodPre abfragen

Sie rufen zunächst die Anweisung "OPC_UA_ServerMethodPre" in Ihrem Anwenderprogramm auf (in Ihrer Servermethode).

Diese Anweisung hat die folgenden Aufgaben:

- Mit dieser Anweisung fragen Sie den OPC UA-Server der CPU, ob Ihre Servermethode von einem OPC UA-Client aufgerufen wurde.
- Wurde die Methode aufgerufen und die Servermethode hat Eingangsparameter, empfängt Ihre Servermethode jetzt die Eingangsparameter. Die Eingangsparameter der Servermethode kommen vom Aufrufen des OPC UA-Client.

2. Servermethode bearbeiten

In diesem Abschnitt der Servermethode stellen Sie das eigentliche Anwenderprogramm bereit.

Sie haben die gleichen Optionen wie bei jedem anderen Anwenderprogramm (beispielsweise Zugriff auf andere Funktionsbausteine oder globale Datenbausteine). Wenn die Servermethode mit Eingangsparameter arbeitet, sind diese Parameter für Sie verfügbar.

Dieser Abschnitt der Servermethode sollte nur ausgeführt werden, wenn ein OPC UA-Client die Servermethode aufgerufen hat.

Nach erfolgreicher Ausführung der Methode legen Sie die Ausgangsparameter der Servermethode fest (wenn die Methode solche Ausgangsparameter hat).

3. Auf Servermethode mit OPC_UA_ServerMethodPost antworten

Um die Servermethode zu vervollständigen, rufen Sie die Anweisung "OPC_UA_ServerMethodPost" auf.

Verwenden Sie die Parameter, um die Anweisung "OPC_UA_ServerMethodPost" zu benachrichtigen, ob das Anwenderprogramm verarbeitet wurde oder nicht.

Wenn das Anwenderprogramm erfolgreich ausgeführt wurde, wird der OPC UA-Server über die relevanten Parameter benachrichtigt. Der OPC UA-Server sendet dann die Ausgangsparameter der Servermethode an den OPC UA-Client.

Rufen Sie die Anweisungen "OPC_UA_ServerMethodPre" und "OPC_UA_ServerMethodPost" immer als Paar auf, unabhängig davon, ob das Anwenderprogramm von beiden Anweisungen verarbeitet wird oder ob die Bearbeitung im nächsten Zyklus fortgesetzt wird.

Ein Beispiel für die Implementierung einer Servermethode finden Sie in der Online-Hilfe von STEP 7.

Servermethode integrieren

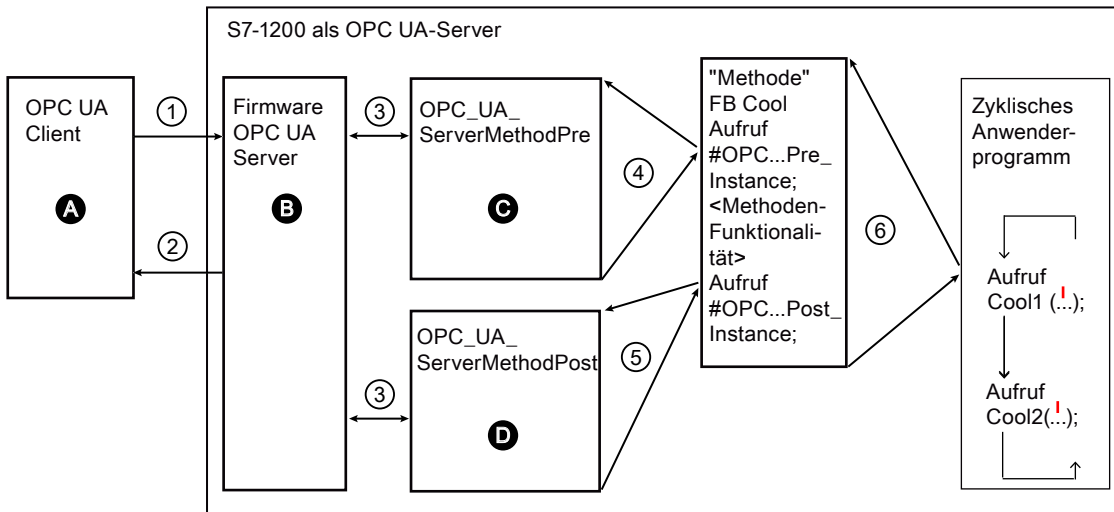
Die nachfolgende Abbildung zeigt, wie ein OPC UA-Client (A) die Servermethode "Cool" aufruft:

Die CPU führt die Instanz "Cool1" der Servermethode "Cool" im zyklischen Anwenderprogramm aus ⑥.

Die CPU fragt zuerst mit der Anweisung "OPC_UA_ServerMethodPre" ab ④, ob ein OPC UA-

Client die Servermethode "Cool" aufgerufen hat ①.

- Wurde die Servermethode nicht aufgerufen, kehrt die Programmausführung über ④ und ⑥ direkt zum zyklischen Anwenderprogramm zurück. Die CPU setzt das zyklische Anwenderprogramm nach "Cool1" fort.
- Wurde die Servermethode aufgerufen, wird diese Information über ④ an die Servermethode "Cool" zurückgegeben. Die tatsächliche Funktionalität wird jetzt in der Servermethode Cool ausgeführt, siehe "<MethodenFunktionalität>" in der Grafik. Die Servermethode benachrichtigt dann mit der Anweisung "OPC-UA-ServerMethodPost" ⑤ die Firmware (B), dass die Anweisung ausgeführt wurde ③. Die Firmware gibt diese Information über ② an den aufrufenden OPC UA-Client (A) zurück. Die CPU setzt das zyklische Anwenderprogramm nach "Cool1" fort.



- A Aufruf der Servermethode und Verwaltung der Information "Fertig" (Methode abgeschlossen)
- ① Asynchroner Aufruf der Servermethode
 - ② Asynchrone Information "Fertig" für den Methodenaufruf (Methode abgeschlossen)
- B Warten auf OPC UA-Client-Aufrufe, Verwaltung der Aufrufe in der Warteschlange, Weiterleiten der Information "Fertig" vom zyklischen Anwenderprogramm zum OPC UA-Client
- ③ Datenübertragung vom OPC UA-Server zu den Methodeninstanzen des Anwenderprogramms und umgekehrt
- C Prüfen, ob die Methode aufgerufen wurde. Ist das der Fall, Weiterleiten der Eingangsdaten vom OPC UA-Server an die Methodeninstanz des Anwenderprogramms und Feedback an die Methodeninstanz, dass die Methode aufgerufen wurde ("aufgerufen").
- ④ Synchroner Aufruf der Anweisung OPC-UA-ServerMethodPre als Multiinstanz mit Angabe des Speicherbereichs für die Eingangsdaten vom OPC UA-Server. Der Rückgabewert gibt an, ob die Methode vom OPC UA-Client aufgerufen wurde oder nicht. ⑤ Prüfen, ob die Methode abgeschlossen wurde oder noch aktiv ist ("beschäftigt").
 - ⑤ Prüfen, ob die Methode abgeschlossen wurde.

- D Ist das der Fall, werden die Ausgangsdaten der Methodeninstanz an den OPC UA-Server weitergeleitet und die Methodeninstanz wird benachrichtigt, dass die Methode abgeschlossen wurde. Der OPC UA-Server wird benachrichtigt.
- ⑥ Aufruf des Methoden-FB (hier: FB Cool) mit der erforderlichen Instanz und den Prozessparametern.

Informationen zu Serveranweisungen

Die Anweisungen "OPC_UA_ServerMethodPre" und "OPC_UA_ServerMethodPost" werden in der Online-Hilfe von STEP 7 unter Anweisungen > Kommunikation > OPC UA > OPC UA-Server ausführlich beschrieben.

Die Abbildung unten zeigt einen ordnungsgemäß formatierten Methodenaufruf.

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 Grenzbedingungen für die Verwendung von Servermethoden

Zulässige Datentypen

Beachten Sie die folgende Regel, wenn Sie Servermethoden bereitstellen:

- Weisen Sie die Datentypen wie nachfolgend gezeigt zu (SIMATIC-Datentyp – OPC UA-Datentyp). Andere Zuweisungen sind nicht zulässig.
- STEP 7 prüft die Einhaltung dieser Regel nicht und verhindert eine inkorrekte Zuweisung nicht. Sie sind für die regelkonforme Auswahl und Zuweisung der Datentypen verantwortlich.

Sie können auch die aufgeführten Datentypen verwenden, beispielsweise als Elemente von Strukturen/Arrays/UDTs für Eingangs- und Ausgangsparameter von selbst erstellten Servermethoden (UAMethod_InParameters und UAMethod_OutParameters).

SIMATIC-Datentyp	OPC UA-Datentyp
BOOL	Boolescher Wert
SINT	SByte
INT	INT16
DINT	INT32
USINT	Byte
UINT	UINT16
UDINT	UINT32
REAL	Float
LREAL	Double
WSTRING	String
DINT	Enumeration (Verschlüsselung Int32) und alle abgeleiteten Datentypen

Webserver

Der Webserver der S7-1200 bietet Webseitenzugriff auf Daten über Ihre CPU und auf Prozessdaten.

Sie können die S7-1200 Webseiten über einen PC oder ein Mobilgerät aufrufen. Bei Geräten mit kleinem Display unterstützt der Webserver eine Sammlung von Basisseiten (Seite 862).

Verwenden Sie einen Webbrowser für den Zugriff auf die IP-Adresse der S7-1200 CPU oder der IP-Adresse eines über Webserver aktivierten CP-(Kommunikationsprozessor-)Moduls (Seite 858) im lokalen Baugruppenträger mit der CPU, um die Verbindung herzustellen. Die S7-1200 unterstützt mehrere gleichzeitige Verbindungen.



Hinweis

Mehrere gleichzeitige Verbindungen zum Webserver

Der S7-1200 Webserver gestattet 30 gleichzeitige Verbindungen (vorausgesetzt, Sie haben genügend dynamische Verbindungen). Offene Browser-Instanzen können jeweils zwischen 2 und 8 Verbindungen verwenden. Der Webserver gestattet bis zu sieben angemeldete Benutzer, doch Siemens empfiehlt, die Anzahl gleichzeitig angemeldeter Benutzer möglichst gering zu halten. Bei mittlerer Auslastung sind im Durchschnitt 7 gleichzeitige Benutzer normalerweise in Ordnung.

Standard-Webseiten

Die S7-1200 umfasst Standard-Webseiten (Seite 861), auf die Sie über den Webbrowser auf Ihrem PC (Seite 855) oder über ein Mobilgerät (Seite 856) zugreifen können:

- Einführung (Seite 866) - Einstiegspunkt in die Standard-Webseiten
- Startseite (Seite 867) - allgemeine Informationen über die CPU
- Diagnose (Seite 868) - ausführliche Informationen über die CPU einschließlich Serien-, Bestell- und Versionsnummer, Programmschutz und Speicherauslastung
- Diagnosepuffer (Seite 870) - der Diagnosepuffer
- Modulinformationen (Seite 872) – Informationen über die lokalen und dezentralen Module und die Möglichkeit von Firmware-Updates für Module
- Kommunikation (Seite 876) - Informationen über die Netzwerkadressen, physikalischen Eigenschaften der Kommunikationsschnittstellen, Statistiken, Parameter sowie eine Verbindungsübersicht und Diagnoseinformationen

- Variablenzustand (Seite 879) - CPU-Variablen und E/A, zugänglich über Adressen oder PLC-Variablenamen
- Beobachtungstabellen (Seite 881) - Beobachtungstabellen, die Sie in STEP 7 konfiguriert haben
- Online-Sicherung (Seite 883) - Möglichkeit, eine Online-CPU zu sichern oder eine frühere Sicherungskopie wiederherzustellen
- Datenprotokolle (Seite 886)- Möglichkeit, eine Liste aller Datenprotokolle auf der PLC anzuzeigen, ein Datenprotokoll von der PLC auf Ihren Computer zu laden, ein Datenprotokoll von der PLC zu löschen und ein Datenprotokoll von der PLC abzurufen und zu löschen
- Anwenderdateien (Seite 888)- Möglichkeit, eine Liste von Anwenderdateien auf der PLC anzuzeigen, eine Anwenderdatei von der PLC auf Ihren Computer zu laden, eine Anwenderdatei von Ihrem Computer in die PLC zu laden und eine Anwenderdatei auf Ihrer PLC zu löschen
- Benutzerdefinierte Seiten (Seite 894)- benutzerdefinierte Webseiten für den Zugriff auf CPU-Daten erstellen
- Dateibrowser (Seite 893) – Browser für in der CPU oder auf einer Memory Card gespeicherte Dateien, beispielsweise Datenprotokolle oder Rezepte
- Anmeldung (Seite 863) - Anmeldung als anderer Nutzer oder Abmeldung.

Diese Seiten sind in die S7-1200 CPU integriert und stehen auf Englisch, Deutsch, Französisch, Spanisch, Italienisch und in vereinfachtem Chinesisch zur Verfügung. Es ist weitere Konfiguration im TIA Portal nötig, um PLC-Diagnosemeldungen anzuzeigen (Kapitel 15). Zum Anzeigen aller Seiten, mit Ausnahme der Einführungs- und der Startseite, sind zusätzliche Benutzerrechte (Seite 852) erforderlich, die Sie in STEP 7 konfigurieren.

Benutzerdefinierte Webseiten

Bei der S7-1200 können Sie auch benutzerdefinierte Webseiten erstellen, die auf CPU-Daten zugreifen können. Sie können diese Seiten mit einer HTML-Authoring-Software Ihrer Wahl anlegen und vordefinierte "AWP"-Befehl (Automation Web Programming) in Ihren HTML-Code einfügen, um auf CPU-Daten zuzugreifen. Im Kapitel Benutzerdefinierte Webseiten (Seite 894) finden Sie spezielle Informationen zum Entwickeln von benutzerdefinierten Webseiten und zur zugehörigen Konfiguration und Programmierung in STEP 7.

Auf die benutzerdefinierten Webseiten können Sie entweder über einen PC oder über ein Mobilgerät durch Aufrufen der Standard- bzw. Basis-Webseiten zugreifen. Sie können auch eine Ihrer benutzerdefinierten Webseiten als Einstiegsseite (Seite 913) für den Webserver konfigurieren.

Web-API

Die S7-1200 CPU bietet auch eine Web-API (Seite 935), die Ihnen als Schnittstelle zum Lesen und Schreiben von Prozessdaten dient.

Voraussetzungen für den Webbrowser

Siemens hat die Webserver-Standardseiten getestet und prüft die Unterstützung folgender Webbrowser:

- Internet Explorer 11
- Microsoft Edge V44
- Microsoft Edge Chromium Based V86
- Mozilla Firefox V64
- Opera V58
- Google Chrome V75
- Android Browser für Android Pie V9
- Mobile Chrome für Android Pie V9
- Mobile Safari und Chrome für iOS V13

Bei Verwendung der HTML-Browserbedienung in einem WinCC-Projekt unterstützt der Webserver die folgenden Siemens HMI-Panels für die Standardseiten:

- Basic Panels
 - Gen 2 KTP400 bis KTP1200
- Comfort Panels
 - TP700 bis TP2200
 - KP400 bis KP1500
 - KTP400
 - TP700 Comfort Outdoor
- Mobile Panels
 - Gen 2 KTP700[F], KTP900[F]
- Unified Comfort Panels
 - Basic Panel (2. Generation)
 - Mobile Panel (2. Generation)

Informationen zu browserspezifischen Einschränkungen, die sich auf die Anzeige von Standard- oder Benutzer-Webseiten auswirken können, finden Sie im Abschnitt Einschränkungen (Seite 937).

Betriebsverhalten des Webserver

Das Betriebsverhalten des Webserver kann von vielen Faktoren beeinflusst werden. Die S7-1200 CPU und das Programmiergerät müssen die Zeit mit anderen Aufgaben teilen, die Ressourcen und Verarbeitungszeit verbrauchen. Wenn Ihr Webserver ein schlechtes Betriebsverhalten zeigt, probieren Sie diese Anpassungen aus, um das Betriebsverhalten des Webserver zu verbessern:

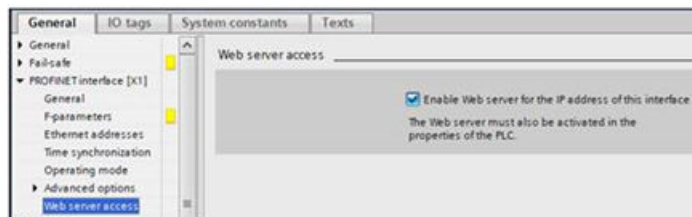
- Erhöhen Sie die Kommunikationsbelastung (Seite 90) auf dem PLC von 20 % auf 50 %.
- Konfigurieren Sie eine minimale Zykluszeit (Seite 90). Das Festlegen einer minimalen Zykluszeit sorgt für eine längere Kommunikationszeit zwischen der S7-1200 CPU und dem Programmiergerät.
- Verwenden Sie statt eines CP-Moduls (Seite 858) für den Zugriff auf den Webserver die Ethernet-Schnittstelle der S7-1200 CPU.

12.1 Webserver aktivieren

Sie aktivieren den Webserver in STEP 7 über die Gerätekonfiguration der CPU, zu der Sie eine Verbindung herstellen möchten.

Um den Webserver zu aktivieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.
2. Wählen Sie im Inspektorfenster unter den CPU-Eigenschaften "Webserver" aus.
3. Aktivieren Sie das Kontrollkästchen für "Webserver auf allen Modulen dieses Geräts aktivieren".
4. Wählen Sie zur verbesserten Sicherheit "Erlaubt nur HTTPS-Zugriff" aus, sodass der sichere Zugriff auf den Webserver erforderlich ist.
5. Wenn Sie für "Automatische Aktualisierung" die Option "Automatische Aktualisierung aktivieren" auswählen, werden die Standard-Webseiten alle zehn Sekunden aktualisiert. Im Feld "Aktualisierungsintervall" können Sie andernfalls auch einen benutzerspezifischen Aktualisierungszeitraum angeben.
 - PLC PROFINET: Stellen Sie sicher, dass die Eigenschaft "Webserver für die IP-Adresse dieser Schnittstelle aktivieren" auf der PLC aktiviert ist, damit der Webserver über die PLC PROFINET-Schnittstelle zugänglich ist.
 - WanCP PROFINET-Port: Wenn Sie ein WanCP-Gerät verwenden, stellen Sie sicher, dass die Eigenschaft "Webserver für die IP-Adresse dieser Schnittstelle aktivieren" aktiviert ist, damit der Webserver über den speziellen WanCP PROFINET-Port zugänglich ist.



**WARNUNG****Nicht berechtigter Zugriff auf die CPU über den Webserver**

Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Schützen Sie die CPU-Zugriffsstufen (Seite 163) und Webserver-Benutzer-IDs (Seite 852) durch starke Passwörter.
- Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden:
 - Großbuchstaben
 - Kleinbuchstaben
 - Ziffern
 - Sonderzeichen
- Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234.
- Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d
- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht.
- Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil Nutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.
- Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN).

Nachdem Sie die Gerätekonfiguration geladen haben, können Sie mit den Standard-Webseiten auf die Einleitungs- und Startseite der CPU zugreifen. Für den Zugriff auf weitere Seiten müssen Sie einen oder mehrere Webserver-Benutzer (Seite 852) einrichten.

Wenn Sie benutzerdefinierte Webseiten (Seite 894) erstellt und aktiviert haben, können Sie über das Navigationsmenü der Standard- bzw. Basis-Webseiten darauf zugreifen.

Hinweis**Geräteaustausch: Ersetzen einer CPU V3.0 durch eine CPU V4.x**

Wenn Sie eine vorhandene CPU V3.0 durch eine CPU V4.x ersetzen (Seite 1456) und Ihr V3.0-Projekt in ein V4.x-Projekt konvertieren, müssen Sie beachten, dass STEP 7 und die CPU V4.x die folgenden Webservereinstellungen beibehalten:

- "Webserver auf allen Modulen dieses Geräts aktivieren"
 - "Erlaubt nur HTTPS-Zugriff"
-

Hinweis

Wenn ein "Laden in RUN" (Seite 1232) durchgeführt wird, werden Datenwerte auf Standard- und benutzerdefinierten Webseiten nicht aktualisiert und Sie können auch erst dann wieder Datenwerte schreiben, wenn der Ladevorgang abgeschlossen ist. Der Webserver bricht jeden Versuch, während des Ladevorgangs Datenwerte zu schreiben, ab.

12.2 Konfigurieren von Webserver-Benutzern

Sie können Benutzer mit verschiedenen Berechtigungsstufen konfigurieren, die über den Webserver Zugriff auf die CPU erhalten.

Um Webserver-Benutzer und die zugehörigen Rechte zu konfigurieren, gehen Sie wie folgt vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.
2. Wählen Sie im Inspektorfenster unter den CPU-Eigenschaften "Webserver" aus und aktivieren Sie den Webserver (Seite 850).
3. Wählen Sie in den Eigenschaften des Webserver "Benutzerverwaltung" aus.
4. Geben Sie Benutzernamen, Zugriffsstufen und Passwörter für die bereitzustellenden Benutzer-Logins ein.

Nachdem Sie die Konfiguration in die CPU geladen haben, können nur berechtigte Benutzer auf die Webserverfunktionen, für die sie die entsprechenden Rechte haben, zugreifen.

Zugriffsstufen für den Webserver

STEP 7 bietet den Standardbenutzer "Jeder" ohne Passwort. Dieser Benutzer hat standardmäßig keine zusätzlichen Rechte und kann nur die Standard-Webseiten Start (Seite 867) und Einführung (Seite 866) öffnen. Sie können jedoch für den Benutzer "Jeder" wie für alle anderen Benutzer zusätzliche Rechte konfigurieren:

- Diagnoseabruf
- Variablen lesen
- Variablen schreiben
- Variablenzustand lesen

- Variablenzustand schreiben
- Benutzerdefinierte Webseiten öffnen
- In benutzerdefinierte Webseiten schreiben
- Dateien lesen
- Dateien schreiben/löschen
- Betriebszustand wechseln
- LEDs blinken lassen
- Firmware-Update durchführen
- Sicherungskopie der CPU erstellen
- CPU wiederherstellen
- Systemparameter ändern
- Anwendungsparameter ändern

Wenn Sie eine benutzerdefinierte Webseite als Einstiegsseite (Seite 913) für den Webserver festgelegt haben, benötigt der Benutzer "Jeder" das Recht "Benutzerdefinierte Webseiten öffnen".

 **WARNUNG**

Unbefugter Zugriff auf die CPU über den Webserver

Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Schützen Sie die CPU-Zugriffsstufen (Seite 163) und Webserver-Benutzer-IDs durch starke Passwörter.
- Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden:
 - Großbuchstaben
 - Kleinbuchstaben
 - Ziffern
 - Sonderzeichen
- Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234.
- Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d
- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht.
- Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil Nutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.
- Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN).

Hinweis

Passwortverschlüsselung für Geräte austausch nach V4.5 (oder höher) aktualisieren

Nach einem Geräte austausch nach V4.5 (oder höher) müssen Sie die Verschlüsselung des Benutzerpassworts für den Webserver aktualisieren. Klicken Sie in der Gerätekonfiguration der CPU im TIA Portal auf die Schaltfläche "Passwortverschlüsselung aktualisieren" in der Webserver-Benutzerverwaltung.

12.3 Über den PC auf die Webseiten zugreifen

Auf die S7-1200 Standard-Webseiten können Sie mit einem PC oder einem mobilen Gerät über die IP-Adresse der S7-1200 CPU oder die IP-Adresse eines über Webserver aktivierten CP (Seite 858) im lokalen Baugruppenträger zugreifen.

Um über einen PC auf die S7-1200 Standard-Webseiten zuzugreifen, gehen Sie folgendermaßen vor:

1. Vergewissern Sie sich, dass sich die S7-1200 und der PC in einem gemeinsamen Ethernet-Netzwerk befinden oder über ein herkömmliches Ethernet-Kabel direkt miteinander verbunden sind.
2. Öffnen Sie einen Webbrowser und geben Sie die URL "https://ww.xx.yy.zz" ein. Dabei entspricht "ww.xx.yy.zz" der IP-Adresse der S7-1200 CPU bzw. der IP-Adresse eines CP im lokalen Baugruppenträger.

Der Webbrowser öffnet die Standard-Webseite "Einführung" (Seite 866) oder die Standard-HTML-Seite Ihrer benutzerdefinierten Webseiten, sofern Sie sie als Einstiegsseite (Seite 913) konfiguriert haben.

Hinweis

Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN). Beachten Sie außerdem alle Einschränkungen (Seite 937), die möglicherweise durch Ihre Webumgebung oder Ihr Betriebssystem gegeben sind.

Zugriff auf Standard-Webseiten durch Eingabe des URL der Seite

Sie können auf eine bestimmte Standard-Webseite über die URL der Seite zugreifen. Geben Sie dafür die URL im Format "https://ww.xx.yy.zz/<Seite>.html" ein, wobei "ww.xx.yy.zz" der IP-Adresse der S7-1200 CPU bzw. der IP-Adresse eines CP im lokalen Baugruppenträger entspricht:

- <https://ww.xx.yy.zz/start.html> - Startseite (Seite 867) mit allgemeinen Informationen über die CPU
- <https://ww.xx.yy.zz/identification.html> - Identifikation (Seite 868) mit ausführlichen Informationen über die CPU einschließlich Serien-, Bestell- und Versionsnummer, jetzt die Diagnose-Seite
- <https://ww.xx.yy.zz/module.html> - Informationen zu den Modulen im lokalen Baugruppenträger und über die Möglichkeit von Firmware-Updates (Seite 872)
- <https://ww.xx.yy.zz/communication.html> - Informationen zur Kommunikation (Seite 876) über die Netzwerkadressen, physikalischen Eigenschaften der Kommunikationsschnittstellen und Kommunikationsstatistik
- <https://ww.xx.yy.zz/diagnostic.html> - Diagnosepuffer (Seite 870)
- <https://ww.xx.yy.zz/variable.html> - CPU-Variablen und E/A (Seite 879), zugänglich über Adressen, PLC-Variablenamen oder Datenbaustein-Variablenamen
- <https://ww.xx.yy.zz/watch.html> - Beobachtungstabellen (Seite 881)
- <https://ww.xx.yy.zz/filebrowser.html> - Browser für den Zugriff auf Datenprotokoll- oder Rezeptdateien (Seite 893), die in der CPU oder auf einer Speicherkarte gespeichert sind

12.4 Über ein mobiles Gerät auf die Webseiten zugreifen

- <https://ww.xx.yy.zz/index.html> -Einführungsseite (Seite 866) für den Einstieg in die Standard-Webseiten
- <https://ww.xx.yy.zz/login.html> - Seite zum Anmelden (Seite 863), sofern gegenwärtig kein Benutzer angemeldet ist; andernfalls ist die Seite leer.

Wenn Sie beispielsweise "<https://ww.xx.yy.zz/communication.html>" eingeben, zeigt der Browser die Kommunikationsseite an.

Hinweis

Beachten Sie, dass jede Standard-Webseite, die nicht speziell oben aufgeführt ist (z. B. die Seite "Online-Sicherung" (Seite 883)) keine URL für direkten Zugriff bietet.

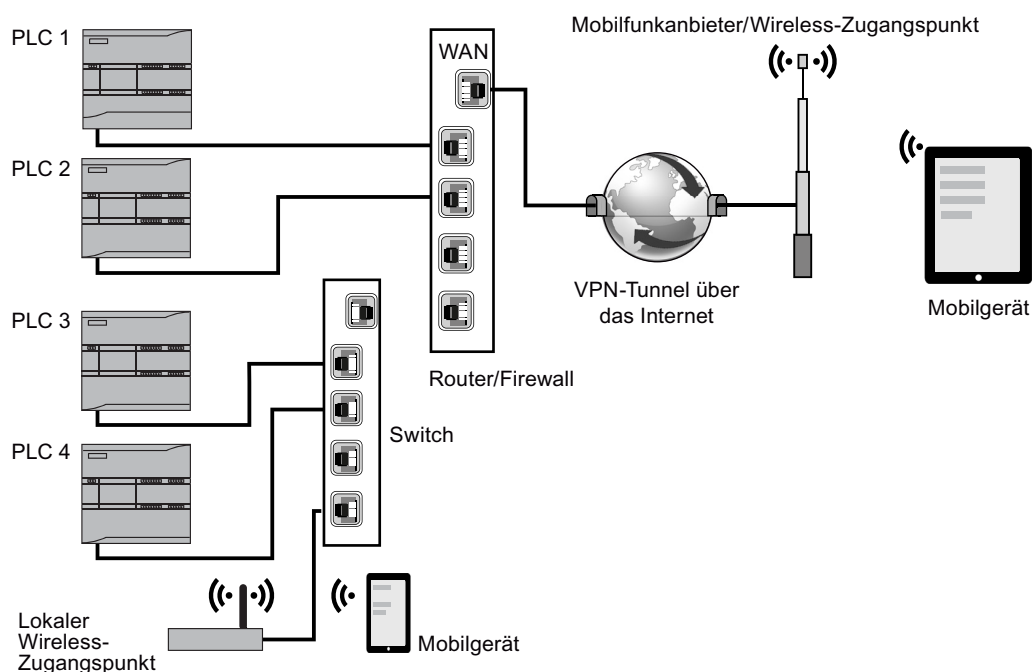
Sicherer Zugriff

Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN). Verwenden Sie statt <http://> für den sicheren Zugriff (Seite 850) auf die Standard-Webseiten <https://> und legen Sie diesen Zugriff als erforderlich fest. Wenn Sie über <https://> eine Verbindung zur S7-1200 herstellen, verschlüsselt die Website die Sitzung mit einem digitalen Zertifikat. Der Webserver überträgt die Daten über eine sichere Verbindung, die für niemanden sichtbar ist. Üblicherweise wird eine Sicherheitswarnung angezeigt, die Sie mit "Ja" bestätigen müssen, um die Standard-Webseiten aufzurufen. Um die Anzeige der Sicherheitswarnung bei jedem sicheren Zugriff zu vermeiden, können Sie das Siemens-Softwarezertifikat in Ihren Webbrowser importieren (Seite 858).

12.4 Über ein mobiles Gerät auf die Webseiten zugreifen

Für den Zugriff auf eine S7-1200 von einem mobilen Gerät aus müssen Sie Ihre PLC an ein Netzwerk mit Internetverbindung oder an einen lokalen Wireless Access Point anschließen. Für den Anschluss eines mobilen Geräts an den S7-1200 PLC-Webserver verwenden Sie ein sicheres VPN (Virtual Private Network). Mit Hilfe der Portweiterleitung im Wireless-Router können Sie die IP-Adresse der PLC einer Adresse zuordnen, durch die ein mobiles Gerät über das Internet auf die PLC zugreifen kann. Zum Einrichten der Portweiterleitung beachten Sie bitte die Anweisungen für die Softwarekonfiguration Ihres Routers. Sie können so viele PLCs und Schaltgeräte anschließen, wie von Ihrem Router unterstützt werden.

Ohne Portweiterleitung ist die Verbindung mit einer PLC nur lokal innerhalb der Reichweite des Funksignals möglich.



In diesem Beispiel kann ein mobiles Gerät, das sich in Reichweite des lokalen Wireless Access Points befindet, die Verbindung zu PLC 3 und PLC 4 über deren IP-Adressen herstellen. Außerhalb der Reichweite des lokalen Funksignals kann ein mobiles Gerät über das Internet die Verbindung zu PLC 1 und PLC 2 mit Hilfe der weitergeleiteten Adresse der jeweiligen PLC herstellen.

Für den Zugriff auf die Standard-Webseiten müssen Sie Zugriff auf einen Mobilfunkdienst oder einem Wireless Access Point haben. Um auf eine PLC über das Internet zuzugreifen, geben Sie in den Webbrowser Ihres mobilen Geräts die weitergeleitete Adresse für den Zugang zur PLC ein, also zum Beispiel `http://ww.xx.yy.zz:pppp` oder `https://ww.xx.yy.zz:pppp`, wobei `ww.xx.yy.zz` die Adresse des Routers und `pppp` die Portzuordnung für eine bestimmte PLC ist.

Für den lokalen Zugriff über einen lokalen Wireless-Zugangspunkt geben Sie die IP-Adresse der S7-1200 CPU oder eines über Webserver aktivierten CP (Seite 858) im lokalen Baugruppenträger ein:

- `http://ww.xx.yy.zz` oder `https://ww.xx.yy.zz` für den Zugriff auf die Standard-Webseiten (Seite 861)
- `http://ww.xx.yy.zz/basic` oder `https://ww.xx.yy.zz/basic` für den Zugriff auf die Basis-Webseiten (Seite 862)

Um die Sicherheit zu erhöhen, konfigurieren Sie den Webserver so, dass nur sicherer Zugriff (HTTPS) (Seite 850) möglich ist.

12.5 Verwenden eines CP-Moduls für den Zugriff auf die Webseiten

Unabhängig davon, ob auf den Webserver über einen PC oder über ein mobiles Gerät zugegriffen wird, können die Standard-Webseiten über eines der folgenden CP-Module geöffnet werden, wenn dieses in STEP 7 konfiguriert und im lokalen Baugruppenträger mit der S7-1200 CPU installiert ist:

- CP 1242-7 GPRS V2
- CP 1243-1
- CP 1243-7 LTE-EU
- CP 1243-7 LTE-US
- CP 1243-8 IRC

Für den Zugriff auf die Webseiten über diese CP-Module wird Standard-Webseite "Start" (Seite 867) verwendet. Die Start-Seite zeigt alle konfigurierten und installierten CP-Module im lokalen Baugruppenträger, der Zugriff auf Webseiten ist jedoch nur über die oben angegebenen Module möglich.

Hinweis

Zugriff auf Standard-Webseiten, wenn über Webserver aktivierte CPs im lokalen Baugruppenträger vorhanden sind

Bei der Verbindung mit den S7-1200 Standard-Webseiten sind bis zu ein oder zwei Minuten lang Verzögerungen möglich, wenn über Webserver aktivierte CPs im lokalen Baugruppenträger installiert sind. Scheinen die Seiten zunächst nicht verfügbar oder werden Fehlermeldungen angezeigt, warten Sie ein oder zwei Minuten und versuchen Sie den Zugriff erneut.

12.6 Herunterladen und Installieren eines Sicherheitszertifikats

Sie können das Siemens-Standard-Sicherheitszertifikat in Ihre Internet Optionen herunterladen.

Mit dem Zertifikat müssen Sie nicht bei jedem Aufruf von <https://ww.xx.yy.zz> (wobei "ww.xx.yy.zz" die IP-Adresse des Geräts ist) in Ihrem Webbrowser die Sicherheitsabfrage beantworten. Wenn Sie eine <http://>-Adresse und keine <https://>-URL eingeben, brauchen Sie das Zertifikat nicht herunterzuladen und zu installieren.

Ab STEP 7 V15 SP1 mit Unterstützung für S7-1200 V4.3 CPUs können Sie in der Gerätekonfiguration einer S7-1200 CPU Zertifikate erstellen. Diese Funktion ist unter der allgemeinen Einstellung "Schutz & Security > Zertifikatsmanager" für das Gerät verfügbar. Anweisungen zum Zertifikatsmanager und dazu, wie Sie globale und lokale CPU-spezifische Zertifikate erstellen, finden Sie im STEP 7-Informationssystem.

Ebenfalls ab STEP 7 V17 mit Unterstützung für S7-1200 V4.5 CPUs können Sie Zertifikate auch für den Webserver einer S7-1200 CPU erzeugen. Sie können die Sicherheit für den Webserver in der Gerätekonfiguration der CPU über "Webserver > Security" konfigurieren.

Hinweis

Begrenzte Anzahl Zertifikate bei der S7-1200

Die S7-1200 hat einen vom System vorgegebenen Grenzwert von 64 Zertifikaten.

In diese Anzahl werden alle Zertifikate eingerechnet (z.B. Webzertifikate, OPC UA-Zertifikate und OUC-Zertifikate).

Wenn der Webserver mit dem Zertifikat einer Zertifikatsstelle (CA) signiert ist und im TIA Portal vorhanden ist, dann werden 2 Zertifikate vom Webserver verwendet (eines für das Webserver-Zertifikat und das andere für das heruntergeladene CA-Zertifikat).

Wenn Sie mehr als 64 Zertifikate haben, zeigt das TIA Portal in einer Fehlermeldung an, dass Sie die maximale Anzahl von 64 Zertifikaten überschritten haben. Sie müssen dann einige Zertifikate aus der PLC-Konfiguration entfernen.

Herunterladen des Zertifikats

Klicken Sie auf der Einführungsseite (Seite 866) auf den Link "Zertifikat herunterladen", um das Siemens-Sicherheitszertifikat auf Ihren PC herunterzuladen. Die Vorgehensweise zum Herunterladen und Importieren unterscheidet sich je nach verwendetem Webbrowser.

Zertifikat im Internet Explorer importieren

1. Klicken Sie auf der Einführungsseite auf den Link "Zertifikat herunterladen".
2. Klicken Sie im folgenden Dialog auf "Öffnen", um die Datei zu öffnen.
3. Klicken Sie im Dialog "Zertifikat" auf die Schaltfläche "Zertifikat installieren", um den Assistenten zum Importieren des Zertifikats aufzurufen.
4. Klicken Sie im Dialog "Assistent für den Zertifikatsimport" auf "Weiter", um den Zertifikatspeicher einzustellen.
5. Wählen Sie "Alle Zertifikate im folgenden Speicher ablegen" und klicken Sie auf die Schaltfläche "Durchsuchen".
6. Im Dialog "Zertifikatspeicher auswählen" wählen Sie "Third-Party Root Certification Authorities" und klicken Sie auf OK.
7. Klicken Sie auf "Weiter" und dann auf "Fertigstellen", um den Assistenten für den Zertifikatsimport zu beenden.

Zertifikat in Mozilla Firefox importieren

1. Klicken Sie auf der Einführungsseite auf den Link "Zertifikat herunterladen".
2. Sobald die Eingabeaufforderung erscheint, klicken Sie auf OK, wenn Sie der S7-1200 Controller Family vertrauen.

12.6 Herunterladen und Installieren eines Sicherheitszertifikats

In älteren Versionen von Mozilla Firefox muss die Datei nach Klick auf "Zertifikat laden" gespeichert und der Assistent ausgeführt werden:

1. Klicken Sie im Dialog mit dem geöffneten Zertifikat auf "Datei speichern". Der Dialog "Downloads" wird angezeigt.
2. Doppelklicken Sie im Dialog "Downloads" auf "MiniWebCA_Cer.crt" oder den Namen des von Ihnen erstellten Zertifikats. Wenn Sie den Download mehrmals gestartet haben, werden mehrere Kopien angezeigt. Doppelklicken Sie auf einen der wiederholten Zertifikatseinträge.
3. Klicken Sie auf "OK", wenn Sie bestätigen sollen, dass eine ausführbare Datei geöffnet wird.
4. Klicken Sie gegebenenfalls im Dialog "Datei öffnen – Sicherheitswarnung" auf "Öffnen". Der Dialog "Zertifikat" wird angezeigt.
5. Klicken Sie im Dialog "Zertifikat" auf die Schaltfläche "Zertifikat installieren".
6. Folgen Sie den Dialogen im "Zertifikatimport-Assistent", um das Zertifikat zu importieren. Dabei wählt das Betriebssystem automatisch den Zertifikatspeicher aus.
7. Wenn der Dialog "Sicherheitswarnung" angezeigt wird, klicken Sie auf "Ja", um die Installation des Zertifikats zu bestätigen.

Andere Browser

Folgen Sie den Konventionen Ihres Webbrowsers, um das Siemens-Zertifikat zu importieren und zu installieren.

Nachdem Sie das Siemens-Sicherheitszertifikat "S7-1200 Controller Family" in den Internetoptionen für die Inhalte Ihres Webbrowsers installiert haben, brauchen Sie keine Sicherheitsabfrage zu bestätigen, wenn Sie über [https:// ww.xx.yy.zz](https://ww.xx.yy.zz) auf den Webserver zugreifen.

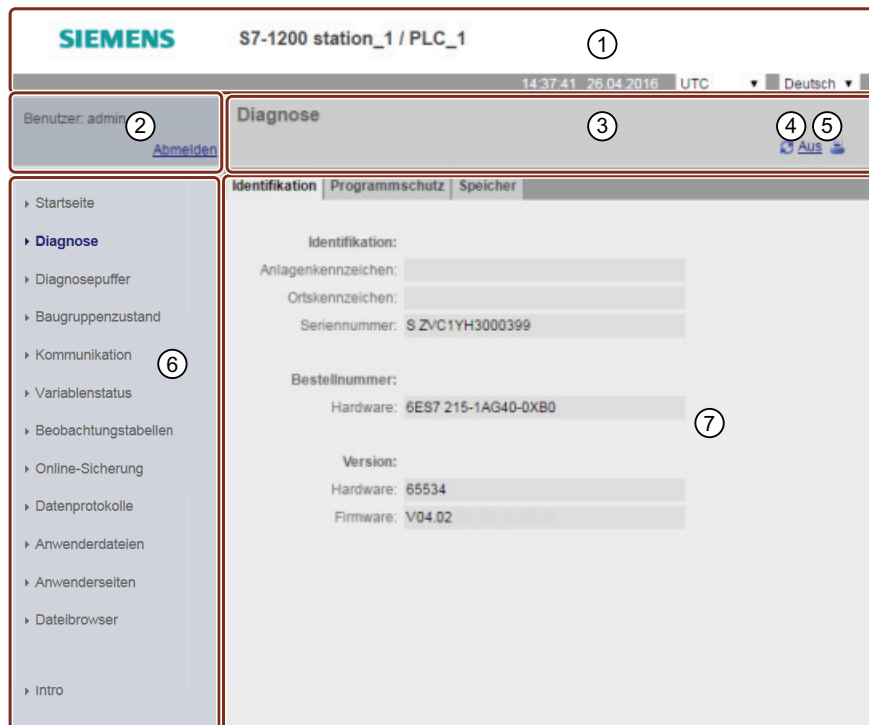
Hinweis

Das Sicherheitszertifikat bleibt auch bei einem Neustart der CPU gültig. Wenn Sie jedoch die IP-Adresse des Geräts ändern, müssen Sie ein neues Zertifikat laden, sofern Sie einen anderen Browser als Internet Explorer oder Mozilla Firefox nutzen.

12.7 Standard-Webseiten

12.7.1 Aufbau der Standard-Webseiten

Alle S7-1200 Standard-Webseiten haben einen gemeinsamen Aufbau und verfügen über Links für die Navigation und Bedienelemente für die Seite. Unabhängig davon, ob Sie die Seite auf einem PC oder einem mobilen Gerät aufrufen, hat jede Seite den gleichen Inhalt. Aufbau und Navigationselemente sind jedoch je nach Bildschirmgröße und Auflösung des Geräts unterschiedlich. Auf einem Standard-PC und großen mobilen Geräten ist eine Standard-Webseite wie folgt aufgebaut:



- ① Webserver-Kopfzeile mit Auswahlmöglichkeiten für PLC-Ortszeit oder UTC-Zeit und die Anzeigesprache (Seite 153)
- ② Anmelden oder Abmelden
- ③ Kopfzeile der Standard-Webseite mit dem Namen der angezeigten Seite. In diesem Beispiel handelt es sich um die Seite "CPU-Diagnose > Identifikation". Auf einigen der Standard-Webseiten, zum Beispiel der Modulinformationsseite, wird hier auch ein Navigationspfad angezeigt, wenn mehrere Seiten dieses Typs aufrufbar sind.
- ④ Symbol zum Aktualisieren: bei Seiten mit automatischer Aktualisierung wird die automatische Aktualisierungsfunktion aktiviert bzw. deaktiviert; bei Seiten ohne automatische Aktualisierung wird die Seite mit den aktuellen Daten aktualisiert
- ⑤ Symbol zum Drucken: erstellt eine druckbare Version der Informationen auf der angezeigten Seite und zeigt sie an

- ⑥ Navigationsbereich zum Wechseln zu einer anderen Seite
- ⑦ Inhaltsbereich der spezifischen Standard-Webseite, die Sie gerade anzeigen. In diesem Beispiel handelt es sich um die Diagnoseseite.

Hinweis

Standard-Webseiten von CP-Modulen

Bestimmte CP-Module (Seite 858) bieten Standard-Webseiten, die vom Aussehen und der Funktionalität her den Standard-Webseiten der S7-1200 CPU ähnlich sind. Eine Beschreibung der Standard-Webseiten Ihres CP finden Sie in der Dokumentation des CP-Moduls.

12.7.2 Basisseiten

Der Webserver bietet Basisseiten zur Verwendung auf Mobilgeräten. Sie rufen die Basisseiten mit Hilfe der IP-Adresse des Geräts auf, indem Sie "basic" an die URL anhängen: `http://ww.xx.yy.zz/basic` oder `https://ww.xx.yy.zz/basic`

Die Basisseiten sehen ähnlich aus wie die Standardseiten, weisen jedoch einige Unterschiede auf. Die Seite hat keinen Navigationsbereich, keinen Anmeldebereich und keinen Kopfbereich, dafür enthält sie Schaltflächen zum Vorwärts- und Rückwärtsblättern durch die Webseiten. Basisseiten haben zudem eine Schaltfläche für die Startseite, mit der Sie auf eine Navigationsseite gelangen. Sie können jedoch auch die Navigationselemente des Mobilgeräts verwenden. So sieht beispielsweise die Basis-Diagnoseseite in der vertikalen Ausrichtung wie folgt aus:

Die Mindestauflösung zum Anzeigen einer Basisseite beträgt 240 x 240 Pixel.



Beachten Sie, dass die Abbildungen der Standard-Webseiten in diesem Kapitel der Darstellung auf einem Standard-PC entsprechen. Die meisten Standard-Webseiten haben entsprechende Basisseiten.

12.7.3 Anmeldung und Benutzerrechte

Auf dem PC bietet jede der Standard-Webseiten ein Anmeldefenster oberhalb des Navigationsbereichs. Aufgrund von Platzbeschränkungen bieten die Basis-Webseiten eine eigene Anmeldeseite. Die S7-1200 unterstützt mehrere Benutzer-Logins mit verschiedenen Zugriffsstufen (Rechten):

- Diagnoseabruf
- Variablen lesen
- Variablen schreiben
- Variablenzustand lesen
- Variablenzustand schreiben
- Benutzerdefinierte Webseiten öffnen
- In benutzerdefinierte Webseiten schreiben
- Dateien lesen
- Dateien schreiben/löschen
- Betriebszustand wechseln
- LEDs blinken lassen
- Firmware-Update durchführen
- Eine Sicherungskopie der PLC erstellen
- Die PLC mittels Backup-Firmware wiederherstellen
- Parameterzugriff (F-Admin) nur für fehlersichere S7-1200 CPUs

In der Gerätekonfiguration der CPU in STEP 7 konfigurieren Sie in den Eigenschaften der Benutzerverwaltung für den Webserver die Benutzerrollen, weisen Zugriffsstufen (Rechte) und Passwörter zu (Seite 852).

Anmelden

STEP 7 bietet den Standardbenutzer "Jeder" ohne Passwort. Dieser Benutzer hat standardmäßig keine zusätzlichen Rechte und kann nur die Standard-Webseiten Start (Seite 867) und Einführung (Seite 866) öffnen. Sie können dem Benutzer "Jeder" jedoch wie allen anderen von Ihnen konfigurierten Benutzern zusätzliche Rechte erteilen:



WARNUNG

Unbefugter Zugriff auf die CPU über den Webserver

Anwender mit vollständigen CPU-Zugriffsrechten oder mit Vollzugriff inkl. Fail-safe können PLC-Variablen lesen und schreiben. Unabhängig von der Zugriffsstufe für die CPU können Webserver-Anwender über Rechte zum Lesen und Schreiben von PLC-Variablen verfügen. Unbefugter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Berechtigte Anwender sind in der Lage, Änderungen des Betriebszustands vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen. Siemens empfiehlt, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Schützen Sie die CPU-Zugriffsstufen (Seite 163) und Webserver-Benutzer-IDs (Seite 852) durch starke Passwörter.
- Starke Passwörter haben mindestens zwölf Zeichen, sind nicht trivial oder leicht zu erraten und enthalten mindestens drei der Folgenden:
 - Großbuchstaben
 - Kleinbuchstaben
 - Ziffern
 - Sonderzeichen
- Ein triviales Passwort ist eines, das leicht zu erraten ist. Üblicherweise beruht es auf etwas Persönlichem des Anwenders, beispielsweise dem Namen des Haustiers, dem eigenen Nachnamen oder dem Namen des Unternehmens, bei dem der Anwender beschäftigt ist. Beispiel: Siemens1\$, Juni2015 oder Qwertz1234.
- Die beste Vorgehensweise zur Erstellung starker, jedoch leicht zu merkender Passwörter ist die Verwendung bedeutungsloser kurzer Sätze und Vermischen einiger zufälliger Wörter. Beispiel: PC;Haus#R3d
- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Erweitern Sie die standardmäßigen Mindestrechte des Webserver-Benutzers "Jeder" nicht.
- Führen Sie eine Fehlerprüfung und eine Bereichsprüfung für die Variablen in Ihrer Programmlogik durch, weil Nutzer von Webseiten für die PLC-Variablen ungültige Werte einrichten können.
- Verwenden Sie für die Verbindung mit dem S7-1200 PLC-Webserver von einem Standort außerhalb Ihres geschützten Netzwerks ein sicheres Virtual Private Network (VPN).

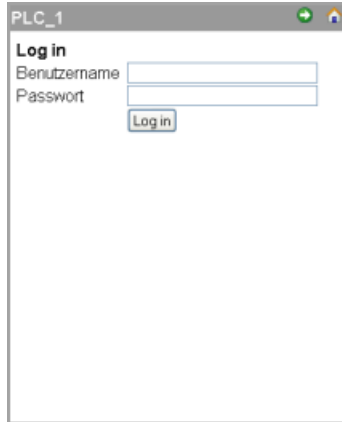
Wenn Sie bestimmte Aktionen durchführen möchten, zum Beispiel die Steuerung in einen anderen Betriebszustand versetzen, Werte in den Speicher schreiben oder die Firmware der CPU aktualisieren, benötigen Sie die dafür erforderlichen Rechte. Wenn Sie für die Schutzstufe der CPU (Seite 163) die Option "Kein Zugriff (kompletter Schutz)" eingerichtet haben, hat der

Benutzer "Jeder" unabhängig von den eingestellten Webserver-Benutzerrechten keine Berechtigung, auf den Webserver zuzugreifen.



Das Feld für die Anmeldung befindet sich bei der Anzeige auf einem PC oder einem Mobilgerät mit großem Display oben links auf jeder Standard-Webseite.

Auf kleinen Mobilgeräten, die die Basisseiten anzeigen, ist die Anmeldeseite eine eigene Seite. Sie kann über die Startseite aufgerufen werden.



Zum Anmelden gehen Sie wie folgt vor:

1. Geben Sie den Benutzernamen in das Feld "Benutzername" ein.
2. Geben Sie das Benutzerpasswort in das Feld "Passwort" ein.

Nach dreißig Minuten Inaktivität werden Sie automatisch abgemeldet. Wenn die angezeigte Seite ständig aktualisiert wird, läuft der Anmeldezeitraum nicht ab und Ihre Sitzung bleibt offen.

Hinweis

Wenn bei der Anmeldung Fehler auftreten, laden Sie das Siemens-Sicherheitszertifikat (Seite 858) von der Einführungsseite (Seite 866) herunter. Danach können Sie sich fehlerfrei anmelden.

Abmelden



Zum Abmelden klicken Sie auf einem PC oder einem Mobilgerät mit großem Display einfach auf einer beliebigen Seite auf den Link "Abmelden".

Navigieren Sie über die Basisseiten von der Startseite zur Seite für die An- und Abmeldung und tippen Sie auf die Schaltfläche "Abmelden".

Nachdem Sie sich abgemeldet haben, können Sie nur die Standard-Webseiten gemäß den Rechten des Benutzers "Jeder" aufrufen und anzeigen. In den Beschreibungen der einzelnen Standard-Webseiten werden die erforderlichen Rechte für die Seite definiert.

Hinweis

Vor dem Schließen des Webrowsers abmelden

Wenn Sie sich am Webserver angemeldet haben, sollten Sie sich vor dem Schließen des Webbrowsers unbedingt abmelden. Der Webserver unterstützt maximal sieben gleichzeitig angemeldete Benutzer.

12.7.4 Einleitung

Die Einführungsseite ist die Begrüßungsseite für den Einstieg in die S7-1200 Standard-Webseiten.



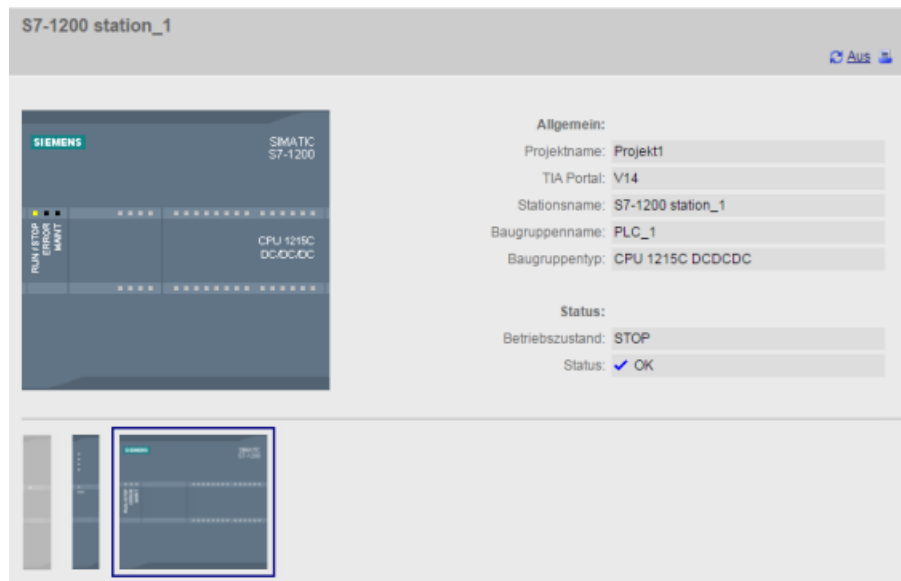
Auf dieser Seite klicken Sie auf "Enter", um die S7-1200 Standard-Webseiten aufzurufen. Oben auf der Seite finden Sie Links zu nützlichen Websites von Siemens sowie einen Link, um das Siemens-Sicherheitszertifikat herunterzuladen (Seite 858). Sie können auch angeben, dass die Einführungsseite bei künftigen Zugriffen auf den Webserver übersprungen werden soll.

12.7.5 Start

Die Startseite zeigt eine Darstellung der CPU oder des CP an, mit der/dem Sie verbunden sind, und führt allgemeine Informationen über das Gerät sowie die Version des TIA Portals auf, mit der Sie das Projekt in die CPU geladen haben. Für die CPU können Sie über die Schaltflächen den Betriebszustand ändern und die LEDs steuern, sofern Sie sich mit dem Recht (Seite 852) "Betriebszustand ändern" angemeldet (Seite 863) haben.

Im unteren Teil des Bildschirms wird angezeigt, ob über Webserver aktivierte CP-Module (Seite 858) im lokalen Baugruppenträger mit der S7-1200 CPU konfiguriert und installiert sind. Für den Zugriff auf die Standard-Webseiten fahren Sie mit dem Mauszeiger über die über Webserver aktivierten CP-Module und klicken Sie auf ein Modul. Informationen über die Webseiten der CP-Module finden Sie in der Dokumentation Ihres CP-Moduls. Der Name des CP-Moduls wird angezeigt, wenn Sie die Maus darüber bewegen.

Der Webserver zeigt ferner weitere CM- und CP-Module im lokalen Baugruppenträger, diese können jedoch nicht angeklickt werden, da sie keine Webseiten enthalten. Diese CM- und CP-Module werden hellgrau dargestellt, um kenntlich zu machen, dass sie lediglich angezeigt werden, aber nicht angeklickt werden können.



Beachten Sie, dass die fehlersicheren S7-1200 CPUs auf dieser Seite zusätzliche Daten in Bezug auf die funktionale Sicherheit anzeigen.

12.7.6 Diagnose

Die Diagnosesseite zeigt identifizierende Eigenschaften der CPU, Konfigurationseinstellungen für Knowhow-Schutz und Speicherauslastung von Ladespeicher, Arbeitsspeicher und remanentem Speicher an:

Die Seite enthält drei Register:

- Identifikation: identifizierende Eigenschaften von Modul und Anlage sowie Informationen zum Standort aus STEP 7
- Programmschutz: Status von Knowhow-Schutz und CPU-Bindung, was bei der Ersatzteilplanung nützlich sein kann, sowie STEP 7-Konfigurationseinstellung zum Zulassen oder Verhindern von Kopien vom internen Ladespeicher zum externen Ladespeicher (SIMATIC Memory Card).
- Speicher: Auslastung von Ladespeicher, Arbeitsspeicher und remanentem Speicher

Bei F-CPUs gibt es ein weiteres Register zur Fehlersicherheit.

Für die Anzeige der Identifikationsseite benötigen Sie das Recht (Seite 852) "Diagnose abfragen".

Register "Identifikation"



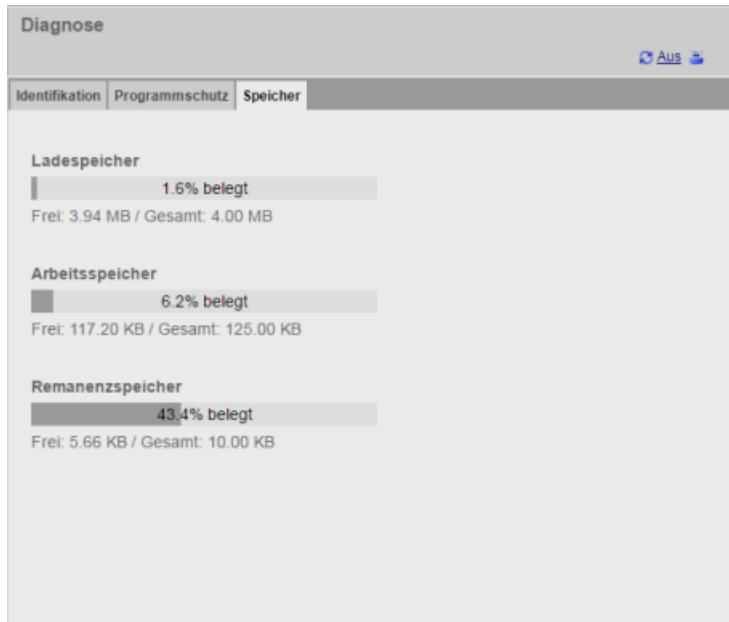
Register "Programmschutz"

Das Register "Programmschutz" enthält die folgenden Informationen:

- Knowhow-Schutz (Seite 167): Zeigt an, ob Sie für die Programmbausteine in STEP 7 Knowhow-Schutz konfiguriert haben.
- Bindung (Seite 168): Zeigt an, ob Sie das Programm entweder an die CPU oder an die SIMATIC Memory Card gebunden haben.
- Programm auf Memory Card kopieren (Seite 167): Zeigt an, ob Sie die Möglichkeit aktiviert haben, das Programm vom internen Ladespeicher in den externen Ladespeicher (SIMATIC Memory Card) zu kopieren.



Register "Speicher"



Register "Fehlersicher"

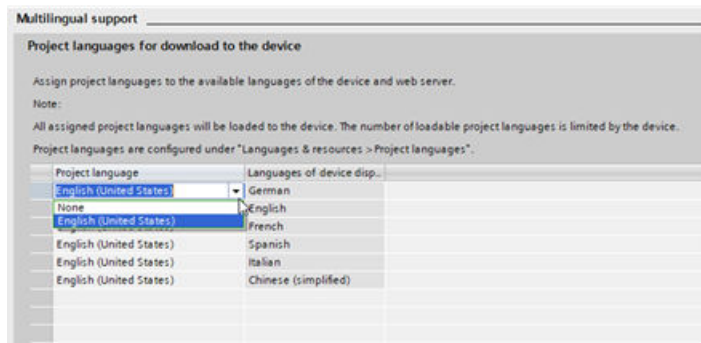
Weitere Informationen zur Diagnosesseite des Registers "Fehlersicher" finden Sie unter S7-1200 Handbuch Funktionale Sicherheit (<https://support.industry.siemens.com/cs/ww/de/view/104547552>).


12.7.7 Diagnosepuffer

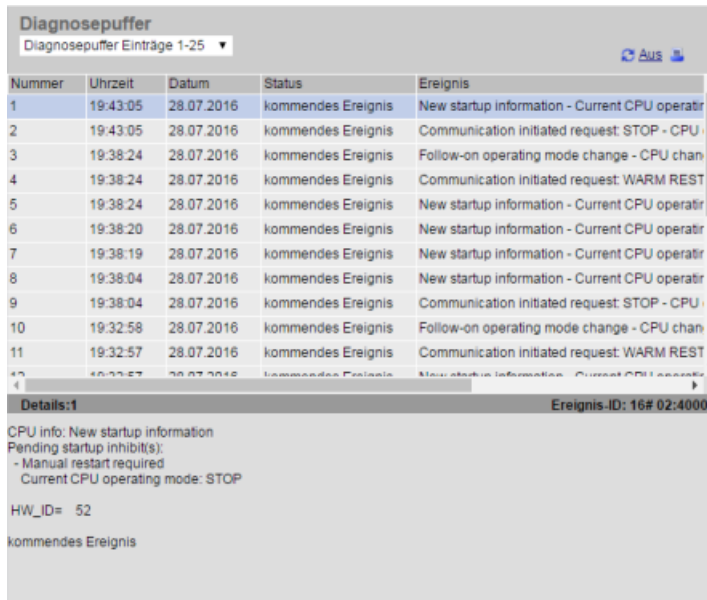
Die Diagnosepufferseite zeigt Diagnoseereignisse an. Das neueste Ereignis wird mit Nummer 1 ganz oben angezeigt. Das älteste Ereignis hat die Nummer 50. Im linken Auswahlfeld können Sie wählen, welcher Bereich von Diagnosepuffereinträgen angezeigt werden soll, entweder die Einträge 1 bis 25 oder die Einträge 26 bis 50. Im rechten Auswahlfeld können Sie wählen, ob die Uhrzeiten in UTC oder lokaler PLC-Zeit angezeigt werden sollen. Im oberen Bereich der Seite werden diese Diagnoseeinträge mit Uhrzeit und Datum des Auftretens angezeigt.

Sie können einen einzelnen Diagnoseeintrag auswählen, um dazu weitere Informationen im unteren Bereich der Seite anzuzeigen. Beachten Sie, dass die Anzeigesprache der Diagnosepuffereinträge von der eingestellten Gerätekonfiguration für mehrsprachigen Support (Seite 158) abhängt.

Sie müssen die Sprachen konfigurieren, die im TIA Portal-Projekt, das in die PLC geladen wird, für Diagnosetext verwendet werden. Diese Konfiguration ist in den PLC-Eigenschaften unter "Mehrsprachiger Support" verfügbar. Jede heruntergeladene Sprache muss einer unterstützten Webserver-Sprache zugewiesen werden. Die PLC ist auf 2 heruntergeladene Sprachen begrenzt.



Die Diagnosepuffer-Seite enthält auch eine Schaltfläche , um den Diagnosepuffer als CSV-Datei zu speichern. Standardmäßig speichert der Webserver die Datei im kommagetrennten Format als ASLog.csv in Ihrem Download-Ordner. Die Datei enthält den vollständigen Diagnosepuffer zum Zeitpunkt des Speichervorgangs. Sie können die Datei so oft speichern, wie Sie möchten, und mehrere Dateien behalten. Sie können die Diagnosepufferdatei mit Microsoft Excel oder einem beliebigen Texteditor öffnen.



Zum Anzeigen der Diagnosepufferseite ist das Recht (Seite 852) für den Diagnoseabruf erforderlich.

12.7.8 Modulinformationen

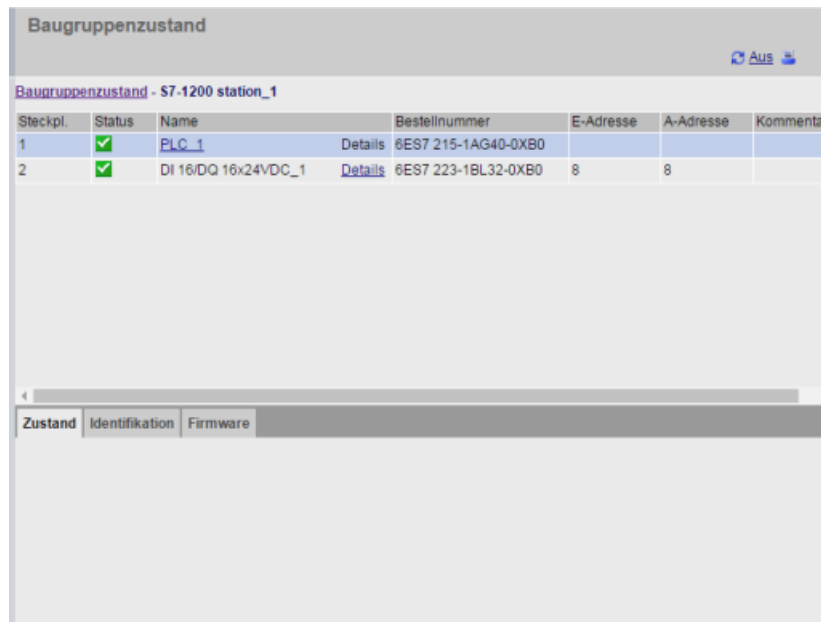
Die Modulinformationsseite:

- Zeigt Informationen zu allen Modulen im lokalen Baugruppenträger an. Im oberen Abschnitt der Seite wird ein Überblick über die Module basierend auf der Gerätekonfiguration in STEP 7 angezeigt, im unteren Abschnitt werden Zustand, Identifikation und Firmware-Informationen des ausgewählten Moduls basierend auf dem entsprechend angeschlossenen Modul angezeigt.
- Bietet die Möglichkeit, eine Firmware-Aktualisierung durchzuführen.
- Informationen über dezentrale Peripheriesysteme

Zum Anzeigen der Modulinformationsseite ist das Recht (Seite 852) für den Diagnoseabruf erforderlich.

Modulinformationen: Register "Status"







Das Register "Status" im unteren Abschnitt der Modulinformationsseite zeigt eine Beschreibung des aktuellen Zustands des Moduls an, das im oberen Abschnitt ausgewählt ist. Wenn der Abschnitt leer ist, hat das Modul keinen ausstehenden Diagnosezustand.



Statussymbole für die Module

Für jedes Modul wird in der Statusspalte im oberen Bereich ein Symbol angezeigt, das den Status des Moduls meldet:

Sym-bol	Bedeutung
✓	Keine Störung
☑	Deaktiviert

Sym- bol	Bedeutung
	Wartung notwendig
	Wartung angefordert
	Fehler
	Die CPU kann das Modul oder Gerät nicht erreichen (bei anderen Geräten als der CPU).
	Die CPU hat eine Verbindung zum Gerät hergestellt, doch der Modulstatus ist unbekannt (bei anderen Geräten als der CPU).
	Daten zu Eingängen und Ausgängen sind nicht verfügbar, weil das Submodul seine E/A-Kanäle gesperrt hat (bei anderen Geräten als der CPU).

Weitere Einzelheiten anzeigen

Sie können im oberen Abschnitt einen Link auswählen, um weitere Einzelheiten der Modulinformationen des spezifischen Moduls anzuzeigen. Module mit Untermodulen haben für jedes Untermodul einen eigenen Link. Die Art der angezeigten Informationen richtet sich nach dem jeweils ausgewählten Modul. Zum Beispiel werden anfänglich in den Modulinformationen der Name der SIMATIC S7-1200 Station, eine Statusanzeige und ein Kommentar angezeigt. Wenn Sie den Link bis zur CPU-Ebene erweitern, zeigen die Modulinformationen die Namen der digitalen und analogen Eingänge und Ausgänge des CPU-Modells sowie Adressierungsinformationen für die E/A, Statusanzeigen, Steckplatznummern und Kommentare an.

Steckpl.	Status	Name	Bestellnummer	E-Adresse	A-Adresse	Kommentar
1.1	✓	DI14/DQ10_1 Details	6ES7 214-1AG40-0XB0	0	0	
1.2	✓	AJ2_1 Details	6ES7 214-1AG40-0XB0	64	---	
1.16	✓	HSC_1 Details	6ES7 214-1AG40-0XB0	1000	---	
1.17	✓	HSC_2 Details	6ES7 214-1AG40-0XB0	1004	---	
1.18	✓	HSC_3 Details	6ES7 214-1AG40-0XB0	1008	---	
1.19	✓	HSC_4 Details	6ES7 214-1AG40-0XB0	1012	---	
1.20	✓	HSC_5 Details	6ES7 214-1AG40-0XB0	1016	---	
1.21	✓	HSC_6 Details	6ES7 214-1AG40-0XB0	1020	---	
1.32	✓	Pulse_1 Details	6ES7 214-1AG40-0XB0	---	1000	

Beim Erweitern der Anzeige gibt die Modulinformationsseite den entsprechenden Pfad an. Sie können auf jeden Link in diesem Pfad klicken, um zu einer höheren Ebene zurückzukehren.



Modulinformationen: Register "Identifikation"

Im Register "Identifikation" werden die Identifikations- und Wartungsdaten (Identification and Maintenance, I&M) des ausgewählten Moduls angezeigt.

Baugruppenzustand Aus

Baugruppenzustand - S7-1200 station_1

Steckpl.	Status	Name	Bestellnummer	E-Adresse	A-Adresse	Kommentar
1	✔	PLC_1	6ES7 215-1AG40-0XB0			
2	✔	DI 16/DQ 16x24VDC_1	6ES7 223-1BL32-0XB0	8	8	

Zustand	Identifikation	Firmware
	Hersteller: Siemens	
	Firmware Version: V4.2	
	Geräteklasse: CPU 1215C DCDCDC	
	Anlagenkennzeichen:	
	Ortskennzeichen:	
	Installationsdatum: 2016-04-25 19:48	
	Beschreibung:	

Wenn Sie im oberen Bereich auf ein F-E/A-Modul klicken, wird im unteren Bereich ein Register "Sicherheit" angezeigt. In diesem Register werden spezifische Daten zum ausgewählten Modul wie unter S7-1200 Handbuch Funktionale Sicherheit (<https://support.industry.siemens.com/cs/ww/de/view/104547552>) beschrieben angezeigt.

Modulinformationen: Register "Firmware"

Im Register "Firmware" der Modulinformationsseite werden Informationen über die Firmware des ausgewählten Moduls angezeigt. Wenn Sie das Recht (Seite 852) zur Durchführung eines Firmware-Updates haben, können Sie auch die Firmware der CPU oder anderer Module im lokalen Baugruppenträger, die Firmware-Updates unterstützen, sowie PROFINET I/O-Module aktualisieren. Bei dezentralen Modulen können Sie die Firmware-Informationen anzeigen, jedoch keine Firmware-Aktualisierung durchführen.

Hinweis

Hinsichtlich der Aktualisierung von CPU-Firmware können Sie nur S7-1200 CPUs ab Version 3.0 aktualisieren.

Status	Identifikation	Firmware
Onlinedaten:		
Bestellnummer:	6ES7 214-1AG40-0XB0	
Firmware:	R 04.00.00_00.13.02.00	
Name:	PLC_1	
Baugruppenträger:	0	
Steckpl.:	1	
Firmware-Loader:		
Firmwaredatei:	<input type="text"/>	<input type="button" value="Browse.."/>
Firmwareversion:	<input type="text"/>	
Geeignet für Module:	<input type="text"/>	
Status:	<input type="text"/>	
<input type="button" value="Aktualisierung ausführen"/>		

Firmware-Update durchführen

Die CPU muss sich im Betriebszustand STOP befinden, um ein Firmware-Update durchzuführen. Befindet sich die CPU im Betriebszustand STOP, klicken Sie auf die Schaltfläche "Durchsuchen", um eine Firmware-Datei auszuwählen. Firmware-Updates stehen auf der Website Siemens Industry Online-Support (<https://support.industry.siemens.com/cs/ww/de>) zur Verfügung.

Während der Aktualisierung wird auf der Seite eine Meldung angezeigt, dass die Aktualisierung gerade durchgeführt wird. Nach Abschluss der Aktualisierung werden auf der Seite die Artikelnummer und die Versionsnummer der aktualisierten Firmware angezeigt. Wenn Sie die Firmware der CPU oder eines Signalboards aktualisiert haben, startet der Webserver die CPU neu.

Ein Firmware-Update können Sie auch auf eine der folgenden Arten durchführen:

- Mit den Online- und Diagnosewerkzeugen von STEP 7 (Seite 1212)
- Mit einer SIMATIC Memory Card (Seite 131)
- Mit dem SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/de/view/98161300>)

Hinweis

Mögliche Probleme bei Durchführung eines Firmware-Updates über den Webserver

Falls es bei Durchführung eines Firmware-Updates über den Webserver zu Kommunikationsstörungen kommt, zeigt der Webbrowser möglicherweise eine Meldung an, in der Sie gefragt werden, ob Sie die aktuelle Seite verlassen möchten oder nicht. Um mögliche Probleme zu vermeiden, wählen Sie die Option aus, auf der aktuellen Seite zu bleiben.

Wenn Sie den Webserver schließen, während ein Firmware-Update über den Webserver durchgeführt wird, können Sie den Betriebszustand der CPU nicht in RUN versetzen. In diesem Fall müssen Sie die CPU aus- und wieder einschalten, um die CPU in den Betriebszustand RUN zu versetzen.

Funktion "Warmstart"

Die Funktion "Warmstart" des Webserver ist als Bestandteil des Firmware-Updates für die S7-1200 CPU ab V4.5 verfügbar. Diese Funktion kann nur während des Firmware-Updates

verwendet werden. Sie können die Funktion "Warmstart" zu keinem anderen Zeitpunkt aufrufen.

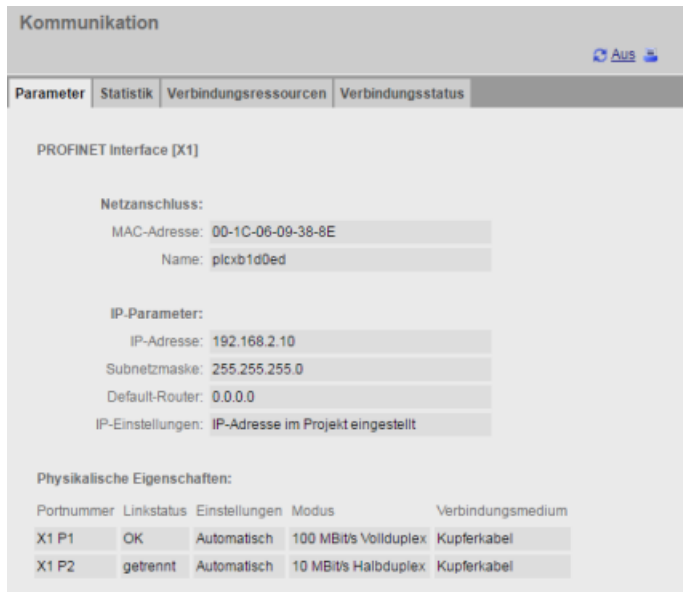
12.7.9 Kommunikation

Die Kommunikationsseite zeigt die Parameter der angeschlossenen CPU, Kommunikationsstatistiken, Ressourcen und Informationen zu Verbindungen an.

Für die Anzeige der Kommunikationsseite benötigen Sie das Recht "Diagnose abfragen".

Register "Parameter"

Das Register "Parameter" zeigt die MAC-Adresse der CPU, die IP-Adresse und die IP-Einstellungen der CPU und die physikalischen Eigenschaften an:



Register "Statistik"

Das Register "Statistik" zeigt eine Statistik der Sende- und Empfangskommunikation an:

Kommunikation	
Parameter	Statistik
Aus	
Gesamtstatistik	
Gesendete Datenpakete:	
Fehlerfrei gesendet:	325254613 Bytes
Kollisionen beim Sendeversuch:	0
Wegen anderer Fehler abgebrochen:	0
Empfangene Datenpakete:	
Fehlerfrei empfangen:	387997542 Bytes
Wegen Fehler abgewiesen:	0
Wegen Ressourcenengpass abgewiesen:	0
X1 P1	
Gesendete Datenpakete:	
Fehlerfrei gesendet:	325254613 Bytes
Kollisionen beim Sendeversuch:	0
Wegen anderer Fehler abgebrochen:	0
Empfangene Datenpakete:	
Fehlerfrei empfangen:	387997542 Bytes
Wegen Fehler abgewiesen:	0
Wegen Ressourcenengpass abgewiesen:	0
X1 P2	
Gesendete Datenpakete:	
Fehlerfrei gesendet:	0 Bytes
Kollisionen beim Sendeversuch:	0
Wegen anderer Fehler abgebrochen:	0
Empfangene Datenpakete:	
Fehlerfrei empfangen:	0 Bytes
Wegen Fehler abgewiesen:	0
Wegen Ressourcenengpass abgewiesen:	0

Register "Verbindungsressourcen"

Das Register "Verbindungsressourcen" zeigt Informationen über die Gesamtanzahl der Verbindungsressourcen an und wie diese den verschiedenen Arten der Kommunikation zugeordnet sind:

Kommunikation																							
Parameter	Statistik	Verbindungsressourcen																					
<p>Anzahl Verbindungen:</p> <p>Maximale Verbindungen: 128</p> <p>Nicht belegte Verbindungen: 126</p>																							
<table border="1"> <thead> <tr> <th>Verbindungen:</th> <th>reserviert</th> <th>belegt</th> </tr> </thead> <tbody> <tr> <td>ES-Kommunikation</td> <td>4</td> <td>1</td> </tr> <tr> <td>HMI-Kommunikation</td> <td>12</td> <td>0</td> </tr> <tr> <td>S7 Kommunikation</td> <td>8</td> <td>0</td> </tr> <tr> <td>OpenUser-Kommunikation</td> <td>8</td> <td>0</td> </tr> <tr> <td>Web Kommunikation</td> <td>0</td> <td>1</td> </tr> <tr> <td>Sonstige Kommunikation</td> <td>---</td> <td>0</td> </tr> </tbody> </table>			Verbindungen:	reserviert	belegt	ES-Kommunikation	4	1	HMI-Kommunikation	12	0	S7 Kommunikation	8	0	OpenUser-Kommunikation	8	0	Web Kommunikation	0	1	Sonstige Kommunikation	---	0
Verbindungen:	reserviert	belegt																					
ES-Kommunikation	4	1																					
HMI-Kommunikation	12	0																					
S7 Kommunikation	8	0																					
OpenUser-Kommunikation	8	0																					
Web Kommunikation	0	1																					
Sonstige Kommunikation	---	0																					

Register "Verbindungsstatus"

Das Register "Verbindungsstatus" zeigt die Verbindungen der CPU und die Verbindungsdetails der ausgewählten Verbindung an.

Kommunikation							
Parameter	Statistik	Verbindungsressourcen	Verbindungsstatus				
Status	Lokale ID (Hex)	Steckplatz von Gateway	Remoter Adresstyp	Remote Adresse	Typ	Art	
✓ Verbindung ist aufgebaut	0	1 (PLC_1)	IPv4	192.168.2.250	ES	Ad hoc	
✓ Verbindung ist aufgebaut	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Ad hoc	
✓ Verbindung ist aufgebaut	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Ad hoc	
✓ Verbindung ist aufgebaut	0	1 (PLC_1)	IPv4	192.168.2.250	WEB	Ad hoc	

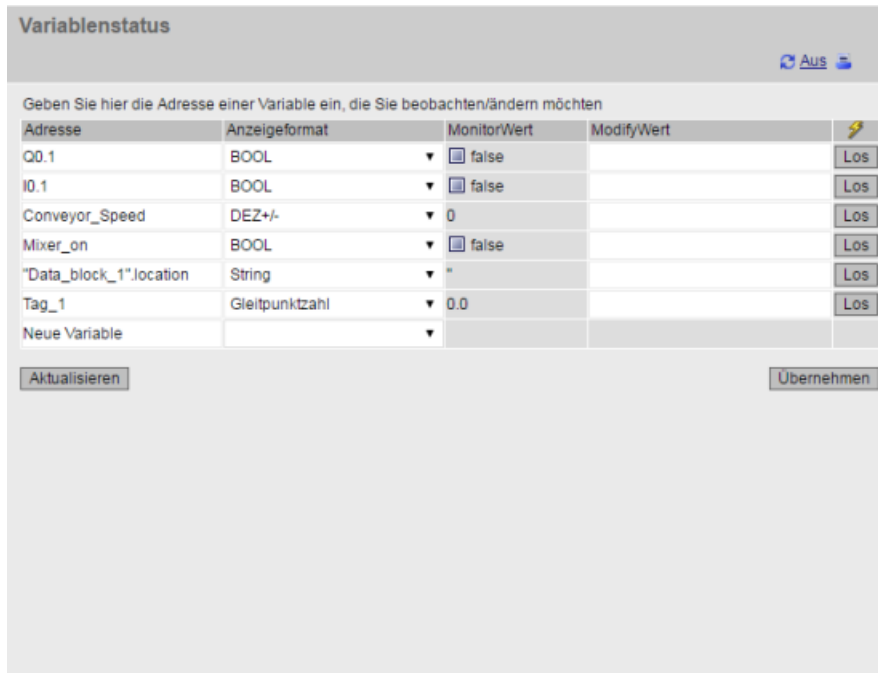
Details:	
Adressdetails	
Lokale Adresse:	192.168.2.10
Lokaler TSAP (Hexadezimal):	53 49 4D 41 54 49 43 2D 52 4F 4F 54 2D 45 53
Lokaler TSAP (ASCII):	SIMATIC-ROOT-ES
Remote Adresse:	192.168.2.250
Remoter TSAP (Hexadezimal):	06 00
Statistik	
Aktuelle Verbindungsaufbauversuche:	0
Erfolgreiche Verbindungsaufbauversuche:	1
Bytes gesendet:	24627
Bytes empfangen:	7058

12.7.10 Variablenstatus

Auf der Seite Variablenstatus können Sie alle E/A oder Speicherdaten in Ihrer CPU anzeigen. Sie können eine direkte Adresse (wie %E0.0), einen PLC-Variablenamen oder eine Variable aus einem spezifischen Datenbaustein eingeben. Bei Datenbausteinvariablen setzen Sie den Bausteinnamen in doppelte Anführungszeichen. Für jeden überwachten Wert können Sie ein Anzeigeformat für die Daten auswählen. Sie können so lange im Rahmen der Seiteneinschränkung Werte eingeben, bis Sie alle gewünschten Werte spezifiziert haben. Die überwachten Werte werden automatisch eingeblendet. Über die Schaltfläche "Aktualisieren" können Sie jederzeit alle überwachten Werte aktualisieren. Wenn Sie in STEP 7 die automatische Aktualisierung aktiviert (Seite 850) haben, können Sie oben rechts auf der Seite auf "Aus" klicken, um diese Einstellung zu deaktivieren. Ist die automatische Aktualisierung deaktiviert, können Sie sie über "Ein" wieder aktivieren.

Zum Anzeigen der Seite Variablenstatus ist das Recht "Variablenstatus lesen" erforderlich.

Wenn Sie sich als Benutzer mit dem Recht (Seite 863) "Variablenstatus schreiben" anmelden, können Sie die Datenwerte auch ändern. Geben im entsprechenden Feld "Steuerwert" die Werte ein, die Sie festlegen möchten. Klicken Sie neben einem Wert auf die Schaltfläche "Los", um den Wert in die CPU zu schreiben. Sie können auch mehrere Werte eingeben und auf "Übernehmen" klicken, um alle Werte in die CPU zu schreiben. Die Schaltflächen und Spaltenbeschriftungen zum Ändern werden nur angezeigt, wenn Sie das Recht "Variablenstatus schreiben" haben.



Wenn Sie die Seite Variablenstatus verlassen und wieder zurückkehren, sind Ihre Eingaben auf der Seite Variablenstatus nicht gespeichert. Sie können die Seite als Lesezeichen speichern und das Lesezeichen erneut aufrufen, damit die gleichen Einträge angezeigt werden. Wenn Sie die Seite nicht als Lesezeichen speichern, müssen Sie die Variablen erneut eingeben.

Bei Werten, die häufig beobachten oder bedienen, sollten Sie in Betracht ziehen, stattdessen eine Beobachtungstabelle (Seite 881) zu verwenden.

Hinweis

Beachten Sie bei der Nutzung der Standardseite für den Variablenstatus Folgendes:

- Setzen Sie alle Zeichenkettenänderungen in einzelne Hochkommata.
 - Die Seite Variablenstatus kann Variablen beobachten und bedienen, die eines der folgenden Zeichen enthält: &, <, (, +, ,(Komma), ., [,], \$ oder %, vorausgesetzt, Sie setzen den Variablennamen in doppelte Anführungszeichen, z. B. "Takt_2,5Hz".
 - Um nur ein Feld einer DTL-Variablen zu beobachten oder zu ändern, nehmen Sie das Feld in die Adresse auf, z. B. "Datenbaustein_1".DTL_Variable.Jahr. Geben Sie für den Steuerwert einen ganzzahligen Wert entsprechend dem Datentyp des spezifischen DTL-Felds ein. Beispiel: Das Jahrfeld hat den Datentyp UInt.
 - Die maximale Anzahl von Variableneinträgen pro Seite beträgt 50.
 - Wenn ein Variablenname Sonderzeichen enthält, so dass der Name als Eintrag auf der Seite Variablenstatus abgelehnt wird, können Sie den Variablennamen in doppelte Anführungszeichen setzen. In den meisten Fällen erkennt dann die Seite den Variablennamen.
-

Siehe auch

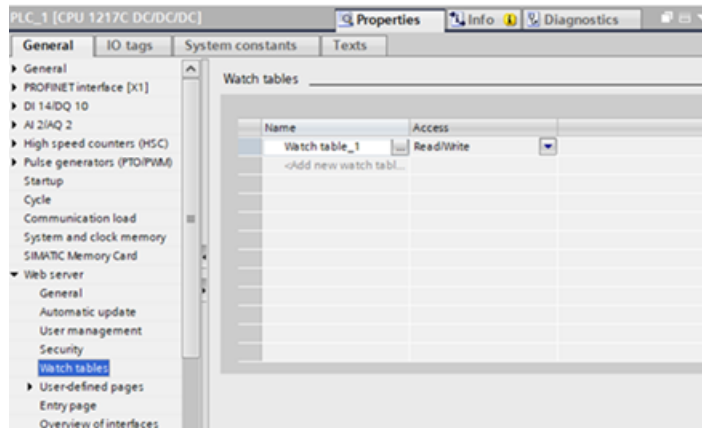
Regeln für die Eingabe von Variablennamen und Werten (Seite 939)

12.7.11 Beobachtungstabellen

Über den Webserver haben Sie Zugriff auf Beobachtungstabellen, die Sie in STEP 7 konfiguriert und in die CPU geladen haben. Beobachtungstabellen mit 50 oder weniger Einträgen bieten die beste Leistung im Webserver.

STEP 7-Konfiguration zur Auswahl von Beobachtungstabellen für den Webserver

In der Gerätekonfiguration der CPU in STEP 7 können Sie Beobachtungstabellen, die vom Webserver angezeigt werden sollen, hinzufügen. Für jede Beobachtungstabelle, die Sie in der Liste der vorhandenen Beobachtungstabellen auswählen, weisen Sie entweder Rechte zum Lesen oder zum Lesen/Schreiben zu. Nach dem Laden in die CPU können Sie die Beobachtungstabellen, denen Sie Rechte zum Lesen zugewiesen haben, nur anzeigen, doch in Beobachtungstabellen, denen Sie das Recht Lesen/Schreiben zugewiesen haben, können Sie Variablen anzeigen und ändern.



Nachdem Sie die Konfiguration der Beobachtungstabelle im Bereich "Webserver" der Gerätekonfiguration beendet haben, laden Sie Ihre Hardwarekonfiguration in die CPU.

Anzeigen von Beobachtungstabellen über den Webserver

Wenn Sie das Recht (Seite 852) "Variablen lesen" haben, können Sie im Navigationsmenü "Beobachtungstabellen" auswählen, um auf die von Ihnen konfigurierten und in die CPU geladenen Beobachtungstabellen zuzugreifen. Wenn Sie mehrere Beobachtungstabellen geladen haben, wählen Sie in der Klappliste die Tabelle aus, die Sie anzeigen möchten. Der Webserver zeigt die Beobachtungstabelle, die Sie in STEP 7 erstellt haben, und die aktuellen Werte entsprechend dem Anzeigeformat an. Sie können das Anzeigeformat nach Belieben ändern, doch wenn Sie auf die Seite der Beobachtungstabellen zurückkehren, verwendet der Webserver standardmäßig wieder die Anzeigeformate in der STEP 7-Beobachtungstabelle.

Ändern von Variablen in Beobachtungstabellen über den Webserver

Wenn Sie eine Beobachtungstabelle mit der Zugriffsstufe "Lesen/Schreiben" geladen haben und sich am Webserver mit dem Recht (Seite 852) "Variablen schreiben" angemeldet haben, können Sie Variablenwerte auch genau wie in einer Beobachtungstabelle in STEP 7 ändern. Sie können einen einzelnen Variablenwert ändern und auf "Los" klicken, um nur einen Wert zu ändern, oder Sie können mehrere Werte eingeben und auf "Übernehmen" klicken, um alle Werte gleichzeitig zu ändern.

Name	Address	Display Format	Monitor Value	Modify Value	Comment
"Data_block_1".Location		String	▼ East		Go
"Data_block_1".ManualOverrideEnable		BOOL	▼ false		Go
"Data_block_1".TurbineNumber		DEC-/-	▼ 2		Go
"Data_block_1".WindSpeed		Floating_Point	▼ 17.5		Go
"Conveyor_speed"	%MW102	DEC-/-	▼ 0		Go

Hinweis

Vorteile von Beobachtungstabellen zum Ändern von Variablen

Damit ein Anwender Variablen und Datenbausteinvariablen in der CPU über eine Beobachtungstabelle ändern kann, müssen Sie die Beobachtungstabelle in den Webservereigenschaften in der STEP 7-Gerätekonfiguration konfigurieren, und Sie müssen der Tabelle Zugriff zum Lesen/Schreiben zuweisen. Dadurch können Sie die Variablen, die ein Anwender mit dem Recht "Variablen schreiben" ändern kann, auf die Variablen in den konfigurierten Webserver-Beobachtungstabellen begrenzen.

Auf der Seite Variablenstatus (Seite 879) kann andererseits jeder Anwender mit dem Recht "Variablenstatus schreiben" in jede beliebige Variable oder Datenbausteinvariablen in der CPU schreiben.

Durch sorgfältige Konfiguration der Rechte im Benutzermanagement (Seite 852) des Webserverns können Sie den Zugriff auf Ihre PLC-Daten schützen.

Siehe auch

Regeln für die Eingabe von Variablennamen und Werten (Seite 939)

12.7.12 Online-Sicherung

Auf der Standard-Webseite Online-Sicherung können Sie eine Sicherungskopie des STEP 7-Projekts für den Online-PLC anlegen sowie eine zuvor angelegte Sicherungskopie des PLC wiederherstellen. Vor dem Erstellen oder Wiederherstellen einer Sicherungskopie versetzen Sie den PLC in den Betriebszustand STOP und beenden sämtliche Kommunikation mit dem PLC wie HMI-Zugriff und Webserverzugriff. Wenn sich Ihre CPU nicht im Betriebszustand STOP befindet, zeigen die Sicherungs- und Wiederherstellungsfunktionen eine Meldung an, in der Sie aufgefordert werden, vor dem Fortfahren die CPU in STOP zu versetzen.

Wenn Sie die Seite "Online-Sicherung" über eines der Web-fähigen CP-Module aufgerufen haben, können Sie eine Sicherung durchführen, jedoch keine Wiederherstellung.

Hinweis

Sicherungs- und Wiederherstellungsfunktionen können Sie auch in STEP 7 (Seite 1247) durchführen. In der vollständigen Beschreibung dieser Themen erfahren Sie, welche Daten Sie sichern und wiederherstellen können. Das SIMATIC Automation Tool (SAT) bietet ebenfalls Sicherungs- und Wiederherstellungsfunktionen.

Wenn Sie Dateien über den Webserver sichern, speichert Ihr PC bzw. Gerät die Sicherungsdateien im Standardordner für Downloads. Wenn Sie Dateien über STEP 7 sichern, speichert STEP 7 die Dateien im STEP 7-Projekt. Sie können STEP 7-Sicherungsdateien nicht über den Webserver wiederherstellen, und Sie können Webserver-Sicherungsdateien nicht über STEP 7 wiederherstellen. Sie können jedoch STEP 7-Sicherungsdateien direkt im Download-Ordner Ihres PCs oder Geräts speichern. Wenn Sie auf diese Weise vorgehen, können Sie diese Dateien über den Webserver wiederherstellen.



PLC sichern

Klicken Sie im Bereich PLC-Sicherung der Seite auf die Schaltfläche "Online-Sicherung erstellen", um von dem Projekt, das gegenwärtig im PLC gespeichert ist, eine Sicherungskopie zu erstellen. Für diese Funktion ist das Recht (Seite 852) "CPU sichern" erforderlich. Wenn sich die CPU im Betriebszustand RUN befindet und Sie sie in den Betriebszustand STOP versetzen müssen, benötigen Sie außerdem das Recht "Betriebszustand ändern". Der PC bzw. das Gerät speichert die Sicherungsdatei im Standardspeicherort für Downloads. Abhängig von Ihrem Browser und den Geräteeinstellungen müssen Sie das Speichern der Datei möglicherweise bestätigen.

PLC wiederherstellen

Geben Sie im Bereich PLC wiederherstellen der Seite das Benutzerpasswort für den Webserver ein und klicken Sie auf die Schaltfläche "Durchsuchen" oder "Datei wählen" (abhängig von Ihrem Browser), um eine zuvor gespeicherte Sicherungsdatei auszuwählen. Klicken Sie auf die Schaltfläche "Online-Sicherung laden" und bestätigen Sie die Meldung zum Laden dieser Datei in den verbundenen PLC. Für diese Seite benötigen Sie das Benutzerrecht (Seite 852) "CPU wiederherstellen". Wenn sich die CPU im Betriebszustand RUN befindet und Sie sie in den Betriebszustand STOP versetzen müssen, benötigen Sie außerdem das Recht "Betriebszustand ändern".

Während des Wiederherstellungsvorgangs werden einige Fortschrittmeldungen angezeigt, und Sie müssen nochmals Ihren Benutzernamen und Ihr Passwort eingeben. Nach jedem erfolgreich abgeschlossenen Schritt werden Ihnen die folgenden Abschlussmeldungen und ein Link zum Neuladen der Seite angezeigt:

Status:

- Das Laden der Online-Sicherung wurde gestartet.
- Die Sicherungsdatei wird ueberprüft.
- Die Speicherkarte wird formatiert und die CPU zurückgesetzt.
- Die Konfiguration wird geladen.
- Die CPU wird zurückgesetzt.

Das Laden der Online-Sicherung wurde erfolgreich abgeschlossen. [Seite neu laden...](#)



WARNUNG

Wiederherstellen von Sicherungskopien mit unbekanntem Inhalt

Wenn Sie eine Sicherungskopie mit unbekanntem Inhalt wiederherstellen, kann dies bei Fehlfunktionen oder Programmfehlern zu schweren Sachschäden oder Verletzungen führen.

Wenn Sie ferner eine Sicherungskopie wiederherstellen, bei der der Webserver in der Gerätekonfiguration der CPU nicht aktiviert ist, können Sie über den Webserver nicht auf die CPU zugreifen.

Achten Sie darauf, dass die Sicherungskopie eine Konfiguration mit bekanntem Inhalt enthält.

Hinweis

Wiederherstellen einer Sicherungskopie, bei der sich die IP-Adresse der CPU unterscheidet

Wenn Sie versuchen, eine Sicherungskopie wiederherzustellen, wobei sich die IP-Adresse der CPU in der Sicherungskopie von der IP-Adresse der aktuellen CPU unterscheidet, kann der Webserver die Meldung, dass die Wiederherstellung beendet ist, nicht anzeigen. Wird die Meldung "CPU wiederherstellen" länger als fünf Minuten angezeigt, geben Sie die neue IP-Adresse ein, die der Adresse in der Sicherungsdatei entspricht. Die CPU hat dann diese Adresse und Sie können den Zugriff auf den Webserver fortsetzen.

12.7.13 Datenprotokollseite

Auf der Seite "Datenprotokolle" können Sie mit Datenprotokollen aus Ihrem STEP 7-Programm interagieren. Sie können:

- eine Liste sämtlicher Datenprotokolle auf Ihrer PLC anzeigen
- ein Datenprotokoll von Ihrer PLC auf Ihren Computer laden
- ein Datenprotokoll von Ihrer PLC löschen
- ein Datenprotokoll abrufen und löschen

Datenprotokolle werden in alphabetisch aufsteigender Reihenfolge mit Beachtung der Groß- und Kleinschreibung angezeigt. Die Liste der Datenprotokolle ist in Seiten mit jeweils 50 Datenprotokollen aufgeteilt.

Name	Size	Changed	Active	Delete	Retrieve and clear
myDataLog.csv	312	06:03:04 pm 2/4/2012	No	X	

Hinweis

Verwaltung von Datenprotokollen

Speichern Sie maximal 1000 Datenprotokolle in einem Dateisystem. Bei Überschreiten dieser Anzahl kann es passieren, dass der Webserver nicht mehr genügend CPU-Ressourcen zur Verfügung hat, um die Datenprotokolle anzuzeigen.

Wenn Sie bemerken, dass auf der Datenprotokoll-Webseite keine Datenprotokolle angezeigt werden können, müssen Sie die CPU in den Betriebszustand STOP versetzen, um Datenprotokolle anzuzeigen und zu löschen.

Stellen Sie über die Verwaltung Ihrer Datenprotolle sicher, dass Sie lediglich die erforderliche Anzahl Datenprotokolle speichern und die maximale Anzahl von 1000 Datenprotokollen nicht überschreiten.

Mit einem Datenprotokoll in Excel arbeiten

Die Protokolldatei steht im amerikanischen/englischen CSV-Format (durch Komma getrennte Werte). Um dieses Format in Excel auf nicht amerikanischen/englischen Systemen zu öffnen, müssen Sie die Datei mit spezifischen Einstellungen nach Excel importieren.

Aktiver Zustand

In der Spalte "Aktiv" auf der Datenprotokollseite wird "Ja" angezeigt, wenn es zu der entsprechenden Datei einen Datenprotokoll-Steuerungsbaustein in der CPU gibt. "Nein" wird angezeigt, wenn dieser Baustein nicht vorhanden ist. Sie können für den aktiven Zustand "Ja" einstellen, auch wenn das STEP 7-Programm aktuell nicht mit dieser Datei interagiert.

Wenn im STEP 7-Programm ein Datenprotokoll geöffnet ist oder in ein Datenprotokoll geschrieben wird, kann der Webserver die Datenprotokolldatei nicht löschen, nicht laden sowie nicht abrufen und löschen. Außerdem können Sie, während der Webserver ein Datenprotokoll entweder über den Befehl zum Laden oder den Befehl zum Abrufen und Löschen lädt, keinen anderen Datenprotokollvorgang durchführen, bis der Ladevorgang beendet ist oder abgebrochen wurde. Der Webserver zeigt die Fehlermeldung "Anwendung beschäftigt" an.

Datenprotokolldatei herunterladen

Sie können eine Datenprotokolldatei durch Klicken auf den Dateinamen herunterladen. Der Webserver zeigt eine Fehlermeldung an, wenn die Datei nicht mehr vorhanden ist oder ein anderer Ladevorgang stattfindet. Die Fehlermeldung wird so lange auf der Seite angezeigt, bis Sie die Datenprotokollseite neu laden. Die folgenden Vorgänge verursachen, dass der Webserver die Datenprotokollseite neu lädt:

- Aktualisieren der Datenprotokollseite oder Verlassen der Datenprotokollseite und Zurückkehren auf die Seite
- Ändern der Einstellung für die Anzahl der auf einer Seite angezeigten Datenprotokolle
- Erfolgreiches Löschen eines Datenprotokolls
- Erfolgreiches Abrufen und Löschen eines Datenprotokolls

Hinweis

Fehlermeldung zu Datenprotokollen



Durch die automatische Aktualisierung des Webserver wird die Fehlermeldung nicht von der Seite entfernt.

Die folgenden Szenarien verursachen Fehlermeldungen auf der Datenprotokollseite:



Vorgang	Fehlerbedingung	Fehlermeldung
Herunterladen Abrufen/Löschen	<ul style="list-style-type: none"> • Datei ist nicht vorhanden • Ungültiger Dateiname • Ungültige HTTP-Anforderungsmethode • SMC ist schreibgeschützt (nur Abrufen/Löschen) 	Fehler beim Herunterladen der Datei
Löschen	<ul style="list-style-type: none"> • Ungültige HTTP-Anforderungsmethode • Ungültiger Dateiname • Datei ist nicht vorhanden 	Fehler beim Löschen der Datei
Löschen	<ul style="list-style-type: none"> • SMC ist schreibgeschützt 	Fehler beim Löschen der Datei: Memory Card ist schreibgeschützt
Herunterladen Löschen Abrufen/Löschen	<ul style="list-style-type: none"> • Erforderliche Berechtigung fehlt • HTTP-Anforderungsmethode ist nicht POST oder GET • Ungültiger ACTION-Parameter in URL 	Dateivorgang nicht zulässig
Herunterladen Löschen Abrufen/Löschen	<ul style="list-style-type: none"> • Ungültiges oder fehlendes HTTP-Referrer-Feld 	Dateivorgang nicht zulässig: kein Referrer

Vorgang	Fehlerbedingung	Fehlermeldung
Herunterladen Löschen	<ul style="list-style-type: none"> Anwenderprogramm hält die Datenprotokolldatei geöffnet 	Anwendung beschäftigt
Abrufen/Löschen	<ul style="list-style-type: none"> Laden eines Datenprotokolls findet gerade statt 	
Herunterladen Löschen Abrufen/Löschen	<ul style="list-style-type: none"> Unerwarteter interner PLC-Fehler 	Interner Fehler

Ein Datenprotokoll löschen

Sie können ein  protokoll löschen, indem Sie in der Spalte "Löschen" eines Datenprotokolls auf das Symbol  klicken. Um die Datenprotokolldatei zu löschen, bestätigen Sie im Dialogfeld "Löschen" den Löschvorgang.

Datenprotokoll abrufen und löschen

Um ein Datenprotokoll  öffnen und alle Einträge zu entfernen, klicken Sie auf das Symbol Abrufen und Löschen . Um die Datenprotokolldatei abzurufen und deren Inhalt zu löschen, bestätigen Sie den Vorgang im Dialogfeld Abrufen und Löschen.

Nachdem Sie den Vorgang bestätigt haben, ermöglicht Ihnen der Webserver das Herunterladen des Inhalts der Datenprotokolldatei. Im Dialogfeld "Datei speichern" können Sie wählen, ob das Datenprotokoll gespeichert werden soll oder nicht. Nach Ihrer Auswahl entfernt der Webserver den Inhalt der Datenprotokolldatei, ohne die Datei zu löschen. Sie können das Löschen des Datenprotokolls im Dialogfeld "Datei speichern" nicht abbrechen. Sie können den Vorgang nur über die anfängliche Bestätigung im Dialogfeld "Abrufen und Löschen" abbrechen.

Wenn während des Vorgangs zum Abrufen und Löschen ein Fehler auftritt, werden im Datenprotokoll keine Daten gelöscht und eine Fehlermeldung angezeigt. Wenn in der Spalte "Aktiv" für das Datenprotokoll "Nein" angezeigt wird, wird beim Vorgang zum Abrufen und Löschen der Inhalt nicht gelöscht, weil das STEP 7-Programm keinen Verweis auf die Datei hat. Sie müssen die Datei dann manuell löschen, um die Daten zu entfernen.

Siehe auch

Webserver (Seite 847)

12.7.14 Anwenderdateien

Anwenderdateien handhaben

Auf der Seite "Anwenderdateien" haben Sie Zugriff auf Dateien auf der Memory Card (externer Ladespeicher).

Ihre Art des Zugriffs auf Anwenderdateien hängt von Ihren Benutzerrechten ab. Alle Benutzer mit Rechten zum Lesen von Dateien können die Dateien und Ordner auf der Seite

"Anwenderdateien" anzeigen. Wenn Sie Rechte zum Lesen und Schreiben von Dateien haben, können Sie außerdem:

- eine Liste sämtlicher Anwenderdateien auf Ihrer PLC anzeigen
- eine Anwenderdatei von Ihrer PLC auf Ihren Computer herunterladen
- eine Anwenderdatei von Ihrem Computer in Ihre PLC hochladen
- eine Anwenderdatei von Ihrer PLC löschen

Um Anwenderdateien anzuzeigen, klicken Sie auf der Hauptnavigationssseite auf "Anwenderdateien".

User Files			
Name	Size	Changed	Delete
50 - Copy (10).html	1.73 KB	03:06:28 pm 01/30/2017	X
50 - Copy (11).html	1.73 KB	03:06:28 pm 01/30/2017	X
50 - Copy (12).html	1.73 KB	03:06:28 pm 01/30/2017	X
50 - Copy (6).html	1.73 KB	03:06:28 pm 01/30/2017	X
50 - Copy (8).html	1.73 KB	03:06:28 pm 01/30/2017	X
50 - Copy (9).html	1.73 KB	03:06:28 pm 01/30/2017	X

Choose File No file chosen Upload file

Die Dateiliste umfasst die aktuelle Größe der Datei und das Datum der letzten Änderung. Die Liste der Dateien wird anhand des Verzeichnisses "Anwenderdateien" im Stammverzeichnis der SMC zusammengestellt.

Hinweis

Anwenderdateiverwaltung

Speichern Sie maximal 1000 Anwenderdateien auf der SMC. Wenn mehr Dateien vorhanden sind, können Sie im Webserver nur die ersten 1.000 Dateien anzeigen. Die Dateien werden in alphabetisch aufsteigender Reihenfolge mit Beachtung der Groß- und Kleinschreibung angezeigt.

Seitenaufteilung in der Liste der Anwenderdateien

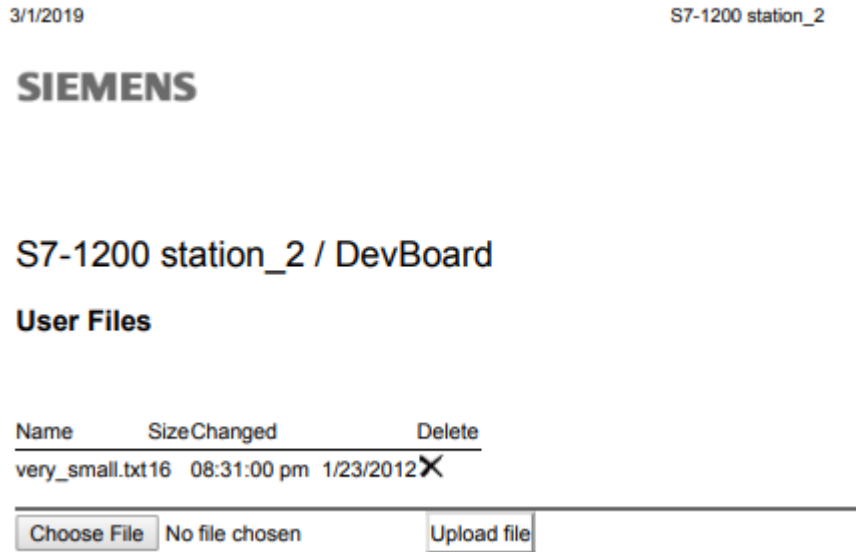
In der Liste der Anwenderdateien werden jeweils 50 Einträge auf einer Seite angezeigt. In einer Klappliste können Sie den Bereich auswählen, den Sie anzeigen möchten.



Name	Size	Changed	Delete
myDataLog - Copy (10).csv	1179	05:28:34 pm 1/11/2012	X
myDataLog - Copy (11).csv	1179	05:28:38 pm 1/11/2012	X
myDataLog - Copy (12).csv	1179	05:28:44 pm 1/11/2012	X
myDataLog - Copy (13).csv	1179	05:28:50 pm 1/11/2012	X

Liste der Anwenderdateien drucken

Sie können die Liste der Anwenderdateien auf Ihrer PLC drucken, indem Sie auf Ihrer PLC-Webseite auf das Symbol "Drucken" klicken:



3/1/2019 S7-1200 station_2

SIEMENS

S7-1200 station_2 / DevBoard

User Files

Name	Size	Changed	Delete
very_small.txt	16	08:31:00 pm 1/23/2012	X

No file chosen

Fehler in Zusammenhang mit Anwenderdateien

Die Seite "Anwenderdateien" zeigt eine Fehlermeldung an, wenn die Ausführung eines Vorgangs fehlschlägt:



Die Fehlermeldung wird so lange auf der Seite angezeigt, bis Sie die Seite "Anwenderdateien" neu laden. Die folgenden Vorgänge verursachen, dass der Webserver die Seite "Anwenderdateien" neu lädt:

- Aktualisieren der Seite "Anwenderdateien" oder Verlassen der Seite "Anwenderdateien" und Zurückkehren auf die Seite
- Ändern der Einstellung für die Anzahl der auf einer Seite angezeigten Anwenderdateien
- Erfolgreiches Löschen einer Anwenderdatei
- Erfolgreiches Hochladen einer Anwenderdatei

Hinweis

Fehlermeldung in Zusammenhang mit Anwenderdateien

Durch die automatische Aktualisierung des Webservers wird eine Fehlermeldung nicht von der Seite "Anwenderdateien" entfernt.

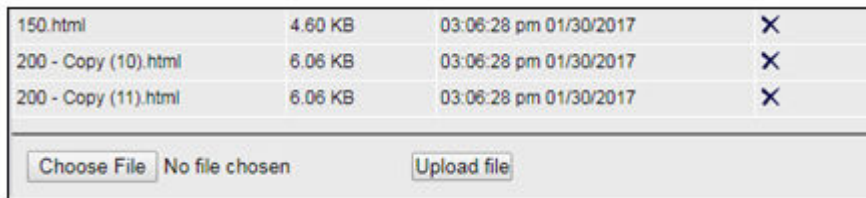
Die folgenden Szenarien verursachen Fehlermeldungen auf der Seite "Anwenderdateien":

Vorgang	Fehlerbedingung	Fehlermeldung
Anwenderdatei herunterladen	<ul style="list-style-type: none"> • Datei ist nicht vorhanden • Interner PLC-Fehler 	Fehler beim Herunterladen der Datei
Anwenderdatei herunterladen Anwenderdatei hochladen Anwenderdatei löschen	<ul style="list-style-type: none"> • Berechtigungen für gewünschten Vorgang nicht ausreichend • Ungültige HTTP-Anforderungsmethode oder ungültiger Aktionsparameter 	Dateivorgang nicht zulässig
Anwenderdatei herunterladen Anwenderdatei hochladen Anwenderdatei löschen	<ul style="list-style-type: none"> • Ungültiges oder fehlendes HTTP-Referrer-Feld 	Dateivorgang nicht zulässig - kein Referrer
Anwenderdatei löschen	<ul style="list-style-type: none"> • Memory Card ist schreibgeschützt 	Dateivorgang nicht zulässig

Vorgang	Fehlerbedingung	Fehlermeldung
Herunterladen Löschen Abrufen/Löschen	<ul style="list-style-type: none"> • Datei ist nicht vorhanden • Interner PLC-Fehler 	Fehler beim Löschen der Datei - Memory Card ist schreibgeschützt
Anwenderdatei hochladen	<ul style="list-style-type: none"> • Dateiname fehlt oder ist ungültig • Interner PLC-Fehler 	Fehler beim Hochladen der Datei
Anwenderdatei hochladen	<ul style="list-style-type: none"> • Dateiname ist auf PLC bereits vorhanden 	Fehler beim Hochladen der Datei - Name ist bereits vorhanden
Anwenderdatei hochladen	<ul style="list-style-type: none"> • SMC ist voll 	Fehler beim Hochladen der Datei - Memory Card ist voll
Anwenderdatei hochladen	<ul style="list-style-type: none"> • Ungültiger Dateiname 	Fehler beim Hochladen der Datei - ungültiges Zeichen im Dateiname
Anwenderdatei hochladen	<ul style="list-style-type: none"> • Datei ist für das SM-Dateisystem zu groß 	Fehler beim Hochladen der Datei – Datei ist zu groß
Anwenderdatei hochladen	<ul style="list-style-type: none"> • SMC ist schreibgeschützt 	Fehler beim Hochladen der Datei - Memory Card ist schreibgeschützt

Eine Anwenderdatei hochladen

Sie können eine Anwenderdatei durch Auswahl einer Datei im Formular zum Hochladen von Anwenderdateien hochladen:



Um eine Anwenderdatei hochzuladen, wählen Sie die Datei aus und klicken auf "Datei hochladen", um die Dateiübertragung von Ihrem Computer in die PLC zu starten. Bevor die Übertragung beginnt, müssen die folgenden Voraussetzungen erfüllt sein:

Voraussetzung	Meldung
Eine Datei muss ausgewählt sein	Bitte geben Sie einen Dateinamen ein
Der Dateiname muss gültig sein	Zulässige Zeichen sind a bis z, 0 bis 9, (){}[]\$!~=~ (Leerzeichen)
Die Datei muss kleiner als 2 GB sein (Einschränkung des Dateisystems).	Fehler beim Hochladen der Datei – Datei ist zu groß

Siehe auch

Webserver (Seite 847)

12.7.15 API für Datenprotokolle und Anwenderdateien

API für Datenprotokolle und Anwenderdateien

Für die S7-1200 gibt es eine API-Funktion für Datenprotokolle und Anwenderdateien. Weitere Informationen finden Sie im *S7-1500, ET200SP, ET200pro Webserver-Handbuch*.

12.7.16 Dateibrowser

Die Dateibrowser-Seite bietet Zugriff auf Dateien im internen Ladespeicher der CPU und auf der Memory Card (externer Ladespeicher). Die Dateibrowser-Seite zeigt zunächst den Stammordner des Ladespeichers an, in dem sich der Ordner "Recipes" und "DataLogs" befindet, doch es werden bei Verwendung einer Memory Card auch alle anderen von Ihnen angelegten Ordner angezeigt.

Ihre Art des Dateizugriffs auf Dateien und Ordner hängt von Ihren Benutzerrechten (Seite 852) ab. Alle Benutzer mit Rechten zum Lesen von Dateien können die Dateien und Ordner im Dateibrowser anzeigen. Unabhängig von Ihren Login-Rechten können Sie den Ordner "Recipes" oder "DataLogs" nicht löschen. Wenn Sie jedoch auf der Memory Card benutzerspezifische Ordner angelegt haben, können Sie diese Ordner löschen, sofern Sie sich mit Rechten zum Schreiben und Löschen von Dateien angemeldet haben.

Klicken Sie auf einen Ordner, um auf die einzelnen Dateien in dem Ordner zuzugreifen.



Rezeptdateien

Im Recipe-Ordner werden alle Rezeptdateien angezeigt, die im Ladespeicher vorhanden sind. Auch die Rezeptdateien sind im CSV-Format und können in Microsoft Excel oder einem anderen Programm geöffnet werden. Sie benötigen Änderungsrechte, um Rezeptdateien zu löschen, zu ändern und zu speichern, umzubenennen oder hochzuladen.

Hochladen von Dateien und automatisches Aktualisieren von Seiten

Wenn Sie mit dem Hochladen einer Datei beginnen, wird der Ladevorgang so lange fortgesetzt, wie Sie auf der Webseite des Dateibrowsers bleiben. Wenn Sie die automatische Aktualisierung der Webserverseiten im Abstand von zehn Sekunden aktiviert haben, sehen Sie den Fortschritt des Datei-Uploads bei jeder Seitenaktualisierung. Wenn Sie beispielsweise eine Datei von 2 MB hochladen, werden Ihnen Aktualisierungen der Dateigröße möglicherweise bei 2500, 5000, 10.000, 15.000 und 20.000 Byte des Ladevorgangs angezeigt.

Wenn Sie die Seite des Dateibrowsers verlassen, bevor der Ladevorgang abgeschlossen ist, löscht der Webserver die unvollständige Datei.

Weitere Informationen

Hinweis

Dateinamenskonventionen

Damit der Webserver mit den Datenprotokoll- und Rezeptdateien arbeiten kann, verwenden Sie Zeichen aus dem ASCII-Zeichensatz. Die Zeichen \ / : * ? " < > | und das Leerzeichen sind nicht zulässig.

Wenn Ihre Dateien mit dieser Benennungskonvention nicht übereinstimmen, kann der Webserver Fehler bei Vorgängen wie z. B. Hochladen, Löschen oder Umbenennen von Dateien haben. Möglicherweise benötigen Sie dann einen Kartenleser und den Windows Datei-Explorer für die Umbenennung von Dateien vom externen Ladespeicher.

Weitere Informationen zur Programmierung mit den Datenprotokollanweisungen und zum Importieren (Seite 489) und Exportieren (Seite 487) von Rezepten finden Sie im Kapitel Rezepte und Datenprotokolle (Seite 482).

Siehe auch

Datenprotokolle im CSV-Format in nicht amerikanische/englische Versionen von Microsoft Excel importieren (Seite 939)

12.8 Benutzerdefinierte Webseiten

Für den S7-1200 Webserver können Sie auch eigene anwendungsspezifische HTML-Seiten mit Daten des Zielsystems anlegen.



WARNUNG

Nicht berechtigter Zugriff auf die CPU über benutzerdefinierte Webseiten

Nicht berechtigter Zugriff auf die CPU über benutzerdefinierte Webseiten kann den Prozessbetrieb unterbrechen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

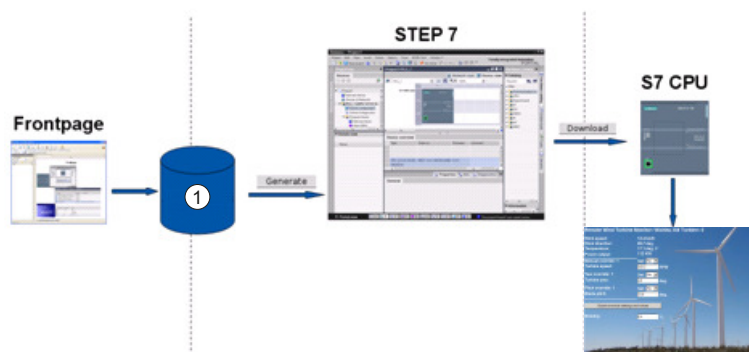
Die nicht sichere Codierung von benutzerdefinierten Webseiten führt zu Sicherheitslücken, über die Angriffe mittels Cross-Site-Scripting (XSS), Code Injection oder andere Methoden möglich sind.

Schützen Sie Ihre S7-1200 CPU durch sichere Installation vor unberechtigtem Zugriff. Informationen dazu finden Sie in den "Operational Guidelines" auf der Industrial Security-Website (<http://www.siemens.com/industrialsecurity>).

Sie erstellen benutzerdefinierte Webseiten mit einem HTML-Editor Ihrer Wahl und laden die Seiten in die CPU, von wo aus sie über das Menü der Standard-Webseiten aufrufbar sind. Hierfür sind mehrere Tätigkeiten durchzuführen:

- HTML-Seiten mit einem HTML-Editor wie Microsoft Frontpage anlegen (Seite 895)
- AWP-Befehle in HTML-Kommentare im HTML-Code einfügen (Seite 896): Bei den AWP-Befehlen handelt es sich um einen fest vorgegebenen Satz Befehle, den Siemens für den Zugriff auf CPU-Informationen bereitstellt.
- STEP 7 zum Lesen und Verarbeiten von HTML-Seiten konfigurieren (Seite 911)
- Bausteine aus den HTML-Seiten generieren (Seite 911)
- STEP 7 für die Steuerung der Verwendung der HTML-Seiten programmieren (Seite 913)
- Bausteine übersetzen und in die CPU laden (Seite 915)
- Über Ihren PC auf die benutzerdefinierten Webseiten zugreifen (Seite 915)

Dieser Prozess wird im Folgenden dargestellt:



① HTML-Dateien mit eingebetteten AWP-Befehlen

Siehe auch

Webserver (Seite 847)

12.8.1 HTML-Seiten anlegen

Sie können das Softwarepaket Ihrer Wahl nutzen, um Ihre eigenen HTML-Seiten für den Webserver anzulegen. Stellen Sie sicher, dass Ihr HTML-Code die HTML-Standards des W3C (World Wide Web Consortium) erfüllt. STEP 7 führt keine Überprüfung Ihrer HTML-Syntax durch.

Sie können eine Software verwenden, bei der Sie im WYSIWYG- oder Design-Layout-Modus programmieren können, doch Sie müssen Ihren HTML-Code auch im reinen HTML-Format bearbeiten können. Die meisten Web-Authoring-Tools bieten diese Art der Bearbeitung. Falls nicht, können Sie den HTML-Code einfach in einem Texteditor bearbeiten. Nehmen Sie die folgende Zeile in Ihre HTML-Seite auf, um UTF-8 als Zeichensatz der Seite festzulegen:

```
<meta http-equiv="content-type" content="text/html; charset=utf-8">
```

Achten Sie auch darauf, dass Sie die Datei im Editor ebenfalls in der UTF-8-Zeichencodierung speichern:

Sie können in STEP 7 die Inhalte Ihrer HTML-Seiten in STEP 7-Datenbausteine übersetzen. Diese Datenbausteine bestehen aus einem Steuerdatenbaustein, der die Anzeige der Webseiten regelt, und einem oder mehreren Datenbausteinfragmenten mit den übersetzten Webseiten. Beachten Sie, dass umfangreiche HTML-Seiten, insbesondere Seiten mit vielen Bildern, einen beträchtlichen Platz im Ladespeicher (Seite 916) für die DB-Fragmente belegen. Wenn der interne Ladespeicher Ihrer CPU für Ihre benutzerdefinierten Webseiten nicht ausreicht, stellen Sie über eine Memory Card (Seite 117) externen Ladespeicher zur Verfügung.

Um Ihren HTML-Code so zu programmieren, dass Daten aus der S7-1200 verwendet werden, können Sie AWP-Befehle (Seite 896) als HTML-Kommentare einfügen. Speichern Sie abschließend Ihre HTML-Seiten auf Ihrem PC und notieren Sie sich den Speicherpfad.

Hinweis

Die Dateigröße von HTML-Dateien mit AWP-Befehlen ist auf 64 kB begrenzt. Die Größe Ihrer Dateien muss unterhalb dieses Grenzwerts bleiben, damit STEP 7 Ihre Seiten richtig übersetzen kann.

Siemens empfiehlt, alle Web-Ressourcendateien (CCC-Dateien, Bilder, JavaScript-Dateien und HTML-Dateien) mit einer Größe von maximal 512 KB zu erstellen. Andernfalls können Probleme beim Senden der Datei vom Webserver zum Browser auftreten. Sie sehen die Größe der jeweiligen Web-Ressourcendatei im Date Explorer des Verzeichnisses.

Benutzerdefinierte Webseiten aktualisieren

Benutzerdefinierte Webseiten werden nicht automatisch aktualisiert. Sie können wählen, ob Sie den HTML-Code so programmieren, dass die Seite aktualisiert wird oder nicht. Bei Seiten, die PLC-Daten anzeigen, bleiben die Daten durch regelmäßige Aktualisierung auf dem aktuellen Stand. Bei HTML-Seiten, die als Formulare zur Dateneingabe dienen, kann eine Aktualisierung die Dateneingabe durch den Benutzer beeinträchtigen. Wenn Ihre gesamte Seite automatisch aktualisiert werden soll, können Sie diese Zeile in Ihre HTML-Kopfzeile aufnehmen, wobei "10" für die Anzahl der Sekunden zwischen zwei Aktualisierungsvorgängen steht:

```
<meta http-equiv="Refresh" content="10">
```

Sie können für die Seiten- und Datenaktualisierung auch JavaScripts oder andere HTML-Techniken nutzen. Hierfür ziehen Sie bitte eine Dokumentation zu HTML oder JavaScripts hinzu.

12.8.2 Vom S7-1200 Webserver unterstützte AWP-Befehle

Der S7-1200 Webserver bietet AWP-Befehle, die Sie zu folgenden Zwecken als HTML-Befehle in Ihre benutzerdefinierten Webseiten einfügen können:

- Variablen lesen (Seite 898)
- Variablen schreiben (Seite 899)
- Sondervariablen lesen (Seite 901)
- Sondervariablen schreiben (Seite 903)

- Enum-Typen definieren (Seite 905)
- Variablen zu Enum-Typen zuweisen (Seite 905)
- Datenbausteinfragmente erstellen (Seite 907)

Allgemeine Syntax

Mit Ausnahme des Befehls zum Lesen einer Variablen haben die AWP-Befehle die folgende Syntax:

```
<!-- AWP_ <Befehlsname und Parameter> -->
```

Sie nutzen die AWP-Befehle in Verbindung mit typischen HTML-Formularbefehlen, um Variablen in die CPU zu schreiben.

In den Beschreibungen der AWP-Befehle auf den folgenden Seiten werden die folgenden Konventionen verwendet:

- In eckigen Klammern [] angegebene Elemente sind optional.
- In spitzen Klammern < > dargestellte Elemente sind anzugebende Parameterwerte.
- Fragezeichen sind ein tatsächlicher Teil des Befehls. Sie müssen wie angezeigt vorhanden sein.
- Sonderzeichen in Variablen- oder Datenbausteinnamen müssen, je nach Verwendung, durch eine Escape-Zeichenfolge gekennzeichnet oder in Anführungszeichen gesetzt werden (Seite 909).

Fügen Sie die AWP-Befehle mit einem Texteditor oder im HTML-Bearbeitungsmodus in Ihre Webseiten ein.

Hinweis

Erwartete Syntax von AWP-Befehlen

Das Leerzeichen nach "<!--" und das Leerzeichen vor "-->" in der Formulierung eines AWP-Befehls sind für die korrekte Übersetzung des Befehls entscheidend. Ein Weglassen der Leerzeichen kann dazu führen, dass die Übersetzung des richtigen Codes nicht möglich ist. In diesem Fall wird keine Fehlermeldung angezeigt.

Überblick über die AWP-Befehle

Genauere Erläuterungen zu den einzelnen AWP-Befehlen finden Sie in den nächsten Abschnitten, hier zunächst ein Überblick über die Befehle:

Variablen lesen

```
:=<Varname>:
```

Variablen schreiben

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Dieser AWP-Befehl deklariert lediglich, dass in die Variable der Namensklausele geschrieben werden kann. Ihr HTML-Code führt namentliche Schreibvorgänge in die Variable aus <input>, <select> oder einer anderen HTML-Anweisung innerhalb eines HTML-Formulars durch.

```

Sondervariablen lesen
<!-- AWP_Out_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->

Sondervariablen schreiben
<!-- AWP_In_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->

Enum-Typen definieren
<!-- AWP_Enum_Def Name='<Name Enum-
Typ>' Values='<Wert>, <Wert>,... ' -->

Enum-Typen definieren
<!-- AWP_In_Variable Name='<Varname>' Enum="<Name Enum-Typ>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<Name Enum-Typ>" -->

Fragmente erstellen
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>] [ID=<ID>] -->

Fragmente importieren
<!-- AWP_Import_Fragment Name='<Name>' -->
    
```

12.8.2.1 Variablen lesen

Benutzerdefinierte Webseiten können Variablen (PLC-Variablen) und Datenbausteinvariablen aus der CPU lesen, vorausgesetzt, Sie haben die Variablen als "Erreichbar aus HMI" konfiguriert.

Syntax

```
:=<Varname>:
```

Parameter

<Varname>	Die zu lesende Variable, bei der es sich um eine PLC-Variable aus Ihrem STEP 7-Programm, eine Datenbausteinvariable, E/A oder eine Adresse im Speicher handeln kann. Setzen Sie bei Speicher- und E/A-Adressen oder Aliasnamen (Seite 909) den Variablennamen nicht in Anführungszeichen. Setzen Sie bei PLC-Variablen den Variablennamen in doppelte Anführungszeichen. Bei Datenbausteinvariablen setzen Sie nur den Bausteinnamen in doppelte Anführungszeichen. Der Variablenname befindet sich außerhalb der Anführungszeichen. Beachten Sie, dass Sie den Namen des Datenbausteins und nicht die Nummer des Datenbausteins verwenden. Für Verweise auf Array-Elemente ist die Syntax für Array-Elemente zu verwenden.
-----------	---

Beispiele

```

:="Fördergeschwindigkeit":
:="Mein_Datenbaustein".Merker 1:
:=I0.0:
:=MW100:
:="Mein_Datenbaustein".Array_Dim1[0]:
:="Mein_Datenbaustein".Array_Dim2[0,0]:
    
```

Beispiel für das Lesen einer über Alias angegebenen Variable

```
<!-- AWP_Out_Variable Name='Merker1'  
Use='Mein_Datenbaustein'.flag1' -->  
:=Merker1:
```

Hinweis

Wie Sie Aliasnamen für PLC-Variablen und Datenbausteinvariablen definieren, wird unter Alias für einen Variablenverweis nutzen (Seite 904) beschrieben.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzlich Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablennamen mit Sonderzeichen (Seite 909) beschrieben.

12.8.2.2 Variablen schreiben

Benutzerdefinierte Webseiten können Daten in die CPU schreiben. Hierfür geben Sie über einen AWP-Befehl eine Variable in der CPU an, in die über die HTML-Seite geschrieben werden soll. Die Variable muss über den PLC-Variablenamen oder den Variablenamen des Datenbausteins angegeben werden. Sie können in einer Anweisung mehrere Variablenamen deklarieren. Um die Daten in die CPU zu schreiben, verwenden Sie den Standard-HTTP-Befehl POST.

Eine typische Verwendung ist die Gestaltung eines Formulars auf Ihrer HTML-Seite mit Texteingabefeldern oder Listenauswahlfeldern, die den schreibbaren CPU-Variablen entsprechen. Wie bei allen benutzerdefinierten Webseiten generieren Sie dann die Bausteine in STEP 7, damit diese in Ihr STEP 7-Programm aufgenommen werden. Wenn ein Benutzer mit Rechten zum Ändern von Variablen anschließend diese Seite aufruft und Daten in die Eingabefelder einträgt oder ein Feld in einer Auswahlliste auswählt, konvertiert der Webserver die Eingaben in den entsprechenden Datentyp für die Variable und schreibt den Wert in die Variable in der CPU. Beachten Sie, dass die Namensklausele für HTML-Eingabefelder und HTML-Auswahllisten eine Syntax verwendet, die für die Namensklausele des Befehls AWP_In_Variable typisch ist. Üblicherweise setzen Sie den Namen in einfache Anführungszeichen und beim Verweis auf einen Datenbaustein setzen Sie den Datenbausteinnamen in doppelte Anführungszeichen.

Weitere Informationen zur Formularverwaltung finden Sie in der HTML-Dokumentation.

Syntax

```
<!-- AWP_In_Variable Name='<Varname1>' [Use='<Varname2>'] ... -->
```

Parameter

<Varname1>	<p>Wenn keine Use-Klausel angegeben ist, ist Varname1 die Variable, in die geschrieben wird. Hierbei kann es sich um eine PLC-Variable aus Ihrem STEP 7-Programm, um eine Variable aus einem spezifischen Datenbaustein oder um einen Datenbausteinnamen handeln.</p> <p>Ist eine Use-Klausel angegeben, ist Varname1 ein alternativer Name für die in <Varname2> referenzierte Variable (Seite 904). Es handelt sich um einen lokalen Namen innerhalb der HTML-Seite.</p>
<Varname2>	<p>Wenn eine Use-Klausel angegeben ist, ist Varname2 die Variable, in die geschrieben wird. Hierbei kann es sich um eine PLC-Variable aus Ihrem STEP 7-Programm oder um eine Variable aus einem spezifischen Datenbaustein handeln.</p>

Sowohl bei Namens- als auch bei Verwendungsklauseln ist der vollständige Name in einfache Anführungszeichen zu setzen. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Der Datenbausteinname befindet sich in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden. Für Verweise auf Array-Elemente ist die Syntax für Array-Elemente zu verwenden.

Wenn Sie einen Datenbaustein mit dem Befehl AWP_In_Variable beschreibbar machen, kann jede Variable in dem Datenbaustein geschrieben werden.

Beispiele mit HTML-Eingabefeldern

```
<!-- AWP_In_Variable Name='Zielstufe' -->
<form method="post">
<p>Eingabe Zielstufe: <input name="Zielstufe" type="text" />
</p>
</form>
```

```
<!-- AWP_In_Variable Name='Datenbaustein_1'.Bremsen' -->
<form method="post">
<p>Bremsen: <input name="Datenbaustein_1'.Bremsen' type="text" />
%</p>
</form>
```

```
<!-- AWP_In_Variable Name='Datenbaustein_1'.Array_Dim2' -->
<form method="post">
<p>Zweidimensionaler Array-Wert [2,1]: <input
name="Datenbaustein_1'.Array_Dim2[2,1]' type="text" /> %</p>
</form>
```

Beispiel für die Verwendungsklausel

```
<!-- AWP_In_Variable Name='Bremsen'
Use='Datenbaustein_1'.Bremsen' -->
<form method="post">
<p>Bremsen: <input name="Bremsen" type="text" /> %</p>
</form>
```

Beispiel mit beschreibbarem Datenbaustein

```
<!-- AWP_In_Variable Name='Datenbaustein_1' -->
<form method="post">
<p>Bremsen: <input name='Datenbaustein_1'.Bremsen' type="text" /> %
</p>
<p>Turbinendrehzahl: <input
name='Datenbaustein_1'.Turbinendrehzahl' size="10"
value='Datenbaustein_1'.Turbinendrehzahl' type="text" />
</p>
</form>
```

Beispiel mit einer HTML-Auswahlliste

```
<!-- AWP_In_Variable
Name='Datenbaustein_1'.FreigabeManuelleÜbersteuerung'-->
<form method="post">
<select name='Datenbaustein_1'.FreigabeManuelleÜbersteuerung'>
<option value="Datenbaustein_1'.FreigabeManuelleÜbersteuerung:'> </
option>
<option value=1>Ja</option>
<option value=0>Nein</option>
</select><input type="submit" value="Einstellung absenden" /></form>
```

Hinweis

Nur ein Benutzer mit dem Recht (Seite 852) "In benutzerdefinierte Webseiten schreiben" darf Daten in die CPU schreiben. Der Webserver ignoriert die Befehle, wenn der Benutzer keine Änderungsrechte hat.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter "Handhabung von Variablenamen mit Sonderzeichen (Seite 909)" beschrieben.

12.8.2.3 Sondervariablen lesen

Der Webserver bietet die Möglichkeit, Werte aus dem PLC-Gerät zu lesen und diese in Sondervariablen in der HTTP-Antwortkopfzeile zu speichern. Sie können z. B. einen Pfadnamen aus einer PLC-Variable auslesen, um die URL über die Sondervariable HEADER:Speicherort zu einem anderen Speicherort umzuleiten.

Syntax

```
<!-- AWP_Out_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->
```

Parameter

<Typ>	Der Typ der Sondervariablen; möglich sind: HEADER COOKIE_VALUE COOKIE_EXPIRES
<Name>	In der HTTP-Dokumentation finden Sie eine Liste aller Namen der HEADER-Variablen. Hier werden einige Beispiele aufgeführt: Status: Antwortcode Location: Pfad für die Umleitung Retry-After: Zeitdauer, über die der Dienst dem anfordernden Client voraussichtlich nicht zur Verfügung steht. Bei den Typen COOKIE_VALUE und COOKIE_EXPIRES, ist <Name> der Name eines bestimmten Cookies. COOKIE_VALUE:name: Wert des genannten Cookies COOKIE_EXPIRES:name: Ablaufzeit in Sekunden des genannten Cookies Die Namensklausele muss in einfache oder doppelte Anführungszeichen gesetzt werden. Ist keine Verwendungsklausel angegeben, entspricht der Sondervariablenname einem PLC-Variablenname. Setzen Sie die vollständige Namensklausele in einfache Anführungszeichen und die PLC-Variable in doppelte Anführungszeichen. Der Sondervariablenname und der PLC-Variablenname müssen sich exakt entsprechen.
<Varname>	Name der PLC-Variablen oder der Datenbausteinvariablen für die auszulesende Variable Der Variablenname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinname in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablenamen mit Sonderzeichen (Seite 909) beschrieben.

Beispiel: Sondervariable ohne Use-Klausel lesen

```
<!-- AWP_Out_Variable Name=' "HEADER:Status" ' -->
```

In diesem Beispiel empfängt die HTTP-Sondervariable "HEADER:Status" den Wert der PLC-Variable "HEADER:Status". Der Name in der PLC-Variablen-tabelle muss dem Namen der Sondervariable exakt entsprechen, wenn keine Use-Klausel angegeben ist.

Beispiel: Sondervariable mit einer Use-Klausel lesen

```
<!-- AWP_Out_Variable Name='HEADER:Status' Use=' "Status" ' -->
```

In diesem Beispiel empfängt die-Sondervariable "HEADER:Status" den Wert der PLC-Variablen "Status".

12.8.2.4 Sondervariablen schreiben

Der Webserver bietet die Möglichkeit, Werte aus Sondervariablen der HTTP-Antwortkopfzeile in die CPU zu schreiben. Sie können beispielsweise in STEP 7 Informationen zu dem Cookie einer benutzerdefinierten Webseite, zu dem Benutzer, der auf eine Seite zugreift, oder Header-Informationen speichern. Der Webserver bietet Zugriff auf bestimmte Sondervariablen, die Sie, wenn Sie als Benutzer mit Rechten zum Ändern von Variablen angemeldet sind, in die CPU schreiben können.

Syntax

```
<!-- AWP_In_Variable Name='<Typ>:<Name>' [Use='<Varname>'] -->
```

Parameter

<Typ>	Der Typ der Sondervariable. Möglich sind: HEADER SERVER COOKIE_VALUE
<Name>	Sondervariable innerhalb der oben definierten Typen, wie in diesen Beispielen gezeigt: HEADER:Accept: akzeptable Inhaltstypene HEADER:User-Agent: Informationen zum Benutzeragenten, von dem die Anforderung stammt. SERVER:current_user_id: ID des aktuellen Benutzers; 0, wenn kein Benutzer angemeldet ist SERVER:current_user_name: Name des aktuellen Benutzers COOKIE_VALUE:<name>: Wert des genannten Cookies Setzen Sie die Namensklausele in einfache Anführungszeichen. Ist keine Verwendungsklausele angegeben, entspricht der Sondervariablenname einem PLC-Variablenamen. Setzen Sie die vollständige Namensklausele in einfache Anführungszeichen und die PLC-Variable in doppelte Anführungszeichen. Der Sondervariablenname muss dem PLC-Variablenamen exakt entsprechen. In der HTTP-Dokumentation finden Sie eine Liste aller Namen der HEADER-Variablen.
<Varname>	Der Variablenname in Ihrem STEP 7-Programm, in das Sie die Sondervariable schreiben möchten. Es kann sich um eine PLC-Variable oder eine Datenbausteinvariable handeln. Der Variablenname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

Beispiele

```
<!-- AWP_In_Variable Name='SERVER:current_user_id' -->
```

In diesem Beispiel schreibt die Webseite den Wert der HTTP-Sondervariablen "SERVER:current_user_id" in die PLC-Variable "SERVER:current_user_id".

```
<!-- AWP_In_Variable Name=SERVER:current_user_id'  
Use='Meine_Benutzer-ID' -->
```

In diesem Beispiel schreibt die Webseite den Wert der HTTP-Sondervariablen "SERVER:current_user_id" in die PLC-Variable "Meine_Benutzer-ID".

Hinweis

Nur ein Benutzer mit Rechten zum Ändern von Variablen kann Daten in die CPU schreiben. Der Webserver ignoriert die Befehle, wenn der Benutzer keine Änderungsrechte hat.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter "Handhabung von Variablennamen mit Sonderzeichen (Seite 909)" beschrieben.

12.8.2.5 Alias für einen Variablenverweis nutzen

Für eine In_Variable oder eine Out_Variable können Sie in Ihren benutzerdefinierten Webseiten einen Alias verwenden. Sie können beispielsweise einen anderen symbolischen Namen in Ihrer HTML-Seite verwenden als den in der CPU, oder Sie können eine Variable in der CPU einer Sondervariablen gleichsetzen. Die AWP-Verwendungsklausel bietet diese Möglichkeit.

Syntax

```
<-- AWP_In_Variable Name='<Varname1>' Use='<Varname2>' -->
<-- AWP_Out_Variable Name='<Varname1>' Use='<Varname2>' -->
```

Parameter

<Varname1>	Der Aliasname oder der Sondervariablenname Varname1 muss in einfachen oder doppelten Anführungszeichen angegeben werden.
<Varname2>	Name der PLC-Variable, der Sie einen Aliasnamen zuweisen möchten. Bei der Variable kann es sich um eine PLC-Variable, eine Datenbausteinvariable oder eine Sondervariable handeln. Varname2 muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable, Sondervariable oder einen Datenbausteinamen in doppelte Anführungszeichen. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden.

Beispiele

```
<-- AWP_In_Variable Name='SERVER:current_user_id'
Use=' "Data_Block_10".server_user' -->
```

In diesem Beispiel wird die Sondervariable SERVER:current_user_id in die Variable "server_user" im Datenbaustein "Data_Block_10" geschrieben.

```
<-- AWP_Out_Variable Name='Weight'
Use=' "Data_Block_10".Tank_data.Weight' -->
```

In diesem Beispiel kann der Wert in Datenbaustein-Strukturelement Data_Block_10.Tank_data.Weight innerhalb der übrigen benutzerdefinierten Webseite einfach als "Weight" referenziert werden.


```
<-- AWP_Out_Variable Name='Gewicht'  
Use='Gewicht_Rohmilchbehälter' -->
```

In diesem Beispiel kann der Wert der PLC-Variable "Gewicht_Rohmilchbehälter" in der übrigen benutzerdefinierten Webseite einfach als "Gewicht" angegeben werden.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzlich Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablennamen mit Sonderzeichen (Seite 909) beschrieben.

12.8.2.6 Enum-Typen definieren

Sie können in Ihren benutzerdefinierten Seiten Enum-Typen definieren und die Elemente in einem AWP-Befehl zuweisen.

Syntax

```
<!-- AWP_Enum_Def Name='<Name Enum-Typ>' Values='<Wert>, <Wert>, ...'  
' -->
```

Parameter

<Name Enum-Typ>	Name des Aufzählungstyps, in einfache oder doppelte Anführungszeichen gesetzt.
<Wert>	<Konstante>:<Name> Die Konstante kennzeichnet den numerischen Wert für die Zuweisung des Enum-Typs. Die Gesamtzahl ist unbegrenzt. Der Name ist der dem Enum-Element zugewiesene Wert.

Beachten Sie, dass die gesamte Zeichenkette der Enum-Wertzuzuweisungen in einzelne Hochkommata eingeschlossen ist, und jede einzelne Elementzuweisung zu einem Enum-Typ steht in doppelten Anführungszeichen. Die Definition eines Enum-Typs hat für die benutzerdefinierten Webseiten globale Gültigkeit. Wenn Sie Ihre benutzerdefinierten Webseiten in sprachspezifischen Ordnern (Seite 928) abgelegt haben, gilt die Enum-Typ-Definition global für alle Seiten in dem Sprachordner.

Beispiel

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",  
1:"Behälter ist voll", 2:"Tank is empty"' -->
```

12.8.2.7 CPU-Variablen mit einem Enum-Typ referenzieren

Sie können eine Variable in der CPU einem Enum-Typ zuweisen. Diese Variable kann an anderer Stelle in Ihren benutzerdefinierten Webseiten in einer Leseoperation (Seite 898) oder einer Schreiboperation (Seite 899) verwendet werden. Bei einer Leseoperation ersetzt der Webserver den aus der CPU gelesenen numerischen Wert durch den entsprechenden Enum-Textwert. Bei einer Schreiboperation ersetzt der Webserver den Textwert mit dem ganzzahligen Wert der Aufzählung, der dem Text entspricht, bevor der Wert in die CPU geschrieben wird.

Syntax

```
<!-- AWP_In_Variable Name='<Varname>' Enum="<Enum-Typ>" -->
<!-- AWP_Out_Variable Name='<Varname>' Enum="<Enum-Typ>" -->
```

Parameter

<Varname>	Name der PLC-Variable oder Datenbausteinvariable, die dem Enum-Typ zugewiesen werden soll, oder Name des Aliasnamens einer PLC-Variable (Seite 904), sofern deklariert. Varname muss in einfache Anführungszeichen gesetzt werden. Innerhalb der einfachen Anführungszeichen setzen Sie eine PLC-Variable sowie einen Datenbausteinnamen in doppelte Anführungszeichen. Beachten Sie, dass Sie für Datenbausteinvariablen den Namen des Bausteins und nicht die Nummer des Datenbausteins verwenden. Der Datenbausteinname steht in doppelten Anführungszeichen, jedoch nicht der Variablenname des Datenbausteins.
<Enum-Typ>	Name des Aufzählungstyps, in einfache oder doppelte Anführungszeichen gesetzt

Die Referenz eines Enum-Typs gilt für das aktuelle Fragment.

Beispiel für die Verwendung beim Lesen einer Variablen

```
<!-- AWP_Out_Variable Name='Alarm' Enum='AlarmEnum' -->...
<p>Der aktuelle Wert von "Alarm" ist := "Alarm":</p>
```

Wenn der Wert von "Alarm" in der CPU gleich 2 ist, zeigt die HTML-Seite 'Der aktuelle Wert von "Alarm" ist Behälter ist leer' an, weil die Definition des Enum-Typs (Seite 905) dem numerischen Wert 2 die Zeichenfolge "Behälter ist leer" zuweist.

Beispiel für die Verwendung beim Schreiben einer Variablen

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Behälter ist leer"' -->
<!-- AWP_In_Variable Name='Alarm' Enum='AlarmEnum' -->...
<form method="POST">
<p><input type="hidden" name="Alarm" value="Behälter ist voll" /></p>
<p><input type="submit" value="Behälter ist voll setzen" /></p>
</form>
```

Weil die Definition des Enum-Typs (Seite 905) dem numerischen Wert 1 den Text "Behälter ist voll" zuweist, wird der Wert 1 in die PLC-Variable "Alarm" in der CPU geschrieben.

Beachten Sie, dass die Enum-Klausel in der Deklaration AWP_In_Variable exakt der Namensklausele in der Deklaration AWP_Enum_Def entsprechen muss.

Beispiel für das Schreiben von Variablen unter Verwendung von Aliasnamen

```
<!-- AWP_Enum_Def Name='AlarmEnum' Values='0:"Keine Alarme",
1:"Behälter ist voll", 2:"Behälter ist leer"' -->
<!-- AWP_In_Variable Name='Alarm' Enum='AlarmEnum'
Use='Data_block_4.Motor1.Alarm' -->...
<form method="POST">
```

```
<p><input type="hidden" name=' "Alarm"' value="Behälter ist voll" /></p>
<p><input type="submit" value='Behälter ist voll setzen' /><p>
</form>
```

Weil die Definition des Enum-Typs (Seite 905) dem numerischen Wert 1 den Text "Behälter ist voll" zuweist, wird der Wert 1 in den Aliasnamen "Alarm" geschrieben, welcher der PLC-Variablen "Motor1.Alarm" im Datenbaustein "Data_Block_4" der CPU entspricht.

Wenn ein Variablenname oder Datenbausteinname Sonderzeichen enthält, müssen Sie zusätzliche Anführungszeichen oder Escape-Zeichen verwenden. Dies wird unter Handhabung von Variablenamen mit Sonderzeichen (Seite 909) beschrieben.

Hinweis

In Vorgängerversionen war eine getrennte AWP_Enum_Ref-Deklaration erforderlich, um einer Variable einen definierten Enum-Typ zuzuordnen. STEP 7 und die S7-1200 unterstützen bestehenden Code mit AWP_Enum_Ref-Deklarationen. Dieser Befehl ist jedoch nicht mehr erforderlich.

12.8.2.8 Fragmente erstellen

STEP 7 konvertiert und speichert benutzerdefinierte Webseiten als Steuerungs-DB und DB-Fragmente, wenn Sie in den CPU-Eigenschaften für den Webserver auf "Bausteine generieren" klicken. Sie können spezifische Fragmente für spezifische Seiten oder für Abschnitte spezifischer Seiten einrichten. Sie können diese Fragmente mit dem AWP-Befehl "Start_Fragment" durch einen Namen und eine Nummer kennzeichnen. Alles auf einer Seite, was auf den Befehl AWP_Start_Fragment folgt, gehört zu dem Fragment, bis ein nachfolgender Befehl AWP_Start_Command abgesetzt wird oder bis das Ende der Datei erreicht ist.

Syntax

```
<!-- AWP_Start_Fragment Name='<Name>' [Type=<Typ>] [ID=<ID>]
[Mode=<Modus>] -->
```

Parameter

<Name>	Text-Zeichenkette: Name des DB-Fragments Fragmentnamen müssen mit einem Buchstaben oder einem Unterstrich beginnen und sich aus Buchstaben, Ziffern und Unterstrichen zusammensetzen. Der Fragmentname ist ein regulärer Ausdruck der Form: [a-zA-Z_][a-zA-Z_0-9]*
<Typ>	"Manuell" oder "Automatisch" Manuell: Das STEP 7-Programm muss dieses Fragment anfordern und kann entsprechend reagieren. Die Funktionsweise des Fragments muss mit STEP 7 und den Variablen des Steuerungs-DBs gesteuert werden. Automatisch: Der Webserver verarbeitet das Fragment automatisch. Wenn Sie den Parameter für den Typ nicht angeben, ist die Voreinstellung "Automatisch".

<ID>	Ganzzahlige Identifikationsnummer. Wenn Sie den Parameter ID nicht angeben, weist der Webserver standardmäßig eine Nummer zu. Geben Sie bei manuellen Fragmenten für die ID eine niedrige Nummer an. Die ID ist der Weg, über den das STEP 7-Programm ein manuelles Fragment steuert.
<Modus>	"sichtbar" oder "ausgeblendet" sichtbar: Inhalte des Fragments werden auf der benutzerdefinierten Webseite angezeigt. ausgeblendet: Inhalte des Fragments werden nicht auf der benutzerdefinierten Webseite angezeigt. Wenn Sie den Parameter für den Typ nicht angeben, ist die Voreinstellung "sichtbar".

Manuelle Fragmente

Wenn Sie ein manuelles Fragment für eine benutzerdefinierte Webseite oder einen Teil einer Seite anlegen, muss Ihr STEP 7-Programm steuern, wann das Fragment gesendet wird. Das STEP 7-Programm muss im Steuer-DB für eine benutzerdefinierte Seite unter manueller Steuerung entsprechende Parameter festlegen und dann die WWW-Anweisung mit dem geänderten Steuer-DB aufrufen. Nähere Erläuterungen zur Struktur des Steuer-DBs und dazu, wie Sie einzelne Seiten und Fragmente ändern, finden Sie unter "Erweiterte Steuerung von benutzerdefinierten Webseiten (Seite 932)".

12.8.2.9 Fragmente importieren

Sie können von einem Teil Ihres HTML-Codes ein benanntes Fragment erstellen und dann dieses Fragment an anderer Stelle in Ihren Satz benutzerdefinierter Webseiten importieren. Stellen Sie sich z. B. einen Satz benutzerdefinierter Webseiten vor, die aus einer Startseite und verschiedenen anderen HTML-Seiten bestehen, die über Links auf der Startseite aufgerufen werden. Angenommen, auf jeder dieser einzelnen Seiten soll das Firmenlogo angezeigt werden. Dafür könnten Sie ein Fragment erstellen (Seite 907), das das Bild des Firmenlogos lädt. Jede einzelne HTML-Seite kann dann dieses Fragment importieren, um das Firmenlogo anzuzeigen. Für diesen Zweck verwenden Sie den AWP-Befehl `Import_Fragment`. Der HTML-Code für das Fragment ist nur in einem Fragment vorhanden, doch Sie können dieses DB-Fragment so oft wie nötig in so viele Webseiten, wie Sie wünschen, importieren.

Syntax

```
<!-- AWP_Import_Fragment Name='<Name>' -->
```

Parameter

<Name>	Text-Zeichenkette: Name des zu importierenden DB-Fragments
--------	--

Beispiel

```
Auszug aus dem HTML-Code, der ein Fragment zum Anzeigen eines Bilds erstellt:
<!-- AWP_Start_Fragment Name='Mein_Firmenlogo' --><p></p>
```

Auszug aus dem HTML-Code in einer anderen *.html-Datei, der das Fragment mit dem Logobild importiert:

```
<!-- AWP_Import_Fragment Name='Mein_Firmenlogo' -->
```

Beide *.html-Dateien (die Datei, die das Fragment erstellt und die Datei, die das Fragment importiert) befinden sich in der Ordnerstruktur, die Sie festlegen, wenn Sie die benutzerdefinierten Seiten in STEP 7 konfigurieren (Seite 911).

12.8.2.10 Definitionen verbinden

Wenn Sie die Variablen für die Verwendung in Ihren benutzerdefinierten Webseiten deklarieren, können Sie eine Variablendeklaration mit einem Alias für die Variable (Seite 904) verbinden. Sie können außerdem mehrere In_Variables in einer Anweisung und mehrere Out_Variables in einer Anweisung verbinden.

Beispiele

```
<!-- AWP_In_Variable Name='"Füllstand"' Name='"Gewicht"'
Name='"Temp"' -->
<!-- AWP_Out_Variable Name='HEADER:Status' Use='"Status"'
Name='HEADER:Location' Use="Speicherort"
Name='COOKIE_VALUE:name' Use="Mein_Cookie" -->
<!-- AWP_In_Variable Name='Alarm' Use='"Datenbaustein_10".Alarm' -->
```

12.8.2.11 Handhabung von Variablennamen mit Sonderzeichen

Bei der Angabe von Variablennamen in benutzerdefinierten Webseiten müssen Sie mit besonderer Vorsicht vorgehen, wenn die Variablennamen Zeichen mit besonderer Bedeutung enthalten.

Variablen lesen

Um eine Variable zu lesen (Seite 898), verwenden Sie die folgende Syntax:

```
:=<Varname>:
```

Die folgenden Regeln gelten beim Lesen von Variablen:

- Bei Variablennamen aus der PLC-Variablen-tabelle setzen Sie den Variablennamen in doppelte Anführungszeichen.
- Bei Variablennamen, bei denen es sich um Datenbausteinvariablen handelt, ist der Datenbausteinname in doppelte Anführungszeichen zu setzen. Die Variable befindet sich außerhalb der Anführungszeichen.
- Bei Variablennamen, bei denen es sich um direkte E/A-Adressen, Speicheradressen oder Aliasnamen handelt, setzen Sie die gelesene Variable nicht in Anführungszeichen.

- Bei Variablenamen oder Datenbausteinvariablenamen, die einen nach links geneigten Schrägstrich ("Backslash") enthalten, stellen Sie dem Backslash einen weiteren Backslash voran.
- Wenn ein Variablenname oder Datenbausteinvariablenname einen Doppelpunkt, ein Kleiner-als-Zeichen, Größer-als-Zeichen oder ein Kaufmanns-Und enthält, definieren Sie für die gelesene Variable einen Alias ohne Sonderzeichen und lesen die Variable über diesen Aliasnamen. Stellen Sie in Variablenamen enthaltenen Doppelpunkten in Verwendungsklauseln einen Backslash voran.

Tabelle 12-1 Beispiele für zu lesende Variablen

Datenbausteinname	Variablenname	Lesebefehl
-/-	ABC:DEF	<code><!--AWP_Out_Variable Name='Sondervariable' Use ='\"ABC:DEF\"' --> :=Sondervariable:</code>
nicht zutreffend	T\	<code>:=\"T\\\":</code>
nicht zutreffend	A \B 'C :D	<code><!--AWP_Out_Variable Name='Weitere_Sondervariable' Use='\"A \\B \\'C :D\"' --> :=Weitere_Sondervariable:</code>
nicht zutreffend	a<b	<code><!--AWP_Out_Variable Name='a_kleiner_b' Use='\"a<b\"' --> :=a_kleiner_b:</code>
Datenbaustein_1	Variable_1	<code>:=\"Datenbaustein_1\".Tag_1:</code>
Datenbaustein_1	ABC:DEF	<code><!-- AWP_Out_Variable Name='Sondervariable' Use='\"Datenbaustein_1\".ABC\ :DEF' --> :=Sondervariable:</code>
DB A' B C D\$ E	Variable	<code>:=\"DB A' B C D\$ E\".Tag:</code>
DB:DB	Variable:Variable	<code><!--AWP_Out_Variable Name='meine_Variable' Use ='\"DB:DB\".Tag \\:Tag' --> :=meine_Variable:</code>

Namens- und Verwendungsklauseln

Die AWP-Befehle `AWP_In_Variable`, `AWP_Out_Variable`, `AWP_Enum_Def`, `AWP_Enum_Ref`, `AWP_Start_Fragment` und `AWP_Import_Fragment` haben NamensklauseIn. HTML-Formularbefehle wie `<input>` und `<select>` haben ebenfalls NamensklauseIn. `AWP_In_Variable` und `AWP_Out_Variable` können zusätzlich Verwendungsklauseln haben. Unabhängig vom Befehl ist die Syntax von Namens- und Verwendungsklauseln hinsichtlich der Handhabung von Sonderzeichen die gleiche:

- Der Text, den Sie für eine Namens- oder Verwendungsklausel angeben, ist in einfache Anführungszeichen zu setzen. Handelt es sich bei dem Namen in Anführungszeichen um eine PLC-Variable oder einen Datenbausteinname, setzen Sie die gesamte Klausel in einfache Anführungszeichen.
- Innerhalb einer Namens- oder Verwendungsklausel sind Datenbausteinname und PLC-Variablenname in doppelte Anführungszeichen zu setzen.
- Wenn ein Variablenname oder Datenbausteinname ein einfaches Anführungszeichen oder einen rückwärts gerichteten Schrägstrich ("Backslash") enthält, stellen Sie dem Zeichen einen Backslash als Escape-Zeichen voran. Der rückwärts gerichtete Schrägstrich "\" dient bei der Übersetzung der AWP-Befehle als Escape-Zeichen.

Tabelle 12-2 Beispiele für NamensklauseIn

Datenbausteinname	Variablenname	Möglichkeiten für NamensklauseIn
-/-	ABC'DEF	<code>Name="ABC\ 'DEF'"</code>
nicht zutreffend	A \B 'C :D	<code>Name="A \\B \'C :D'"</code>
Datenbaustein_1	Variable_1	<code>Name="Datenbaustein_1".Tag_1'</code>
Datenbaustein_1	ABC'DEF	<code>Name="Datenbaustein_1".ABC\ 'DEF'</code>
Datenbaustein_1	A \B 'C :D	<code>Name="Datenbaustein_1".A \\B \'C :D'</code>
DB A' B C D\$ E	Variable	<code>Name="DB A\ ' B C D\$ E".Variable'</code>

Für Verwendungsklauseln gelten die gleichen Konventionen wie für NamensklauseIn.

Hinweis

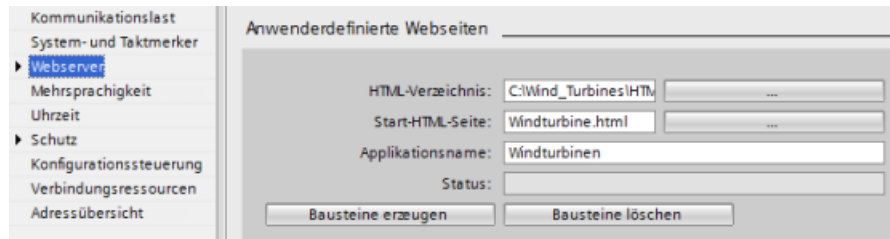
Unabhängig von den Zeichen, die Sie in Ihrer HTML-Seite verwenden, legen Sie als Zeichensatz der HTML-Seite UTF-8 fest und speichern die Seite im Editor mit der Zeichenverschlüsselung UTF-8.

12.8.3 Verwendung von benutzerdefinierten Webseiten konfigurieren

Um benutzerdefinierte Webseiten in STEP 7 zu konfigurieren, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.
2. Rufen Sie im Inspektorfenster der CPU die Eigenschaften des Webserver auf.
3. Sofern nicht bereits geschehen, aktivieren Sie das Kontrollkästchen für "Webserver auf diesem Modul aktivieren".

4. Wählen Sie "Erlaubt nur HTTPS-Zugriff" aus, um sicherzustellen, dass der Webserver die Kommunikation verschlüsselt, und um die Sicherheit Ihrer über das Web zugänglichen CPU zu erhöhen.
5. Geben Sie den Namen des Ordners auf Ihrem PC ein, in dem Sie die HTML-Standardseite (Startseite) gespeichert haben.
6. Geben Sie den Namen der Standardseite ein.
7. Geben Sie einen Namen für Ihre Anwendung ein (optional). Der Webserver verwendet den Anwendungsname zur weiteren Unterteilung bzw. Gruppierung der Webseiten. Wenn Sie einen Anwendungsname eingeben, erstellt der Webserver eine URL für Ihre benutzerdefinierte Seite in folgendem Format: `http[s]://ww.xx.yy.zz/awp/<Anwendungsname>/<Seitenname>.html`. Wird kein Anwendungsname eingegeben, lautet die URL `http[s]://ww.xx.yy.zz/awp/<Seitenname>.html`.
Vermeiden Sie Sonderzeichen im Anwendungsname. Einige Zeichen können verursachen, dass der Webserver die benutzerdefinierten Seiten nicht anzeigen kann.



8. In Abschnitt "Erweitert" geben Sie Namenserverweiterungen von Dateien ein, die AWP-Befehle enthalten. Standardmäßig analysiert STEP 7 Dateien mit den Erweiterungen *.htm, *.html und *.js. Wenn Sie andere Dateierweiterungen nutzen, fügen Sie sie ein. Um Prozessressourcen zu sparen, geben Sie keine Dateierweiterungen ein, wenn keine Dateien dieses Typs AWP-Befehle enthalten.
9. Übernehmen Sie die Standardnummer für den Web-DB oder geben Sie eine Nummer Ihrer Wahl ein. Dies ist die DB-Nummer des Steuer-DBs, der die Anzeige der Webseiten steuert.
10. Übernehmen Sie die standardmäßige Anfangsnummer für das DB-Fragment oder geben Sie eine Nummer Ihrer Wahl ein. Dies ist das erste der DB-Fragmente mit den Webseiten.

Programmbausteine generieren

Wenn Sie auf die Schaltfläche "Bausteine generieren" klicken, generiert STEP 7 aus den HTML-Seiten im von Ihnen angegebenen HTML-Quellverzeichnis Datenbausteine und einen Steuerdatenbaustein für den Betrieb Ihrer Webseiten. Sie können diese Attribute nach Bedarf für Ihre Anwendung festlegen (Seite 913). STEP 7 generiert außerdem einen Satz Datenbausteinfragmente, um die Darstellung aller Ihrer HTML-Seiten zu speichern. Wenn Sie die Datenbausteine generieren, aktualisiert STEP 7 die Eigenschaften, um die Nummer des Steuerdatenbausteins und die Nummer des ersten Datenbausteinfragments anzuzeigen. Nachdem Sie die Datenbausteine generiert haben, sind Ihre benutzerdefinierten Webseiten Teil Ihres STEP 7-Programms. Die diesen Seiten entsprechenden Bausteine erscheinen im Webserverordner, der sich in der Projektnavigation unter den Programmbausteinen im Ordner "Systembausteine" befindet.

Programmbausteine löschen

Um Datenbausteine zu löschen, die Sie zuvor generiert haben, klicken Sie auf die Schaltfläche "Datenbausteine löschen". STEP 7 löscht den Steuerdatenbaustein und alle Datenbausteinfragmente aus Ihrem Projekt, in dem sich die benutzerdefinierten Webseiten befinden.

12.8.4 Konfigurieren der Einstiegsseite

In der Gerätekonfiguration der CPU können Sie eine benutzerdefinierte Webseite als Einstiegsseite für den Zugriff auf den Webserver von einem PC oder einem Mobilgerät zuweisen. Ansonsten wird als Einstiegsseite die Standard-WebseiteEinführung (Seite 866) angezeigt.

Um eine benutzerdefinierte Webseite als Einstiegsseite auszuwählen, gehen Sie wie folgt vor:

1. Wählen Sie die CPU in der Gerätekonfiguration aus.
2. Wählen Sie im Inspektorfenster unter den CPU-Eigenschaften "Webserver" aus und aktivieren Sie den Webserver (Seite 850).
3. Wählen Sie in den Webserver-Eigenschaften "Einstiegsseite" aus.
4. Wählen Sie in der Klappliste "AWP1", um den Webserver so zu konfigurieren, dass beim Zugriff eine benutzerdefinierte Seite angezeigt wird. (Bei der anderen Auswahl "Startseite" zeigt der Webserver beim Aufruf die Standard-Einführungsseite an.)

Sie müssen außerdem für den Benutzer "Jeder" das Recht (Seite 852) "Benutzerdefinierte Webseiten öffnen" konfigurieren und einen Aufruf der Anweisung WWW (Seite 913) in Ihr Programm aufnehmen.

Nachdem Sie die Konfiguration beendet und das Projekt in die CPU geladen haben, kann der Webserver die "Standard-HTML-Seite" verwenden, die Sie beim Konfigurieren Ihrer benutzerdefinierten Webseiten (Seite 911) als Einstiegsseite ausgewählt haben.

Hinweis

Die CPU muss sich im Betriebszustand RUN befinden, um eine benutzerdefinierte Einstiegsseite anzuzeigen.

12.8.5 WWW-Anweisung für benutzerdefinierte Webseiten programmieren

Die WWW-Anweisung muss in Ihrem STEP 7-Anwenderprogramm enthalten sein und ausgeführt werden, damit die benutzerdefinierten Webseiten über die Standard-Webseiten aufrufbar sind. Der Steuerdatenbaustein ist der Eingangsparameter für die WWW-Anweisung und gibt den Inhalt der Seiten wie in den Datenbausteinfragmenten dargestellt sowie Zustands- und Steuerinformationen an. STEP 7 erstellt den Steuerdatenbaustein, wenn Sie in der Konfiguration der benutzerdefinierten Webseiten (Seite 911) auf die Schaltfläche "Bausteine erstellen" klicken.

WWW-Anweisung programmieren

Das STEP 7-Programm muss die Anweisung WWW ausführen, damit die benutzerdefinierten Webseiten über die Standard-Webseiten aufrufbar sind. Sie können festlegen, dass die benutzerdefinierten Webseiten nur unter bestimmten Bedingungen entsprechend den Anwendungsvoraussetzungen und Einstellungen verfügbar sind. Dann kann Ihre Programmlogik steuern, wann die Anweisung WWW aufzurufen ist.

Tabelle 12-3 Anweisung WWW

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := WWW(ctrl_db:=_uint_in_);</pre>	Zugriff auf die benutzerdefinierten Webseiten über die Standard-Webseiten

Sie müssen den Eingangsparameter des Steuerdatenbausteins (CTRL_DB), der der ganzzahligen DB-Nummer des Steuer-DBs entspricht, eingeben. Sie finden diese Bausteinnummer des Steuer-DBs (als Web-DB-Nummer bezeichnet) in den Webserver-Eigenschaften der CPU, nachdem Sie die Bausteine für die benutzerdefinierten Webseiten erstellt haben. Geben Sie die ganzzahlige DB-Nummer als Parameter CTRL_DB der Anweisung WWW an. Der Rückgabewert (RET_VAL) enthält das Funktionsergebnis. Beachten Sie, dass die Anweisung WWW asynchron ausgeführt wird und dass der Ausgang RET_VAL einen Anfangswert von 0 hat, auch wenn später ein Fehler auftreten kann. Das Programm kann den Zustand des Steuer-DBs abfragen, um sicherzustellen, dass die Anwendung erfolgreich gestartet wurde, oder es kann mit einem nachfolgenden Aufruf von WWW den Parameter RET_VAL abfragen.

Tabelle 12-4 Rückgabewert

RET_VAL	Beschreibung
0	Kein Fehler
16#00yx	x: Die vom entsprechenden Bit dargestellte Anforderung ist im Wartezustand: x=1: Anforderung 0 x=2: Anforderung 1 x=4: Anforderung 2 x=8: Anforderung 3 Die x-Werte können logisch durch ODER verknüpft werden, um die Wartezustände mehrerer Anforderungen darzustellen. Wenn z. B. x = 6 ist, sind die Anforderungen 1 und 2 im Wartezustand. y: 0: kein Fehler; 1: Fehler vorhanden und "last_error" wurde im Steuer-DB gesetzt (Seite 932)
16#803a	Der Steuer-DB ist nicht geladen.
16#8081	Datentyp, Format oder Version des Steuer-DBs ist falsch.
16#80C1	Für die Initialisierung der Webanwendung sind keine Ressourcen verfügbar.

Verwendung des Steuer-DBs

STEP 7 erstellt den Steuerdatenbaustein, wenn Sie auf die Schaltfläche "Bausteine generieren" klicken. Die Nummer des Steuer-DBs wird in den Eigenschaften der benutzerdefinierten Webseiten angezeigt. Sie finden den Steuer-DB auch im Ordner "Programmbausteine" in der Projektnavigation.

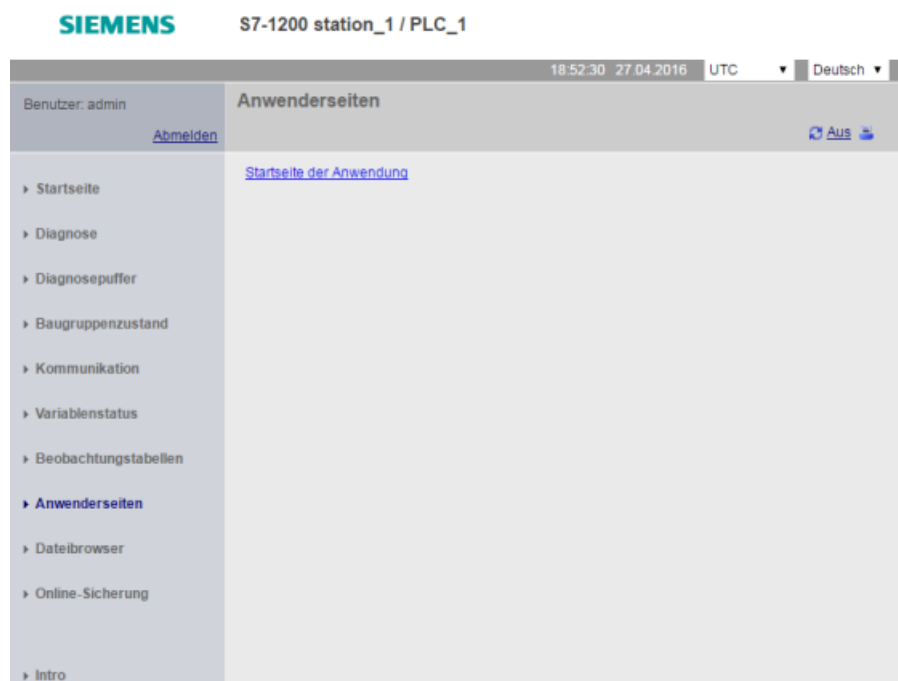
Typischerweise verwendet Ihr STEP 7-Programm den Steuer-DB direkt wie über den Vorgang "Bausteine generieren" angelegt, ohne weitere Änderungen. Das STEP 7-Anwenderprogramm kann jedoch globale Befehle im Steuer-DB festlegen, um den Webserver zu deaktivieren oder ihn anschließend wieder zu aktivieren. Außerdem muss das STEP 7-Anwenderprogramm das Verhalten von benutzerdefinierten Webseiten, die Sie als manuelle DB-Fragmente anlegen (Seite 911), über eine Anforderungstabelle im Steuer-DB steuern. Informationen zu diesen erweiterten Aufgaben finden Sie unter Erweiterte Steuerung von benutzerdefinierten Webseiten (Seite 932).

12.8.6 Programmbausteine in die CPU laden

Nachdem Sie die Bausteine für die benutzerdefinierten Webseiten generiert haben, sind diese Teil Ihres STEP 7-Programms, genau wie alle anderen Programmbausteine. Um die Programmbausteine in die CPU zu laden, gehen Sie wie üblich vor. Beachten Sie, dass Sie Programmbausteine für benutzerdefinierte Webseiten nur in die CPU laden können, wenn sich diese im Betriebszustand STOP befindet.

12.8.7 Zugriff auf die benutzerdefinierten Webseiten

Sie rufen die benutzerdefinierten Webseiten über die Standard-Webseiten (Seite 855) auf. Die Standard-Webseiten zeigen im Navigationsmenü auf der linken Seite einen Link "Anwenderdefinierte Seiten" an. Die Navigation auf den Basisseiten zeigt auch einen Link "Anwenderdefinierte Seiten" an. Wenn Sie auf den Link "Anwenderdefinierte Seiten" klicken, ruft Ihr Webbrowser die Seite auf, die einen Link auf Ihre Standardseite bietet. Innerhalb der benutzerdefinierten Seiten entspricht die Navigation dem Aufbau Ihrer spezifischen Seiten.



Hinweis

Sie können für den Webserver auch eine benutzerdefinierte Seite als Einstiegsseite definieren (Seite 913).

12.8.8 Einschränkungen bei benutzerdefinierten Webseiten

Die Einschränkungen bei Standard-Webseiten (Seite 937) gelten auch bei benutzerdefinierten Webseiten. Zudem gibt es bei benutzerdefinierten Webseiten einige spezifische Aspekte.

Kapazität des Ladespeichers

Ihre benutzerdefinierten Webseiten werden, wenn Sie auf "Bausteine generieren" klicken, zu Datenbausteinen, die Platz im Ladespeicher benötigen. Wenn Sie eine Memory Card gesteckt haben, steht die Kapazität Ihrer Memory Card als externer Ladespeicher für die benutzerdefinierten Webseiten zur Verfügung.

Wenn Sie keine Memory Card gesteckt haben, belegen diese Bausteine Platz im internen Ladespeicher, der je nach CPU-Modell begrenzt ist.

Mit den Online- und Diagnosefunktionen in STEP 7 können Sie den belegten und den freien Platz im Ladespeicher abfragen. Sie können zudem in den Eigenschaften der einzelnen Bausteine, die STEP 7 für Ihre benutzerdefinierten Webseiten generiert, den benötigten Platz im Ladespeicher prüfen.

Hinweis

Wenn Sie den Platz für Ihre benutzerdefinierten Webseiten verringern müssen, entfernen Sie ggf. einige der eingefügten Bilder.

Anführungszeichen in Textzeichenketten

In Datenbausteinvariablen, die in benutzerdefinierten Webseiten benutzt werden, sind keine Textzeichenketten mit eingebetteten einfachen oder doppelten Anführungszeichen zu verwenden. Da die HTML-Syntax oft einfache oder doppelte Anführungszeichen als Trennzeichen verwendet, können Anführungszeichen in Textzeichenfolgen die Anzeige benutzerdefinierter Webseiten stören.

Für Datenbausteinvariablen des Typs String, die in benutzerdefinierten Webseiten verwendet werden, sind die folgenden Regeln zu beachten:

- Im String-Wert der Datenbausteinvariablen in STEP 7 keine einfachen oder doppelten Anführungszeichen verwenden.
- Darauf achten, dass das Anwenderprogramm diesen Datenbausteinvariablen keine Zeichenketten mit Anführungszeichen zuweist.

12.8.9 Beispiel für eine benutzerdefinierte Webseite

12.8.9.1 Webseite zum Beobachten und Steuern einer Windturbine

Stellen Sie sich als Beispiel für eine benutzerdefinierte Webseite eine Webseite vor, die dazu dient, eine Windturbine entfernt zu beobachten und zu steuern:



Beschreibung

In dieser Anwendung ist jede Windturbine des Windparks mit einer S7-1200 zur Steuerung der Turbine ausgestattet. Im STEP 7-Programm hat jede Windturbine einen Datenbaustein mit Daten, die für die jeweilige Windturbine spezifisch sind.

Die benutzerdefinierte Webseite bietet dezentralen Zugriff auf die Turbine über einen PC. Ein Benutzer kann die Standard-Webseiten der CPU einer bestimmten Windturbine aufrufen und auf die benutzerdefinierte "Remote Wind Turbine Monitor"-Webseite (Webseite für die dezentrale Beobachtung der Windturbine) zugreifen, um die Daten der Turbine einzusehen. Ein Benutzer mit Rechten zum Ändern von Variablen kann zudem die Turbine in den manuellen Modus versetzen und die Variablen für Drehzahl, Ausrichtung und Anstellwinkel der Turbine über die Webseite steuern. Ein Benutzer mit Rechten zum Ändern von Variablen kann auch einen Bremswert festlegen, unabhängig davon, ob die Turbine manuell oder automatisch gesteuert wird.

Das STEP 7-Programm prüft die Booleschen Werte für Übersteuerung der automatischen Steuerung und verwendet, sofern diese eingestellt ist, die vom Anwender eingegebenen Werte für Drehzahl, Ausrichtung und Anstellwinkel der Turbine. Andernfalls ignoriert das Programm diese Werte.

Verwendete Dateien

Dieses Beispiel für eine benutzerdefinierte Webseite besteht aus drei Dateien:

- **Windturbine.html:** Dies ist die HTML-Seite mit der oben dargestellten Anzeige. Über AWP-Befehle wird auf Steuerungsdaten zugegriffen.
- **Windturbine.css:** Dies ist das Cascading Style Sheet, das die Formatierungsvorgaben für die HTML-Seite enthält. Die Verwendung eines Cascading Style Sheet ist optional, doch es kann die Entwicklung von HTML-Seiten vereinfachen.
- **Wind_turbine.jpg:** Dies ist das Hintergrundbild, das auf der HTML-Seite angezeigt wird. Die Verwendung von Bildern in benutzerdefinierten Webseiten ist natürlich freigestellt, und Bilder benötigen zusätzlichen Speicherplatz in der CPU.

Diese Dateien sind nicht in Ihrer Installation enthalten, werden jedoch als Beispiel beschrieben.

Implementierung

Die HTML-Seite verwendet AWP-Befehle zum Auslesen von Werten aus dem PLC-Gerät (Seite 898) für die Anzeigefelder und sie verwendet AWP-Befehle zum Schreiben von Werten in das PLC-Gerät (Seite 899) für die Daten der Benutzereingabe. Diese Seite nutzt zudem AWP-Befehle für die Definition von Enum-Typen (Seite 905) und für die Referenz (Seite 905) zur Handhabung von EIN/AUS-Einstellungen.

Der erste Teil der Seite zeigt eine Kopfzeile mit der Nummer der Windturbine an.

Fernüberwachung von Windturbinen: Turbine Nr. 5

Der nächste Teil der Seite zeigt die atmosphärischen Bedingungen an der Windturbine an. Die E/A am Turbinenstandort liefern die Windgeschwindigkeit, Windrichtung und die aktuelle Temperatur.

Windgeschwindigkeit:	7.5 km/h
Windrichtung:	23.5 Grad
Temperatur:	17.2 Grad C

Dann zeigt die Seite die aus der S7-1200 ausgelesene Leistungsabgabe der Turbine an.

Leistungsabgabe:	1000 kW
------------------	---------

Die folgenden Abschnitte ermöglichen die manuelle Steuerung der Turbine, eine Übersteuerung der normalen Automatiksteuerung der S7-1200. Die folgenden Übersteuerungen sind möglich:

- **Manuelle Übersteuerung:** Aktiviert die manuelle Übersteuerung der Turbine. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung wahr ist, damit die manuellen Einstellungen für Drehzahl, Ausrichtung oder Anstellwinkel der Turbine verwendet werden können.

Manuelle Übersteuerung: Ein	Einstellen: Ja
Turbinendrehzahl:	15 U/min

- **Übersteuerung der Ausrichtung:** Aktiviert die manuelle Übersteuerung der Turbinenausrichtung und die manuelle Einstellung der Ausrichtung. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung und für Übersteuerung der Ausrichtung wahr ist, damit die Ausrichtungseinstellung angewendet werden kann.
- **Übersteuerung des Anstellwinkels:** Aktiviert die manuelle Übersteuerung des Anstellwinkels der Rotorblätter. Das STEP 7-Anwenderprogramm verlangt, dass die Einstellung für manuelle Übersteuerung und für Übersteuerung des Anstellwinkels wahr ist, damit die Einstellung des Anstellwinkels der Rotorblätter angewendet werden kann.

Übersteuerung Ausrichtung: Ein Einstellen: Ja
Ausrichtung Turbine: 52 Grad

Übersteuerung Anstellwinkel: Ein Einstellen: Ja
Anstellwinkel Rotorblätter: 4.5 Grad

Die HTML-Seite enthält eine Schaltfläche "Senden", um die Übersteuerungseinstellungen an die Steuerung zu senden.

Einstellungen und Werte für Übersteuerung absenden

Das Eingabefeld für die Bremsung bietet eine manuelle Einstellung eines Prozentwerts für die Bremsung. Das STEP 7-Anwenderprogramm verlangt keine manuelle Übersteuerung, um einen Bremswert zu übernehmen.

Bremsen: 2.5 %

Außerdem nutzt die HTML-Seite einen AWP-Befehl, um die Sondervariable, die die Benutzer-ID des Benutzers enthält, der auf die Seite zugreift, in eine Variable in der PLC-Variablen-tabelle zu schreiben (Seite 903).

12.8.9.2 Steuerungsdaten lesen und anzeigen

Die HTML-Seite "Remote Wind Turbine Monitor" nutzt zahlreiche AWP-Befehle zum Lesen von Daten aus der Steuerung (Seite 898), um diese Daten auf der Seite anzuzeigen. Beachten Sie beispielsweise den HTML-Code zum Anzeigen der Leistungsausgabe in diesem Teil der Beispiel-Webseite:

Leistungsabgabe: 1000 kW

Beispiel für einen HTML-Code

Der folgende Auszug aus der HTML-Seite "Remote Wind Turbine Monitor" zeigt den Text "Leistungsabgabe:" in der linken Zelle einer Tabellenzeile an und liest die Variable für die Leistungsabgabe und zeigt sie in der rechten Zelle der Tabellenzeile zusammen mit dem Text für die Einheiten, kW, an.

Der AWP-Befehl := "Datenbaustein_1".Leistungsabgabe: führt die Leseoperation durch. Beachten Sie, dass Datenbausteine über den Namen und nicht über die Nummer des Datenbausteins referenziert werden (d.h. über "Datenbaustein_1" und nicht über "DB1").

```
<tr style="height:2%; ">
<td>
```

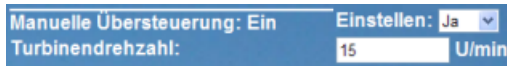
```

<p>Leistungsabgabe:</p>
</td>
<td>
<p style="margin-bottom:5px;"> :="Datenbaustein_1".Leistungsabgabe:
kW</p>
</td>
</tr>

```

12.8.9.3 Enum-Typ verwenden

Die HTML-Seite "Remote Wind Turbine Monitor" verwendet Enum-Typen für die drei Stellen, an denen die HTML-Seite "ON" oder "OFF" für einen Booleschen Wert anzeigt und an der der Benutzer einen Booleschen Wert eingibt. Der Enum-Typ für "ON" führt zu einem Wert von 1, und der Enum-Typ für "OFF" führt zu einem Wert von 0. Beispiel: Betrachten Sie den HTML-Code zum Lesen und Schreiben der Einstellung zum Aktivieren der manuellen Übersteuerung im Wert "Datenbaustein_1".FreigabeManuelleÜbersteuerung mittels eines Enum-Typs:



Beispiel für einen HTML-Code

Die folgenden Auszüge aus der HTML-Seite "Remote Wind Turbine Monitor" zeigen die Deklaration eines Enum-Typs mit dem Namen "Übersteuerungsstatus" mit den Werten für "Aus" und "Ein" von 0 und 1 und das nachfolgende Festlegen einer Enum-Typreferenz von "Übersteuerungsstatus" für die Boolesche Variable "FreigabeManuelleÜbersteuerung" im Datenbaustein "Datenbaustein_1".

```

<!-- AWP_In_Variable
Name=' "Datenbaustein_1".FreigabeManuelleÜbersteuerung'
Enum="Übersteuerungsstatus" -->

```

```

<!-- AWP_Enum_Def Name="Übersteuerungsstatus"
Values=' 0: "Aus" , 1: "Ein" ' -->

```

Wenn die HTML-Seite ein Anzeigefeld in einer Tabellenzelle für den aktuellen Zustand von "FreigabeManuelleÜbersteuerung" beinhaltet, wird lediglich ein normaler Lesebefehl für Variablen verwendet, doch dank des zuvor deklarierten und referenzierten Enum-Typs zeigt die Seite "Off" oder "On" und nicht 0 oder 1 an.

```

<td style="width:24%; border-top-style: Solid; border-top-width:
2px; border-top-color: #ffffff;">
<p>Manuelle
Übersteuerung: :="Datenbaustein_1".FreigabeManuelleÜbersteuerung:</p>
>
</td>

```

Die HTML-Seite enthält eine Klappliste, damit der Anwender den Wert von "FreigabeManuelleÜbersteuerung" ändern kann. Die Auswahlliste zeigt den Text "Ja" und "Nein" als Optionen an. Durch den Enum-Typ ist der Text "Ja" mit dem Wert "Ein" und der Text "Nein" mit dem Wert "Aus" verbunden. Wird keine Auswahl getroffen, bleibt der Wert von "FreigabeManuelleÜbersteuerung" unverändert.

```

<select name=' "Datenbaustein_1".FreigabeManuelleÜbersteuerung'>
<option value=' : "Datenbaustein_1".FreigabeManuelleÜbersteuerung: '>

```



```

</option>
<option value="Ein">Ja</option>
<option selected value="Aus">Nein</option>
</select>

```

Die Auswahlliste ist in einem Formular auf der HTML-Seite enthalten. Wenn der Anwender auf die Schaltfläche "Senden" klickt, lädt die Seite das Formular hoch. Dadurch wird der Wert "1" in die Boolesche Variable "FreigabeManuelleÜbersteuerung" in Datenbaustein_1 geschrieben, sofern der Anwender "Ja" ausgewählt hat, bzw. es wird "0" geschrieben, sofern der Anwender "Nein" ausgewählt hat.

12.8.9.4 Benutzereingaben in die Steuerung schreiben

Die HTML-Seite "Remote Wind Turbine Monitor" umfasst verschiedene AWP-Befehle zum Schreiben von Daten in die Steuerung (Seite 899). Die HTML-Seite deklariert AWP_In_Variables für Boolesche Variablen, so dass ein Benutzer mit Rechten zum Ändern von Variablen die Windturbine manuell steuern kann und die manuelle Übersteuerung für die Turbinendrehzahl, die Übersteuerung der Turbinenausrichtung und/oder des Anstellwinkels der Rotorblätter aktivieren kann. Die Seite umfasst zudem AWP_In_Variables, damit ein Benutzer mit Rechten zum Ändern von Variablen nachfolgend Gleitpunktwerte für Turbinendrehzahl, Ausrichtung, Anstellwinkel und Prozentwert für die Bremsung festlegen kann. Die Seite nutzt einen HTTP-Befehl zum Einstellen eines Formulars (Form POST), um die AWP_In_Variables in die Steuerung zu schreiben.

Beachten Sie z. B. den HTML-Code für die manuelle Einstellung des Bremswerts:



Beispiel für einen HTML-Code

Der folgende Ausschnitt aus der HTML-Seite "Remote Wind Turbine Monitor" deklariert zunächst eine AWP_In_Variable für "Datenbaustein_1", die es der HTML-Seite ermöglicht, in beliebige Variablen im Datenbaustein "Datenbaustein_1" zu schreiben. Die Seite zeigt den Text "Bremsen:" in der linken Zelle der Tabellenzeile an. In der rechten Zelle der Tabellenzeile befindet sich das Feld, in dem Benutzereingaben für die Variable "Bremsen" von "Datenbaustein_1" möglich sind. Dieser Benutzereingabewert befindet sich in einem HTML-Formular, das die eingegebenen Textdaten über die HTTP-Methode "POST" in die CPU schreibt. Die Seite liest dann den tatsächlichen Verzögerungswert aus der Steuerung aus und zeigt ihn im Dateneingabefeld an.

Ein Benutzer mit Rechten zum Ändern von Variablen kann anschließend über diese Seite einen Bremswert in den Datenbaustein der CPU schreiben, der die Bremsung steuert.

```

<!-- AWP_In_Variable Name="Datenbaustein_1" -->
...
<tr style="vertical-align: top; height: 2%;">
<td style="width: 22%;"><p>Bremsen:</p></td>
<td>
<form method="POST">
<p><input name=' "Datenbaustein_1".Bremsen' size="10" type="text"> %</
p>
</form>
</td>
</tr>

```

Hinweis

Beachten Sie Folgendes: Wenn eine benutzerdefinierte Seite ein Dateneingabefeld für eine schreibbare Datenbausteinvariable vom Datentyp "String" enthält, muss der Anwender die Zeichenkette bei der Eingabe des Textes in das Feld in einfache Anführungszeichen setzen.

Hinweis

Beachten Sie, dass, wenn Sie einen gesamten Datenbaustein in einer Deklaration AWP_In_Variable deklarieren, z. B. <!-- AWP_In_Variable Name="Datenbaustein_1" -->, dann kann jede Variable in dem Datenbaustein über die benutzerdefinierte Webseite geschrieben werden. Gehen Sie auf diese Weise vor, wenn Sie möchten, dass alle Variablen in einem Datenbaustein schreibbar sein sollen. Andernfalls, wenn Sie möchten, dass nur spezifische Datenbausteinvariablen über die benutzerdefinierte Webseite beschreibbar sein sollen, deklarieren Sie dies spezifisch anhand einer Deklaration wie <!-- AWP_In_Variable Name="Datenbaustein_1".Bremsung' -->

12.8.9.5 Sondervariablen schreiben

Die Webseite "Remote Wind Turbine Monitor" schreibt die Sondervariable SERVER:current_user_id in eine PLC-Variable in der CPU, wenn der Benutzer über Änderungsberechtigungen verfügt. In diesem Fall enthält der PLC-Variablenwert die Benutzer-ID des Benutzers, der auf die Webseite "Remote Wind Turbine Monitor" zugreift.

Die Sondervariable wird von der Webseite in das PLC-Gerät geschrieben und benötigt keine Benutzeroberfläche.

Beispiel für einen HTML-Code

```
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="Benutzer-ID"-->
```

12.8.9.6 Referenz: HTML-Code der Webseite "Remote Wind Turbine Monitor"

Windturbine.html

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<!--
Dieses Testprogramm simuliert eine Webseite zur Überwachung und
Bedienung einer Windturbine.
Erforderliche PLC-Variablen und Datenbausteinvariablen in STEP 7:

PLC-Variable:
Benutzer-ID: Int
Datenbausteine:
Datenbaustein_1
Variablen in Datenbaustein_1:
```

```

Turbinennummer: Int
Windgeschwindigkeit: Real
Windrichtung: Real
Temperatur: Real
Leistungsabgabe: Real
FreigabeManuelleÜbersteuerung: Bool
Turbinendrehzahl: Real
ÜbersteuerungAusrichtung: Bool
Ausrichtung: Real
ÜbersteuerungAnstellwinkel: Bool
Anstellwinkel: Real
Bremsen: Real

```

Die benutzerdefinierte Webseite zeigt aktuelle Werte für die PLC-Daten und bietet eine Auswahlliste, um die drei Booleschen Werte mit zugewiesenem Aufzählungstyp festzulegen. Über die Schaltfläche "Senden" werden die ausgewählten Booleschen Werte ebenso wie die Dateneingabefelder für Drehzahl, Ausrichtung und Anstellwinkel der Turbine hochgeladen. Der Bremswert kann ohne die Schaltfläche "Senden" festgelegt werden.

Für die Verwendung dieser Seite ist kein tatsächliches STEP 7-Programm erforderlich. Theoretisch würde das STEP 7-Programm nur auf die Werte für Drehzahl, Ausrichtung und Anstellwinkel der Turbine reagieren, wenn die zugewiesenen Booleschen Werte festgelegt wären. Die einzige Anforderung an STEP 7 ist, die WWW-Anweisung mit der DB-Nummer der generierten Datenbausteine für diese Seite aufzurufen.

```

-->
<!-- AWP_In_Variable Name="Datenbaustein_1" -->
<!-- AWP_In_Variable
Name="Datenbaustein_1".FreigabeManuelleÜbersteuerung'
Enum="Übersteuerungsstatus" -->
<!-- AWP_In_Variable
Name="Datenbaustein_1".ÜbersteuerungAnstellwinkel'
Enum="Übersteuerungsstatus" -->
<!-- AWP_In_Variable
Name="Datenbaustein_1".ÜbersteuerungAusrichtung'
Enum="Übersteuerungsstatus" -->
<!-- AWP_In_Variable Name="SERVER:current_user_id" Use="Benutzer-
ID"-->
<!-- AWP_Enum_Def Name="Übersteuerungsstatus"
Values='0:"Aus",1:"Ein"' -->

```

```

<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=utf-8"><link rel="stylesheet" href="Windturbine.css">
<title>Fernüberwachung von Windturbinen</title>
</head>
<body>
<table cellpadding="0" cellspacing="2">
<tr style="height: 2%;">
<td colspan="2">

```

```

<h2>Fernüberwachung von Windturbinen: Turbine
Nr. := "Datenbaustein_1".Turbinennummer:</h2>
</td>

<tr style="height: 2%;"><td style="width:
25%;"><p>Windgeschwindigkeit:</p></td>
<td><p> := "Datenbaustein_1".Windgeschwindigkeit: km/h</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Windrichtung:</p></td>
<td><p> := "Datenbaustein_1".Windrichtung: Grad</p></td>
</tr>

<tr style="height: 2%;"><td style="width:
25%;"><p>Temperatur:</p></td>
<td><p> := "Datenbaustein_1".Temperatur: Grad C</p></td>
</tr>

<tr style="height: 2%;">
<td style="width: 25%;"><p>Leistungsabgabe:</p></td>
<td><p style="margin-
bottom: 5px;"> := "Datenbaustein_1".Leistungsabgabe: 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>Manuelle
Übersteuerung: := "Datenbaustein_1".FreigabeManuelleÜbersteuerung:</p>
>
</td>
<td class="Text">Einstellen:

<select name=' "Datenbaustein_1".FreigabeManuelleÜbersteuerung'>
<option value=' := "Datenbaustein_1".FreigabeManuelleÜbersteuerung:'>
</option>
<option value="Ein">Ja</option>
<option value="Aus">Nein</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;"><td style="width:
25%;"><p>Turbinendrehzahl:</p></td>
<td>
<p style="margin-bottom: 5px;"><input
name= "Datenbaustein_1".Turbinendrehzahl' size="10"
value=' := "Datenbaustein_1".Turbinendrehzahl:' type="text"> U/min</p>

```

```

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Übersteuerung
Ausrichtung: := "Datenbaustein_1".ÜbersteuerungAusrichtung: </p>
</td>
<td class="Text">Einstellen:

<select name=' "Datenbaustein_1".ÜbersteuerungAusrichtung'>
<option value=' := "Datenbaustein_1".ÜbersteuerungAusrichtung:' </
option>
<option value="Ein">Ja</option>
<option value="Aus">Nein</option>
</select>

</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Ausrichtung Turbine:</p>
</td>
<td>
<p style="margin-bottom:5px;"><input
name=' "Datenbaustein_1".Ausrichtung' size="10"
value=' := "Datenbaustein_1".Ausrichtung:' type="text"> Grad</p>
</td>
</tr>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;">
<p>Übersteuerung
Anstellwinkel: := "Datenbaustein_1".ÜbersteuerungAnstellwinkel: </p>
</td>
<td class="Text">Einstellen:

<select name=' "Datenbaustein_1".ÜbersteuerungAnstellwinkel'>
<option value=' := "Datenbaustein_1".ÜbersteuerungAnstellwinkel:' </
option>
<option value="Ein">Ja</option>
<option value="Aus">Nein</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>Anstellwinkel Rotorblätter:</p>

```

```

</td>
<td>
<p style="margin-bottom:5px;"><input
name='"Datenbaustein_1".Anstellwinkel' size="10"
value=':="Datenbaustein_1".Anstellwinkel:' type="text"> Grad</p>
</td>

</tr>
<tr style="height: 2%;">
<td colspan="2">
<input type="submit" value="Einstellungen und Werte für
Übersteuerung absenden">
</td>
</tr>
</form>

<tr style="vertical-align: top; height: 2%;">
<td style="width: 25%;"><p>Bremsen:</p></td>
<td>
<form method="POST" action="">
<p> <input name='"Datenbaustein_1".Bremsen' size="10"
value=':="Datenbaustein_1".Bremsen:' type="text"> %</p>
</form>
</td>
</tr>
<tr><td></td></tr>

</table>
</body>
</html>

```

Windturbine.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;
}

```

```

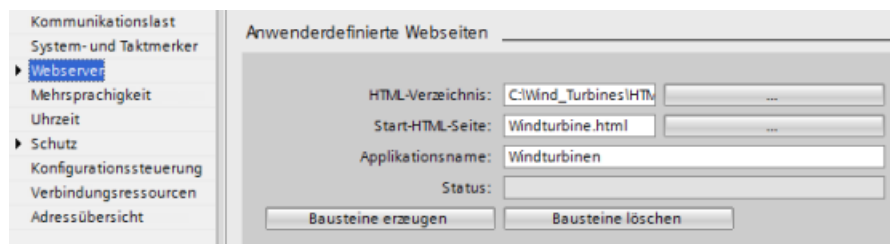
margin-top: 0px;
margin-bottom: 0px;
}
TD.Text {
font-family: Arial;
font-weight: bold;
color: #FFFFFF;
font-size: 12.0pt;
margin-top: 0px;
margin-bottom: 0px;
}

```

12.8.9.7 Konfiguration der Beispiel-Webseite in STEP 7

Um die HTML-Seite "Remote Wind Turbine Monitor" als benutzerdefinierte Webseite für die S7-1200 einzufügen, konfigurieren Sie die Daten für die HTML-Seite in STEP 7 und erstellen aus der HTML-Seite Datenbausteine.

Sie greifen auf die CPU-Eigenschaften der S7-1200, die die Windturbine steuert, zu und geben die Konfigurationsinformationen in die Eigenschaften der benutzerdefinierten Seiten des Webserver ein:



Konfigurationsfelder

- **HTML-Verzeichnis:** Dieses Feld gibt den vollständig qualifizierten Pfadnamen auf den Ordner an, in dem sich die Standardseite (Startseite) auf dem Computer befindet. Über die Schaltfläche "..." können Sie zu dem gewünschten Ordner blättern.
- **Standard-HTML-Seite:** Dieses Feld gibt den Dateinamen der Standardseite bzw. der Startseite der HTML-Anwendung an. Über die Schaltfläche "..." können Sie die gewünschte Datei auswählen. In diesem Beispiel ist Windturbine.html die Standard-HTML-Seite. Das Beispiel "Remote Wind Turbine Monitor" besteht lediglich aus einer einzigen Seite, doch in anderen benutzerdefinierten Anwendungen können von der Standardseite weitere Seiten über Links aufgerufen werden. Im HTML-Code muss die Standardseite andere Seiten relativ zum HTML-Quellordner referenzieren.
- **Anwendungsname:** Dieses Feld ist optional und enthält den Namen, den der Webbrowser im Adressfeld anzeigt, wenn die Seite aufgerufen wird. In diesem Beispiel lautet der Name "Remote Wind Turbine Monitor", Sie können jedoch einen beliebigen Namen eingeben.

Es müssen keine weiteren Felder konfiguriert werden.

Abschließende Schritte

Um die HTML-Seite "Remote Wind Turbine Monitor" wie konfiguriert nutzen zu können, erstellen Sie die Bausteine, programmieren die WWW-Anweisung (Seite 913) mit der Nummer des generierten Steuer-DBs als Eingangsparameter, laden die Programmbausteine in die CPU und versetzen die CPU in den Betriebszustand RUN.

Wenn ein Bediener danach die Standard-Webseiten für die S7-1200 aufruft, von der die Windturbine gesteuert wird, kann er über den Link "Anwenderdefinierte Seiten" in der Navigationsleiste die Webseite "Remote Wind Turbine Monitor" anzeigen. Diese Seite bietet nun die Mittel, um die Windturbine zu überwachen und zu steuern.

12.8.10 Benutzerdefinierte Webseiten in mehreren Sprachen einrichten

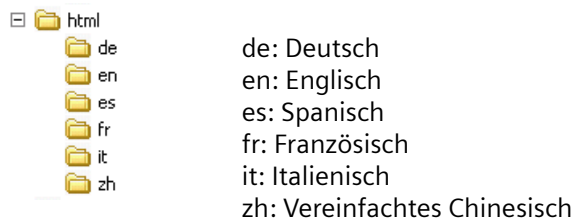
Der Webserver bietet Ihnen die Möglichkeit, benutzerdefinierte Webseiten in den folgenden Sprachen anzulegen:

- Deutsch (de)
- Englisch (en)
- Spanisch (es)
- Französisch (fr)
- Italienisch (it)
- Vereinfachtes Chinesisch (zh)

Hierfür legen Sie Ihre HTML-Seiten in einer Ordnerstruktur (Seite 928) entsprechend den Sprachen an und richten das spezifische Cookie "siemens_automation_language" für Ihre Seiten ein (Seite 929). Der Webserver reagiert auf dieses Cookie und schaltet zu der Standardseite in dem jeweiligen Sprachordner um.

12.8.10.1 Ordnerstruktur anlegen

Um benutzerdefinierte Webseiten in mehreren Sprachen bereitzustellen, richten Sie in Ihrem HTML-Verzeichnis eine Ordnerstruktur ein. Die aus zwei Buchstaben bestehenden Ordnernamen sind spezifisch und müssen wie im Folgenden gezeigt vergeben werden:



Auf der gleichen Ebene können Sie auch andere Ordner für Ihre Seiten anordnen, z. B. Ordner für Bilder oder Skripte.

Sie können beliebige Sprachordner einfügen. Sie brauchen nicht alle sechs Sprachen aufzunehmen. In den Sprachordnern erstellen und programmieren Sie Ihre HTML-Seiten in der jeweiligen Sprache.

12.8.10.2 Sprachumschaltung programmieren

Der Webserver führt die Sprachumschaltung anhand des Cookies "siemens_automation_language" durch. Dieses Cookie wird in den HTML-Seiten definiert und eingerichtet und vom Webserver ausgewertet, um eine Seite in der jeweiligen Sprache aus dem Sprachordner mit dem Namen der entsprechenden Sprache anzuzeigen. Die HTML-Seite muss JavaScript enthalten, um für dieses Cookie eine der vordefinierten Sprachkennungen einzurichten: "de", "en", "es", "fr", "it" oder "zh".

Setzt die HTML-Seite z. B. das Cookie auf "de", schaltet der Webserver zum Ordner "de" um und zeigt die Seite mit dem HTML-Standardseitennamen gemäß der STEP 7-Konfiguration (Seite 931) an.

Beispiel

Im folgenden Beispiel gibt es eine HTML-Standardseite mit dem Namen "langswitch.html" in jedem der Sprachordner. Im HTML-Verzeichnis befindet sich außerdem der Ordner "script". Der Ordner "script" enthält eine JavaScript-Datei mit dem Namen "lang.js". Jede Seite "langswitch.html" nutzt dieses JavaScript, um das Sprachcookie "siemens_automation_language" festzulegen.

HTML für "langswitch.html" im Ordner "en"

Die Kopfzeile der HTML-Seite richtet die Sprache "Englisch" ein, legt den Zeichensatz "UTF-8" fest und gibt den Pfad der JavaScript-Datei "lang.js" an.

```
<!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>
```

Der Inhalt der Datei enthält eine Auswahlliste, über die der Anwender zwischen Deutsch und Englisch wählen kann. Englisch ("en") ist die vorausgewählte Sprache. Wenn der Anwender die Sprache wechselt, ruft die Seite die JavaScript-Funktion DoLocalLanguageChange() mit dem Wert der ausgewählten Option auf.

```
<!-- 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 für "langswitch.html" im Ordner "de"

Die Kopfzeile für die deutsche Seite "langswitch.html" ist mit der Kopfzeile der englischen Seite identisch, außer dass die deutsche Sprache eingestellt ist.

```
<!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>
```

Der HTML-Code auf der deutschen Seite ist mit dem der englischen Seite identisch, außer dass der Standardwert der ausgewählten Sprache Deutsch ("de") ist.

```
<!-- 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" im Ordner "script"

Die Funktion "DoLocalLanguageChange()" befindet sich in der Datei "lang.js". Diese Funktion ruft die Funktion "SetLangCookie()" auf und aktualisiert dann das Fenster, in dem die HTML-Seite angezeigt wird.

Die Funktion "SetLangCookie()" weist dem Cookie "siemens_automation_language" des Dokuments den Wert aus der Auswahlliste zu. Die Funktion legt auch den Pfad der Anwendung fest, so dass die umgeschaltete Seite und nicht die anfordernde Seite den Wert des Cookies empfängt.

Optional kann die Seite im kommentierten Abschnitt einen Wert für die Gültigkeit des Cookies angeben.

```
function DoLocalLanguageChange(oSelect) {
  SetLangCookie(oSelect.value);
  top.window.location.reload();
}
function SetLangCookie(value) {
  var strval = "siemens_automation_language=";
  // Dies ist das Cookie, über das der Webserver
  // die gewünschte Sprache erkennt.
  // Dieser Name ist für den Webserver erforderlich.
  strval = strval + value;
  strval = strval + "; path=/ ";
```

```

// Geben Sie den Pfad für die Anwendung an, weil ansonsten
// der Pfad auf die anfordernde Seite gelegt werden würde
// und diese Seite das Cookie nicht erhalten würde.
/* OPTIONAL
   Geben Sie eine Ablaufzeit an, wenn dieses Cookie länger
existieren soll
   als die aktuelle Browser-Sitzung:
   var now      = new Date();
   var endtime = new Date(now.getTime() + expiration);
   strval = strval + "; expires=" +
           endtime.toGMTString() + ";";
*/
document.cookie = strval;
}

```

Hinweis

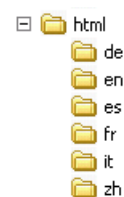
Wenn Ihre benutzerdefinierte Webseitenimplementierung sowohl HTML-Dateien in sprachspezifischen Ordnern (z. B. en, de) als auch HTML-Dateien enthält, die sich nicht in sprachspezifischen Ordnern befinden, können Sie mit dem Befehl `AWP_Enum_Def` Enum-Typen nicht in Dateien an beiden Speicherorten definieren. Stattdessen müssen Sie bei der Verwendung von Enum-Typen diese entweder in Dateien in den sprachspezifischen Ordnern oder in Dateien außerhalb der sprachspezifischen Ordner definieren. Sie können Enum-Deklarationen nicht in Dateien in beiden Speicherorten vornehmen.

12.8.10.3 STEP 7 für die Verwendung einer mehrsprachigen Seitenstruktur konfigurieren

Um mehrsprachige benutzerdefinierte Webseiten zu konfigurieren, gehen Sie ähnlich vor wie beim Konfigurieren von benutzerdefinierten Webseiten (Seite 911). Wenn Sie jedoch Ordner für Sprachen eingerichtet haben, geben Sie in der Einstellung des HTML-Verzeichnisses den Ordner an, der die einzelnen Sprachordner enthält. Sie richten das HTML-Verzeichnis nicht als einen der Sprachordner ein.

Wenn Sie die HTML-Standardseite auswählen, navigieren Sie in den Sprachordner und wählen die HTML-Seite aus, die die Startseite sein soll. Wenn Sie anschließend Bausteine generieren und die Bausteine in die CPU laden, zeigt der Webserver die Startseite aus dem konfigurierten Sprachordner an.

Beispiel: Befindet sich die hier gezeigte Ordnerstruktur auf C:\, ist die Einstellung für das HTML-Verzeichnis C:\html. Und wenn zunächst die englische Seite angezeigt werden soll, navigieren Sie für die Einstellung der HTML-Standardseite zum Pfad `en\langswitch.html`.



12.8.11 Erweiterte Steuerung von benutzerdefinierten Webseiten

Wenn Sie für Ihre benutzerdefinierten Webseiten Datenbausteine generieren, erstellt STEP 7 einen Steuer-DB, der die Anzeige von und die Interaktion mit den benutzerdefinierten Webseiten steuert. STEP 7 erstellt zudem einen Satz DB-Fragmente, die die einzelnen Seiten darstellen. Unter normalen Umständen müssen Sie die Struktur des Steuer-DBs nicht kennen und brauchen auch nicht zu wissen, wie Sie ihn ändern.

Wenn Sie beispielsweise eine Webanwendung ein- und ausschalten möchten oder einzelne Fragmente manuell ändern möchten, können Sie dies über die Variablen des Steuer-DBs und die WWW-Anweisung erreichen.

Struktur des Steuer-DBs

Der Steuer-DB besitzt eine umfangreiche Datenstruktur und ist zugänglich, wenn Sie Ihr STEP 7-Anwenderprogramm programmieren. Hier werden nur einige der Variablen des Steuerdatenbausteins beschrieben.

Commandstate-Struktur

"Commandstate" ist eine Struktur, die globale Befehle und globale Zustände für den Webserver enthält.

Globale Befehle in der "Commandstate"-Struktur

Die globalen Befehle gelten für den Webserver im Allgemeinen. Sie können den Webserver über die Parameter des Steuer-DBs deaktivieren oder neu starten.

Bausteinvariable	Datentyp	Beschreibung
init	BOOL	Auswertung des Steuer-DB und Initialisierung der Webanwendung
deactivate	BOOL	Deaktivierung der Webanwendung

Globale Zustände in der Commandstate-Struktur

Die globalen Zustände gelten für den Webserver im Allgemeinen und enthalten Statusinformationen über die Webanwendung.

Bausteinvariable	Datentyp	Beschreibung
initializing	BOOL	Webanwendung liest den Steuer-DB
error	BOOL	Webanwendung konnte nicht initialisiert werden
deactivating	BOOL	Webanwendung wird beendet
deactivated	BOOL	Webanwendung ist beendet

Bausteinvariable	Datentyp	Beschreibung
initialized	BOOL	Webanwendung ist initialisiert
last_error	INT	Letzter Fehler, der von einem WWW-Anweisungsaufwurf (Seite 913) zurückgegeben wurde, wenn der Rückgabecode für WWW 16#0010 lautet: 16#0001: Struktur DB-Fragment inkonsistent 16#0002: Name der Anwendung existiert bereits 16#0003: keine Ressourcen (Speicher) 16#0004: Struktur Steuer-DB inkonsistent 16#0005: DB-Fragment nicht verfügbar 16#0006: DB-Fragment nicht für AWP 16#0007: Aufzählungsdaten inkonsistent 16#000D: Konflikt durch Größe des Steuer-DBs

Anforderungstabelle

Die Anforderungstabelle ist ein Array aus Strukturen mit Befehlen und Zuständen, die für einzelne DB-Fragmente gelten. Wenn Sie mit dem Befehl AWP_Start_Fragment (Seite 907) Fragmente des "manuellen" Typs angelegt haben, muss das STEP 7-Anwenderprogramm diese Seiten über den Steuer-DB steuern. Die Anforderungszustände sind schreibgeschützt und liefern Informationen zum aktuellen Fragment. Die Anforderungsbefehle nutzen Sie, um das aktuelle Fragment zu steuern.

Bausteinvariable	Datentyp	Beschreibung
requesttab	ARRAY [1 .. 4] OF STRUCT	Array aus Strukturen für die Steuerung einzelner DB-Fragmente Der Webserver kann bis zu vier Fragmente gleichzeitig bearbeiten. Der Array-Index für ein bestimmtes Fragment ist beliebig, wenn der Webserver mehrere Fragmente oder Fragmente aus mehreren Browser-Sessions verarbeitet.

Struct-Elemente der Requesttab-Struktur

Bausteinvariable	Datentyp	Beschreibung
page_index	UINT	Nummer der aktuellen Webseite
fragment_index	UINT	Nummer des aktuellen Fragments - Festlegung eines anderen Fragments möglich
// Anforderungsbefehle		
continue	BOOL	Gibt die aktuelle Seite bzw. das aktuelle Fragment zum Senden frei und fährt mit dem nächsten Fragment fort
repeat	BOOL	Gibt die aktuelle Seite bzw. das aktuelle Fragment zum erneuten Senden frei und fährt mit demselben Fragment fort
abort	BOOL	HTTP-Verbindung ohne Senden schließen
finish	BOOL	Dieses Fragment senden; Seite ist vollständig - keine weiteren Fragmente bearbeiten
// Anforderungszustände		
idle	BOOL	Nichts durchzuführen, jedoch aktiv

Bausteinvariable	Datentyp	Beschreibung
waiting	BOOL	Fragment wartet auf Freigabe
sending	BOOL	Fragment sendet
aborting	BOOL	Benutzer hat die aktuelle Anforderung abgebrochen

Funktionsweise

Immer wenn Ihr Programm den Steuer-DB ändert, muss es die WWW-Anweisung mit der Nummer des geänderten Steuer-DBs als Parameter aufrufen. Die globalen Befehle und Anforderungsbefehle werden wirksam, wenn das STEP 7-Anwenderprogramm die WWW-Anweisung (Seite 913) ausführt.

Das STEP 7-Anwenderprogramm kann fragment_index explizit festlegen und bewirkt dadurch, dass der Webserver das angegebene Fragment mit einem Anforderungsbefehl verarbeitet. Andernfalls verarbeitet der Webserver das aktuelle Fragment für die aktuelle Seite, wenn die WWW-Anweisung ausgeführt wird.

Mögliche Techniken zur Verwendung von fragment_index sind u.a.:

- Aktuelles Fragment verarbeiten: fragment_index unverändert lassen und den Fortsetzungsbefehl setzen.
- Aktuelles Fragment überspringen: fragment_index auf 0 setzen und den Fortsetzungsbefehl setzen
- Aktuelles Fragment durch ein anderes Fragment ersetzen: Für fragment_index die neue Fragment-ID angeben und den Fortsetzungsbefehl setzen

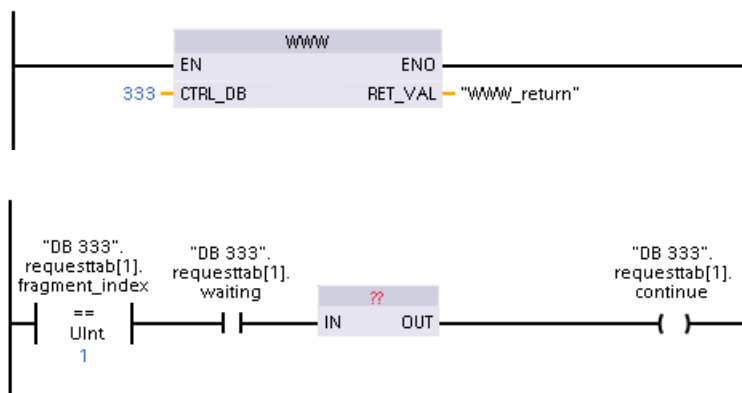
Um sich möglicherweise verändernde globale Zustände oder Anforderungszustände abzufragen, muss das STEP 7-Anwenderprogramm die WWW-Anweisung aufrufen, um die aktuellen Werte dieser Zustände auszuwerten. Eine typische Verwendung könnte es sein, die WWW-Anweisung regelmäßig aufzurufen, bis ein bestimmter Zustand eintritt.

Hinweis

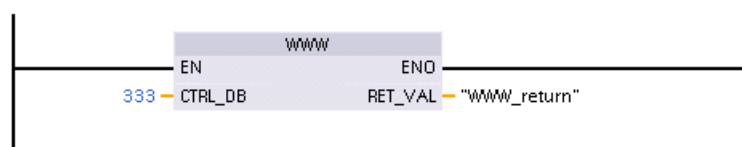
Wenn das STEP 7-Anwenderprogramm mehrere Anforderungsbefehle setzt, verarbeitet die WWW-Anweisung nur einen davon und hält dabei diese Reihenfolge ein: abbrechen, beenden, wiederholen, fortfahren. Die WWW-Anweisung löscht alle Anforderungsbefehle nach der Verarbeitung.

Beispiele

Das folgende Beispiel zeigt ein STEP 7-Anwenderprogramm, das auf ein Fragment mit der ID 1 im Wartezustand prüft, nachdem zuvor die WWW-Anweisung aufgerufen wurde. Es wartet möglicherweise auch, dass andere anwendungsspezifische Bedingungen auftreten. Dann führt es die für das Fragment erforderliche Verarbeitung durch, wobei es z. B. Datenbausteinvariablen setzt, Berechnungen durchführt oder andere anwendungsspezifische Aufgaben verarbeitet. Danach setzt es den Fortsetzungsmerker, damit der Webserver dieses Fragment ausführt.



Wenn das Programm die WWW-Anweisung mit diesem geänderten Steuer-DB aufruft, kann die benutzerdefinierte Webseite mit diesem Fragment vom Webbrowser angezeigt werden.



Beachten Sie, dass es sich hier um ein vereinfachtes Beispiel handelt. Das zu prüfende Fragment kann sich in jeder der vier requeststab-Strukturen des Arrays befinden. Ihr Programm muss alle vier requeststab-Strukturen bearbeiten.

12.8.12 Web-API

12.8.12.1 Web-API

Die S7-1200 CPU bietet eine Web-API, die Ihnen als Schnittstelle zum Lesen und Schreiben von Prozessdaten dient. Die S7-1200 Web-API implementiert die Funktionalität der S7-1500 Web-API, die im Kapitel **Webseiten > Anwendungsprogrammierschnittstelle (API)** in diesem Dokument (<https://support.industry.siemens.com/cs/us/en/view/59193560/de>) dokumentiert wird.

Das S7-1500 Dokument beschreibt die Web-API-Funktionalität für einen bestimmten Satz S7-1500 CPUs. Diese Funktionalität ist für alle S7-1200 CPUs ab Firmware V4.5 verfügbar. In STEP 7 müssen Sie die Gerätekonfiguration ab V4.5 konfigurieren.

Beachten Sie, dass die S7-1200 CPU die Anzahl gleichzeitiger API-Sitzungen auf 50 begrenzt.

Die S7-1200 Web-API unterstützt von der S7-1500 Web-API unterstützten Datentypen mit Ausnahme der folgenden Datentypen:

Datentyp

- 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

Die von der S7-1500 Web-API unterstützten Datentypen finden Sie hier: **Webseiten > Anwendungsprogrammierschnittstelle (API) > Prozessdaten lesen und schreiben > Unterstützte Datentypen** in diesem Dokument (<https://support.industry.siemens.com/cs/us/en/view/59193560/de>).

12.8.12.2 Unterstützte Web-API-Methoden

Ab V4.5 unterstützt die S7-1200 die folgenden Web-API-Methoden:

- Unterstützte Web-API-Methoden
- Api.Ping
- Api.Version
- Api.GetCertificateUrl
- Api.Browse

Api.Login
Api.Logout
Api.GetPermissions
PicProgram.Browse
PicProgram.Write
PicProgram.Read

12.9 Einschränkungen

Die folgenden IT-Aspekte können sich auf Ihre Nutzung des Webserver auswirken:

- Für den Zugriff auf die Standard-Webseiten oder auf benutzerdefinierte Webseiten muss im Allgemeinen die IP-Adresse der CPU oder die IP-Adresse eines Wireless-Routers mit einer Portnummer verwendet werden. Wenn Ihr Webbrowser keine direkte Verbindung mit einer IP-Adresse gestattet, wenden Sie sich an Ihren IT-Administrator. Wenn Ihre lokalen Richtlinien DNS unterstützen, können Sie über einen DNS-Eintrag eine Verbindung zu der IP-Adresse herstellen.
- Firewalls, Proxy-Einstellungen und andere standortspezifische Einschränkungen können ebenfalls den Zugriff auf die CPU begrenzen. Um solche Probleme zu beheben, wenden Sie sich an Ihren IT-Administrator.
- Die Standard-Webseiten verwenden JavaScript und Cookies. Sind JavaScript oder Cookies in Ihrem Webbrowser deaktiviert, aktivieren Sie sie. Wenn Sie sie nicht aktivieren können, sind einige Funktionen eingeschränkt (Seite 938). Die Verwendung von JavaScript und Cookies in benutzerdefinierten Webseiten ist optional. Wenn Sie sie verwenden, müssen sie in Ihrem Browser aktiviert sein.
- Der Webserver unterstützt Secure Sockets Layer (SSL). Sie können die Standard-Webseiten und benutzerdefinierten Webseiten über eine der beiden URLs `http://ww.xx.yy.zz` oder `https://ww.xx.yy.zz` aufrufen. Dabei steht "ww.xx.yy.zz" für die IP-Adresse der CPU.
- Siemens bietet für den sicheren Zugriff auf den Webserver ein Sicherheitszertifikat. Auf der Standard-Webseite "Einführung" (Seite 866) können Sie das Zertifikat herunterladen und in die Internetoptionen Ihres Webbrowsers importieren (Seite 858). Wenn Sie das Zertifikat nicht importieren, wird Ihnen bei jedem Aufruf des Webserver über die `https://`-Adresse eine Sicherheitsabfrage angezeigt.

Anzahl der Verbindungen

Der Webserver unterstützt maximal 30 aktive Verbindungen. Diese 30 Verbindungen können auf verschiedene Weise genutzt werden, je nach dem von Ihnen verwendeten Webbrowser und der Anzahl der unterschiedlichen Objekte pro Seite (CSS-Dateien, Bilder, JavaScript-Dateien, weitere HTML-Dateien). Einige Verbindungen bleiben bestehen, während der Webserver eine Seite anzeigt, andere werden nach der Erstverbindung freigegeben.

Wenn Sie beispielsweise bestimmte Versionen von Mozilla Firefox nutzen, die maximal sechs beständige Verbindungen unterstützen, könnten Sie fünf Browser oder Browserregister verwenden, bevor der Webserver beginnt, Verbindungen abubrechen. Sollte eine Seite nicht alle sechs Verbindungen nutzen, können Sie weitere Browser oder Browserregister verwenden.

Beachten Sie auch, dass die Anzahl aktiver Verbindungen die Leistung der Seite beeinflussen kann. Deshalb werden die Webseiten möglicherweise nicht vollständig geladen.

Hinweis

Vor dem Schließen des Webserver abmelden

Wenn Sie sich am Webserver angemeldet haben, sollten Sie sich vor dem Schließen des Webbrowsers unbedingt abmelden. Der Webserver unterstützt maximal sieben gleichzeitig angemeldete Benutzer.

Durch Nicht-Abmelden kann es, abhängig von Ihrem Browser, zu mehreren gleichzeitig geöffneten Verbindungen kommen. Wenn Sie Webserver-Browserfenster mehrmals öffnen und schließen, ohne sich abzumelden, können Sie auf diese Weise alle 30 Verbindungen belegen. Wenn Sie alle Verbindungen belegen, wird Ihnen bei nächstem Anmeldeversuch die Meldung "Ungültiges Login" angezeigt. Sie müssen dann bis zu 30 Minuten warten, bis der Webserver genügend Verbindungen freisetzt und Sie sich erneut anmelden können. Um dieses Problem zu vermeiden, melden Sie sich immer ab, wenn Sie den Webserver schließen (sofern Sie sich angemeldet hatten).

12.9.1 Verwendung von JavaScript

Die Standard-Webseiten verwenden HTML, JavaScript und Cookies. Wenn an Ihrem Standort die Verwendung von JavaScript und Cookies eingeschränkt ist, aktivieren Sie diese, damit die Webseiten einwandfrei funktionieren. Wenn Sie für Ihren Webbrowser JavaScript nicht aktivieren können, können die Standard-Webseiten nicht ausgeführt werden. Sie können die Basisseiten verwenden, die kein JavaScript benötigen.

Siehe auch

Aufbau der Standard-Webseiten (Seite 861)

12.9.2 Eingeschränkte Funktionen, wenn Cookies in den Internetoptionen nicht erlaubt sind

Wenn Cookies in Ihrem Webbrowser deaktiviert sind, gelten die folgenden Einschränkungen:

- Sie können sich nicht anmelden.
- Sie können die Sprache nicht wechseln.
- Sie können nicht von UTC- auf PLC-Zeit umschalten. Ohne Cookies sind alle Zeiten UTC-Zeiten.

12.9.3 Regeln für die Eingabe von Variablennamen und Werten

Beachten Sie bei der Nutzung der Standardseiten Variablenstatus (Seite 879) und Beobachtungstabellen (Seite 881) Folgendes:

- Wenn Sie den gesamten Wert einer DTL-Variablen ändern, z. B. "Datenbaustein_1_DTL_Variable, verwenden Sie die folgende DTL-Syntax für den Änderungswert: DTL#JJJ-MM-TT-HH-MM-SS[.ssssssss]
- Gehen Sie bei der Eingabe eines Werts vom Datentyp Real oder LReal in Exponentialdarstellung wie folgt vor:
 - Um einen Realzahlenwert (Real oder LReal) mit positivem Exponenten (wie +3.402823e+25) einzugeben, geben Sie den Wert in einem der folgenden Formate ein:
+3.402823e25
+3.402823e+25
 - Um einen Realzahlenwert (Real oder LReal) mit negativem Exponenten (wie +3.402823e-25) einzugeben, geben Sie den Wert wie folgt ein:
+3.402823e-25
 - Achten Sie darauf, dass der Anteil der Mantisse des Realzahlenwerts in der Exponentialdarstellung einen Dezimalpunkt enthält. Wird der Dezimalpunkt weggelassen, führt dies dazu, dass der Wert zu einem unerwarteten ganzzahligen Wert verändert wird. Beispiel: Geben Sie -1.0e8 und nicht -1e8 ein.
- Werte für LReal dürfen nur 15 Ziffern haben (unabhängig von der Position des Dezimalpunkts). Wenn Sie mehr als 15 Ziffern eingeben, entsteht ein Rundungsfehler.

Einschränkungen auf der Seite Variablenstatus und Beobachtungstabelle:

- Die maximale Anzahl von Zeichen für die URL beträgt 2083. Sie sehen die URL, die Ihrer aktuellen Variablenseite entspricht, in der Adressleiste Ihres Browsers.
- Hinsichtlich des Zeichenanzeigeformats zeigt die Seite, wenn es sich nach Auswertung des Browsers bei den tatsächlichen CPU-Werten nicht um gültige ASCII-Zeichen handelt, das Zeichen mit vorangestelltem Dollarzeichen an: \$.

12.9.4 Datenprotokolle im CSV-Format in nicht amerikanische/englische Versionen von Microsoft Excel importieren

Die Protokolldateien sind im CSV-Dateiformat (durch Komma getrennte Werte) gespeichert. Sie können diese Dateien von der Datenprotokollseite direkt in Excel öffnen, wenn auf Ihrem System eine amerikanische oder englische Version von Excel installiert ist. In anderen Ländern ist dieses Format jedoch nicht so weit verbreitet, weil Kommas häufig in numerischer Notation auftreten.

Wenn Sie nicht eine amerikanische/englische Version von Excel verwenden, gehen Sie folgendermaßen vor, um eine gespeicherte Protokolldatei zu öffnen:

1. Öffnen Sie Microsoft Excel und legen Sie eine leere Arbeitsmappe an.
2. Wählen Sie im Menü "Daten > Externe Daten importieren" den Befehl "Daten importieren".
3. Navigieren Sie zu der Protokolldatei, die Sie öffnen möchten, und wählen Sie sie aus. Der Textimport-Assistent wird aufgerufen.

12.9 Einschränkungen

4. Ändern Sie im Textimport-Assistenten die Standardoption für "Ursprünglicher Datentyp" von "Feste Breite" in "Getrennt".
5. Wählen Sie die Schaltfläche "Weiter".
6. Aktivieren Sie im Dialog "Schritt 2" das Kontrollkästchen "Komma", um den Trennzeichentyp von "Tab" in "Komma" zu ändern.
7. Wählen Sie die Schaltfläche "Weiter".
8. Im Dialog "Schritt 3" können Sie optional das Datenformat von MDJ (Monat/Tag/Jahr) in ein anderes Format ändern.
9. Führen Sie die übrigen Schritt im Textimport-Assistent durch, um die Datei zu importieren.

Kommunikationsprozessor und Modbus-TCP

13.1 Mit den seriellen Kommunikationsschnittstellen arbeiten

Zwei Kommunikationsmodule (CMs) und ein Kommunikationsboard (CB) bieten die Schnittstelle für die PtP-Kommunikation:

- CM 1241 RS232 (Seite 1427)
- CM 1241 RS422/485 (Seite 1428)
- CB 1241 RS485 (Seite 1424)

Sie können bis zu drei CMs (jeden Typs) plus ein CB, insgesamt vier Kommunikationsschnittstellen anschließen. Installieren Sie das CM links von der CPU oder einem anderen CM. Stecken Sie das CB auf der Vorderseite der CPU. Weitere Informationen zum Ein- und Ausbau von Modulen finden Sie in den Einbaurichtlinien (Seite 56).

Die seriellen Kommunikationsschnittstellen haben die folgenden Eigenschaften:

- Potentialgetrennter Anschluss
- Unterstützung von Punkt-zu-Punkt-Protokollen
- Konfiguration und Programmierung über die Anweisungen des Punkt-zu-Punkt-Kommunikationsprozessors
- Anzeige der Sende- und Empfangsaktivität über LEDs
- Diagnose-LED (nur CMs)
- Spannungsversorgung über die CPU: Keine externe Stromversorgung erforderlich.

Siehe technische Daten der Kommunikationsschnittstellen (Seite 1414).

LED-Anzeigen

Die Kommunikationsmodule haben drei LED-Anzeigen:

- Diagnose-LED (DIAG): Diese LED blinkt rot, bis sie von der CPU angesprochen wird. Nach dem Anlauf der CPU prüft diese auf CMs und adressiert diese. Die Diagnose-LED beginnt, grün zu blinken. Das bedeutet, dass die CPU das CM adressiert, ihm jedoch noch keine Konfiguration zugewiesen hat. Die CPU lädt die Konfiguration in die konfigurierten CMs, wenn das Programm in die CPU geladen wird. Nach dem Laden in die CPU muss die Diagnose-LED am Kommunikationsmodul dauerhaft grün leuchten.
- Sende-LED (Tx): Die Sende-LED leuchtet, wenn Daten über den Kommunikationsport gesendet werden.
- Empfangs-LED (Rx): Diese LED leuchtet, wenn Daten über den Kommunikationsport empfangen werden.

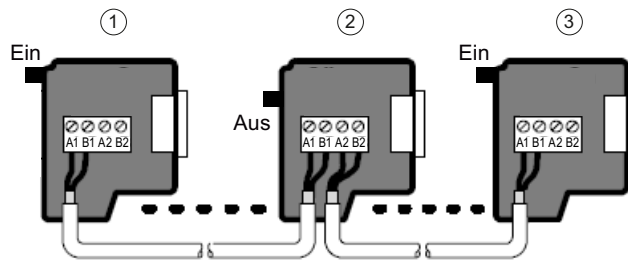
Das Kommunikationsboard bietet LEDs für Senden (TxD) und Empfangen (RxD). Es hat keine Diagnose-LED.

13.2 Abschließen eines RS485-Busanschlussteckers

Siemens bietet einen RS485-Busanschlusstecker (Seite 1443), mit dem Sie problemlos mehrere Geräte an ein RS485-Netz anschließen können. Der Stecker verfügt über zwei Klemmsätze, mit denen Sie die Eingangs- und Ausgangskabel für das Netz befestigen können. Der Stecker verfügt zudem über Schalter, um das Netz selektiv abzuschließen zu können.

Hinweis

Sie schließen nur die zwei Enden des RS485-Netzes ab. Die Geräte zwischen den beiden Endgeräten werden nicht abgeschlossen. Blanker Kabelschirm: Ca. 12 mm muss blank auf der Metallführung liegen.



- ① Schalterstellung = Ein: Abschlusswiderstand zugeschaltet
- ② Schalterstellung = Aus: Abschlusswiderstand nicht zugeschaltet
- ③ Schalterstellung = Ein: Abschlusswiderstand zugeschaltet

Tabelle 13-1 Abschluss des RS485-Steckers

Abschlussgerät (Abschluss EIN)	Kein Abschlussgerät (Abschluss AUS)

- ① Pinnummer
- ② Busanschlusstecker
- ③ Kabelschirm

Das CB 1241 bietet interne Widerstände für den Netzabschluss. Um die Verbindung abzuschließen, verbinden Sie TRA mit TA und TRB mit TB, um die internen Widerstände in den

Schaltkreis aufzunehmen. CB 1241 hat keinen 9-poligen Steckverbinder. Die folgende Tabelle zeigt die Anschlüsse an einen 9-poligen Steckverbinder am Kommunikationspartner.

Tabelle 13-2 Abschluss des CB 1241

Abschlussgerät (Abschluss EIN)	Kein Abschlussgerät (Abschluss AUS)

① M an den Kabelschirm anschließen

① A = TxD/RxD - (grüne Ader/Pol 8)

② B = TxD/RxD + (rote Ader/Pol 3)

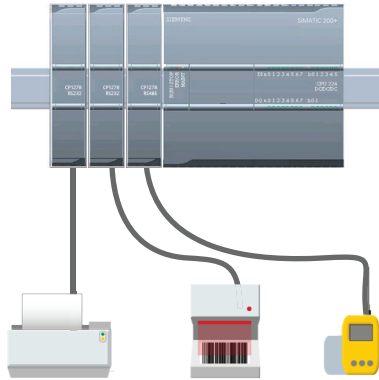
13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Die CPU unterstützt die folgende Punkt-zu-Punkt-Kommunikation (PtP) für zeichenbasierte serielle Protokolle.

- PtP, frei programmierbare Kommunikation (Seite 944)
- PtP, 3964(R) (Seite 946)
- USS (Seite 998)
- Modbus (Seite 1017)

13.3.1 PtP, frei programmierbare Kommunikation

Das Punkt-zu-Punkt-Protokoll für die frei programmierbare Kommunikation bietet maximale Freiheit und Flexibilität, erfordert jedoch eine aufwändige Implementierung im Anwenderprogramm.



PtP bietet eine Vielzahl an Möglichkeiten:

- Direktes Senden von Informationen an ein externes Gerät, wie z. B. einen Drucker
- Empfangen von Informationen von anderen Geräten, wie z. B. Strichcodelesern, RFID-Lesern, Kamera- oder Überwachungssystemen anderer Hersteller und vielen anderen Gerätearten
- Informationsaustausch, Senden und Empfangen von Daten mit anderen Geräten, wie z. B. GPS-Geräten, Kamera- oder Überwachungssystemen anderer Hersteller, Funkmodems und vielen anderen

Diese PtP-Kommunikation ist eine serielle Kommunikation mit Standard-UARTs, die verschiedene Baudraten und Paritäten unterstützen. Die RS232- und RS422/485-Kommunikationsmodule (CM 1241) sowie das RS485-Kommunikationsboard (CB 1241) stellen die elektrischen Schnittstellen für die PtP-Kommunikation zur Verfügung.

Frei programmierbare PtP-Kommunikation über PROFIBUS oder PROFINET

Mit PtP können Sie einen dezentralen PROFINET- oder PROFIBUS-Peripheriebaugruppenträgers für die Kommunikation mit verschiedenen Geräten (RFID-Lesegeräte, GPS und anderen) nutzen:

- PROFINET (Seite 598): Die Ethernet-Schnittstelle der S7-1200 CPU wird mit einem PROFINET Schnittstellenmodul verbunden. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.
- PROFIBUS (Seite 782): Ein PROFIBUS-Kommunikationsmodul wird an der linken Seite des Baugruppenträgers mit der S7-1200 CPU eingesteckt. Das PROFIBUS-Kommunikationsmodul wird mit einem Baugruppenträger verbunden, der ein PROFIBUS-Schnittstellenmodul enthält. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.

Die S7-1200 unterstützt daher zwei Sätze PtP-Anweisungen:

- Ältere Punkt-zu-Punkt-Anweisungen (Seite 1117): Diese Anweisungen wurden vor der Version V4.0 der S7-1200 benutzt und ermöglichen nur die serielle Kommunikation mit Hilfe eines Kommunikationsmoduls CM 1241 oder eines Kommunikationsboards CB 1241.
- Punkt-zu-Punkt-Anweisungen (Seite 963): Diese Anweisungen stellen alle Funktionen der alten Anweisungen bereit und zusätzlich die Möglichkeit zur Unterstützung von PtP-Kommunikationsmodulen über dezentrale PROFINET- und PROFIBUS-Peripherie. Über die Punkt-zu-Punkt-Anweisungen können Sie über den dezentralen Peripheriebaugruppenträger auf die Kommunikationsmodule zugreifen. Die S7-1200 CM 1241-Module benötigen mindestens die Firmwareversion V2.1, um die Punkt-zu-Punkt-Anweisungen verwenden zu können. Diese Module können nur im lokalen Baugruppenträger an der linken Seite der S7-1200 CPU angeschlossen werden. Die Punkt-zu-Punkt-Anweisungen können auch für einen CB 1241 verwendet werden. Bei der Kommunikation über dezentrale Peripherie werden die folgenden Module verwendet:

Station	Modul	Artikelnummer	Schnittstelle
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 und RS422/RS485

Hinweis

Mit den Punkt-zu-Punkt-Anweisungen können Sie auf ein Kommunikationsboard, lokale serielle Module (an der linken Seite), auf serielle Module über PROFINET und auf serielle Module über PROFIBUS zugreifen. STEP 7 stellt die alten PtP-Anweisungen nur zur Unterstützung vorhandener Programme bereit. Die alten Anweisungen funktionieren jedoch noch bei den aktuellen S7-1200 CPUs. Es ist nicht nötig, ältere Programme auf die neuen Anweisungen zu konvertieren.

Hinweis

Anforderung der CM-Firmware-Version für die Uhrzeitsynchronisation und PtP-Kommunikation

Wenn die Option "CPU synchronisiert die Module des Geräts" in den Eigenschaften Uhrzeitsynchronisation (Seite 172) für die Profinet-Schnittstelle in der Gerätekonfiguration aktiviert ist, aktualisieren Sie die Firmware-Versionen der angeschalteten Kommunikationsmodule auf die neuesten Versionen. Die Freigabe der Uhrzeitsynchronisation für Kommunikationsmodule mit älteren Firmware-Versionen kann zu Kommunikationsfehlern führen.

13.3.2 3964(R)-Kommunikation

Die S7-1200 CPU unterstützt das Protokoll 3964(R) für die Kommunikation zwischen einem CM 1241-Modul (RS-232) oder einem CM 1241-Modul (RS-422/485) und einem Kommunikationspartner, der das Protokoll 3964(R) verwendet. Im Gegensatz zur oben beschriebenen PtP-Kommunikation, bei der Sie spezifische Sende- bzw. Übertragungs- und Empfangseigenschaften für die Meldungen definieren, schreibt das Protokoll 3964(R) ein strenges Protokoll mit den folgenden Steuerzeichen vor:

- STX Start of Text
Start der zu übertragenden Zeichenfolge
- DLE Data Link Escape
Datenübertragungsumschaltung
- ETX End of Text
Ende der zu übertragenden Zeichenfolge
- BCC Block Check Character – Blockprüfzeichen
- NAK Negative Acknowledge – Negative Quittierung

Eine vollständige Beschreibung des Protokolls finden Sie in dem Kapitel des Handbuchs Handbuch S7-300 Punkt-zu-Punkt-Kopplung CP 341 Aufbauen und Parametrieren. (<https://support.industry.siemens.com/cs/de/de/view/1117397>), in dem die Prinzipien der seriellen Datenübertragung beschrieben werden.

Kommunikationsmodul konfigurieren

Um mit einem Partner über das Protokoll 3964(R) zu kommunizieren, müssen Sie eines der folgenden Kommunikationsmodule in Ihre Gerätekonfiguration in STEP 7 einschließen:

- CM 1241 (RS-232)
- CM 1241 (RS-422/485)

Die Firmware-Version des CM-Moduls muss V2.2.0 oder höher sein.

Für das Kommunikationsmodul konfigurieren Sie dann die Kommunikationsports (Seite 947), die Priorität und die Protokollparameter (Seite 961).

Kommunikation mit einem Partner über das Protokoll 3964(R)

Wenn Sie ein CM für das Protokoll 3964(R) konfigurieren, verwenden Sie die standardmäßigen Punkt-zu-Punkt-Anweisungen zum Senden und Empfangen, um Daten zwischen der CPU und dem Kommunikationspartner zu übertragen.

Das CM bündelt Ihre Daten aus dem Parameter BUFFER der Sendeanweisung in das Protokoll 3964(R) ein und sendet die Daten an den Kommunikationspartner.

Das CM empfängt Daten vom Kommunikationspartner über das Protokoll 3964(R), entfernt die Protokollinformationen und gibt die Daten im Parameter BUFFER der Empfangsanweisung aus.

Weitere Informationen finden Sie bei den folgenden Punkt-zu-Punkt-Anweisungen:

- Send_P2P (Sendepufferdaten übertragen) (Seite 977)
- Receive_P2P (Meldungsempfang aktivieren) (Seite 980)

Sie können auch die früheren Punkt-zu-Punkt-Anweisungen zum Senden und Empfangen verwenden:

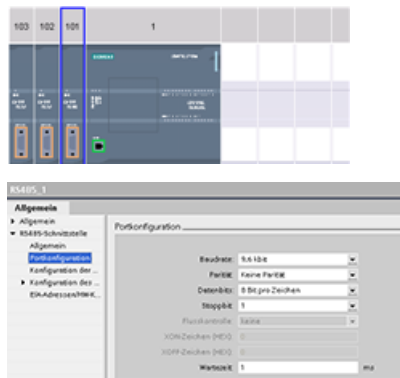
- SEND_PTP (Sendepufferdaten übertragen) (Seite 1125)
- RCV_PTP (Empfangsmeldungen aktivieren) (Seite 1128)

13.3.3 Konfigurieren der frei programmierbaren PtP-Kommunikation

Zur Konfiguration der Kommunikationsschnittstellen für die frei programmierbare PtP-Kommunikation sind die folgenden Vorgehensweisen möglich:

- Konfigurieren Sie in der Gerätekonfiguration in STEP 7 die Portparameter (Baudrate und Parität), die Sendeparameter und die Empfangsparameter. Die CPU speichert die Einstellungen in der Gerätekonfiguration und übernimmt die Einstellungen nach einem Neustart und einem Wechsel von RUN zu STOP.
- Richten Sie die Parameter über die Anweisungen Port_Config (Seite 965), Send_Config (Seite 968) und Receive_Config (Seite 970) ein. Die von den Anweisungen festgelegten Porteinstellungen sind gültig, wenn sich die CPU in der Betriebsart RUN befindet. Die Porteinstellungen kehren nach einem Wechsel in STOP oder nach einem Neustart zu den Einstellungen der Gerätekonfiguration zurück.

Nach der Konfiguration der Hardwaregeräte (Seite 135) parametrieren Sie die Kommunikationsschnittstellen durch Auswahl eines der CMs in Ihrem Baugruppenträger bzw. des CBs, sofern konfiguriert.



Im Inspektorfenster werden im Register "Eigenschaften" die Parameter des ausgewählten CMs oder CB angezeigt. Wählen Sie "Portkonfiguration", um die folgenden Parameter zu bearbeiten:

- Baudrate
- Parität
- Datenbits pro Zeichen
- Anzahl Stopbits
- Flusskontrolle (nur RS232)
- Wartezeit

Beim CM 1241 RS232 und beim CB RS485 (außer für die Flusskontrolle (Seite 949), die nur vom CM 1241 RS232 unterstützt wird) haben die RS232- und RS485-Kommunikationsmodule und das RS485-Kommunikationsboard die gleichen Parameter für die Portkonfiguration. Die Parameterwerte können verschieden sein.

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Beim CM 1241 RS422/485 stehen zusätzliche Optionen für die Portkonfiguration, wie unten dargestellt, zur Verfügung. Der 422-Modus des CM 1241 RS422/485 unterstützt auch die Software-Flusskontrolle.



Wählen Sie "Portkonfiguration", um die folgenden RS422/485-Parameter zu bearbeiten:

- "Betriebsart":
 - Vollduplex (RS422), Vierdrahtmodus (Punkt-zu-Punkt-Verbindung)
 - Vollduplex (RS422), Vierdrahtmodus (Multi-point-Master)
 - Vollduplex (RS422), Vierdrahtmodus (Multi-point-Slave)
 - Halbduplex (RS485), Zweidrahtmodus
- "Ausgangszustand Empfangsleitung":
 - Ohne
 - Vorspannung in Vorwärtsrichtung (Signal R(A) 0 V, Signal R(B) 5 V)

Das STEP 7 Anwenderprogramm kann auch den Port konfigurieren oder die vorhandene Konfiguration mit der Anweisung Port_Config (Seite 965) ändern. Die Beschreibung der Anweisung enthält genauere Angaben zu Betriebsart und Ausgangszustand der Leitung sowie weiteren Parametern.

Parameter	Definition
Baudrate	Die Voreinstellung für die Baudrate ist 9,6 kBit/s. Gültige Werte sind: 300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s, 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s und 115,2 kBit/s.
Parität	Die Voreinstellung für die Parität ist "Keine". Gültige Werte sind: Keine Parität, gerade, ungerade, Mark (Paritätsbit immer auf 1) und Space (Paritätsbit immer auf 0).
Datenbits pro Zeichen	Die Anzahl der Datenbits in einem Zeichen. Gültig sind die Einstellungen 7 oder 8.
Anzahl Stoppbits	Die Anzahl der Stoppbits kann eins oder zwei sein. Die Voreinstellung ist eins.
Flusskontrolle	Für das RS232-Kommunikationsmodul kann die Hardware- oder Software-Flusskontrolle (Seite 949) gewählt werden. Ist die Hardware-Flusskontrolle eingestellt, so können Sie wählen, ob das RTS-Signal immer ein ist oder ob RTS geschaltet wird. Ist die Software-Flusskontrolle eingestellt, können Sie die Zeichen XON und XOFF definieren. Die RS485-Kommunikationsschnittstellen unterstützen keine Flusskontrolle. Der 422-Modus des CM 1241 RS422/485 unterstützt die Software-Flusskontrolle.
Wartezeit	Die Wartezeit ist die Zeit, während der das CM oder CB nach dem Absetzen von RTS auf den Empfang von CTS oder nach dem Empfang von XOFF auf den Empfang von XON, je nach Art der Flusskontrolle, wartet. Ist die Wartezeit beendet, bevor die Kommunikationsschnittstelle ein erwartetes CTS oder XON empfängt, so wird die Sendung durch das CM oder CB abgebrochen und eine Fehlermeldung an das Anwenderprogramm ausgegeben. Geben Sie die Wartezeit in Millisekunden ein. Der zulässige Wertebereich ist 0 bis 65535 Millisekunden.
Betriebszustand	Hier wählen Sie den Betriebszustand RS422 oder RS485 und die Netzwerkkonfigurationen aus.
Ausgangszustand Empfangsleitung	Hier wählen Sie die Vorspannung aus. Gültig sind die Werte Ohne, Vorspannung in Vorwärtsrichtung und Vorspannung in Sperrrichtung. Vorspannung in Sperrrichtung ermöglicht die Drahtbrucherkennung.

13.3.3.1 Steuerung der Flusskontrolle

Die Flusskontrolle sorgt für den Ausgleich zwischen unterschiedlichen Kapazitäten von Absender und Empfänger der Daten, so dass keine Daten verlorengehen. Mit der Flusskontrolle ist gewährleistet, dass der Absender nicht mehr Informationen überträgt als der Empfänger verarbeiten kann. Die Flusskontrolle ist auf der Hardware- oder der Softwareebene möglich. Das RS232-Kommunikationsmodul unterstützt die Hardware- und Software-Flusskontrolle. Die RS485-CM und -CB unterstützen keine Flusskontrolle. Der 422-Modus des CM 1241 RS422/485 unterstützt die Software-Flusskontrolle. Welche Flusskontrolle eingerichtet werden soll, geben Sie beim Konfigurieren der Schnittstelle (Seite 947) oder über die Anweisung PORT_CFG (Seite 1117) ein.

Die Hardware-Flusskontrolle beruht auf den Kommunikationssignalen RTS (Request-to-send) und CTS (Clear-to-send). Beim RS232-Kommunikationsmodul wird das RTS-Signal an Pin 7 und das CTS-Signal an Pin 8 abgenommen. Das RS232-Kommunikationsmodul ist ein DTE-Gerät (Data Terminal Equipment, Datenendgerät), das RTS als Ausgang sicherstellt und CTS als Eingang überwacht.

Hardware-Flusskontrolle: RTS-geschaltet

Ist die RTS-geschaltete Hardware-Flusskontrolle für ein RS232-Kommunikationsmodul aktiviert, setzt das Modul das RTS-Signal auf 1, damit Daten gesendet werden. Es überwacht das CTS-Signal und erkennt damit, ob das Empfangsgerät empfangsbereit ist. Ist das CTS-Signal aktiv, kann das Modul Daten übertragen, solange das CTS-Signal aktiv bleibt. Wird das CTS-Signal inaktiv, muss die Übertragung angehalten werden.

Sie wird fortgesetzt, wenn das CTS-Signal erneut eingeschaltet wird. Wird das CTS-Signal nicht innerhalb der eingestellten Wartezeit erneut eingeschaltet, bricht das Modul die Übertragung ab und gibt einen Fehler an das Anwenderprogramm zurück. Die Wartezeit müssen Sie in der Schnittstellenkonfiguration (Seite 947) eingeben.

Die RTS-geschaltete Flusskontrolle ist bei Geräten nützlich, die ein Signal benötigen, dass der Sendevorgang aktiv ist. Ein Beispiel hierfür ist ein Funkmodem, bei dem RTS als "Schlüssel"signal den Funksender aktiviert. Die RTS-geschaltete Flusskontrolle funktioniert nicht bei herkömmlichen Telefonmodems. Verwenden Sie für Telefonmodems die Option "RTS immer ein".

Hardware-Flusskontrolle: RTS immer ein

Bei der Option "RTS immer ein" aktiviert das CM 1241 das RTS-Signal standardmäßig. Ein Gerät, wie z. B. ein Telefonmodem, überwacht das RTS-Signal vom CM und nutzt dieses Signal als Mitteilung der Sendebereitschaft. Das Modem sendet nur dann an das CM, wenn RTS aktiv ist, d. h. wenn das Telefonmodem ein aktives CTS erkennt. Ist RTS inaktiv, überträgt das Telefonmodem keine Daten zum CM.

Damit das Modem jederzeit Daten an das CM senden kann, ist für die Hardware-Flusskontrolle "RTS immer ein" einzustellen. Das CM setzt das RTS-Signal dann immer auf 1. Das CM schaltet RTS auch dann nicht aus, wenn das Modul keine Zeichen empfangen kann. Das Sendegerät muss gewährleisten, dass es keinen Überlauf im Empfangspuffer des CMs verursacht.

Signale Data Terminal Ready (DTR) und Data Set Ready (DSR)

Das CM setzt DTR für jede Art der Hardware-Flusskontrolle auf 1. Es überträgt erst, wenn das Signal DSR eingeschaltet wird. Der Zustand von DSR wird nur am Anfang der Sendefunktion ausgewertet. Wenn DSR nach Beginn der Übertragung inaktiv wird, wird die Übertragung nicht angehalten.

Software-Flusskontrolle

Bei der Software-Flusskontrolle werden spezielle Zeichen in den Meldungen für die Flusskontrolle verwendet. Sie können Hexadezimalzeichen konfigurieren, die XON und XOFF darstellen.

XOFF zeigt an, dass eine Übertragung angehalten werden muss. XON zeigt an, dass eine Übertragung fortgesetzt werden kann. XOFF und XON dürfen nicht das gleiche Zeichen sein.

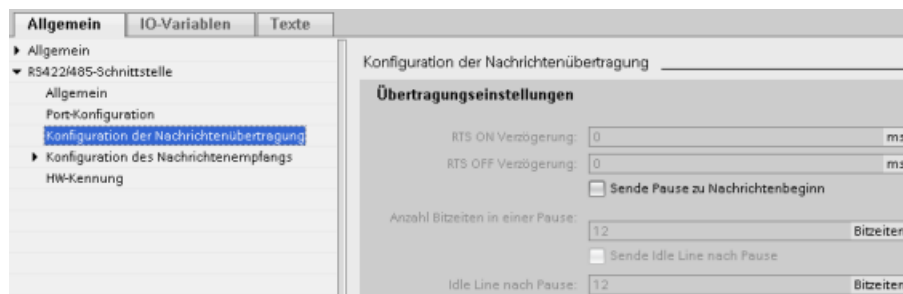
Empfängt das Sendegerät ein Zeichen XOFF vom Empfänger, wird die Übertragung angehalten. Sie wird fortgesetzt, wenn das Sendegerät das Zeichen XON empfängt. Empfängt das CM innerhalb der in der Schnittstellenkonfiguration (Seite 947) angegebenen Wartezeit kein XON-Zeichen, bricht es die Übertragung ab und gibt eine Fehlermeldung an das Anwenderprogramm aus.

Die Software-Flusskontrolle setzt die Vollduplex-Kommunikation voraus, da der Empfänger in der Lage sein muss, während einer laufenden Übertragung XOFF an den Sender zu senden. Die Software-Flusskontrolle ist nur bei Meldungen möglich, die nur aus ASCII-Zeichen bestehen. Binäre Protokolle können keine Software-Flusskontrolle nutzen.

Damit die frei programmierbare Punkt-zu-Punkt-Kommunikation aufgenommen werden kann, müssen die Parameter zum Senden und Empfangen von Meldungen eingerichtet werden. Diese Parameter legen fest, wie die Kommunikation abläuft, wenn Meldungen von einem Zielgerät gesendet oder von diesem empfangen werden.

13.3.3.2 Sendeparameter konfigurieren

In der Gerätekonfiguration der CPU richten Sie ein, wie eine Kommunikationsschnittstelle Daten sendet. Hierfür geben Sie die Eigenschaften "Konfiguration Sendemeldung" für die ausgewählte Schnittstelle an.



Sie können ferner die Parameter für das Senden einer Meldung im Anwenderprogramm mit der Anweisung `Send_Config` (Seite 968) dynamisch konfigurieren oder ändern.

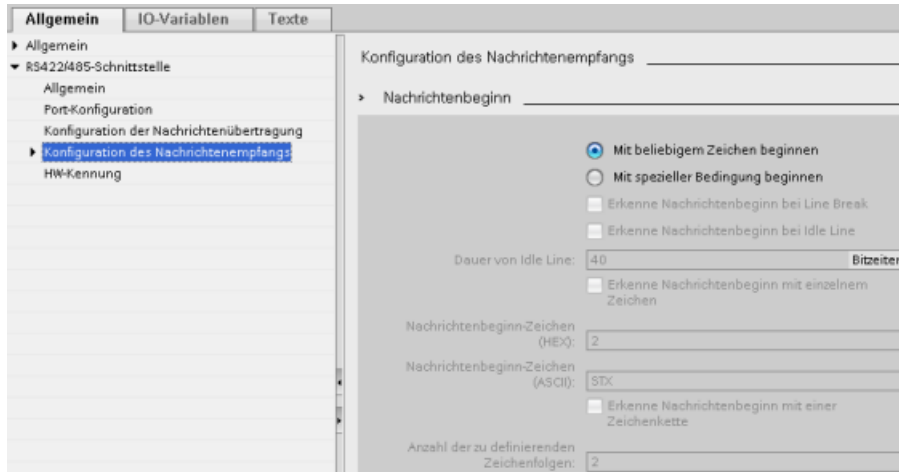
Hinweis

Parameterwerte, die mit `Send_Config` im Anwenderprogramm eingegeben wurden, überschreiben die Eigenschaften "Konfiguration Sendemeldung". Parameter, die mit der Anweisung `Send_Config` eingegeben wurden, werden bei einer Abschaltung nicht in der CPU gespeichert.

Parameter	Definition
RTS-Einschaltverzögerung	Wartezeit nach dem Aktivieren von RTS, bis die Sendung ausgelöst wird. Der gültige Bereich ist 0 bis 65535 ms, voreingestellt ist 0. Dieser Parameter ist nur gültig, wenn in der Schnittstellenkonfiguration (Seite 947) die Hardware-Flusskontrolle eingestellt ist. CTS wird nach dem Ablauf der RTS-Einschaltverzögerung bearbeitet. Dieser Parameter gilt nur für RS232-Module.
RTS-Ausschaltverzögerung	Wartezeit nach der beendeten Übertragung, bis RTS deaktiviert wird. Der gültige Bereich ist 0 bis 65535 ms, voreingestellt ist 0. Dieser Parameter ist nur gültig, wenn in der Schnittstellenkonfiguration (Seite 947) die Hardware-Flusskontrolle eingestellt ist. Dieser Parameter gilt nur für RS232-Module.
Sendepause bei Meldungsbeginn Anzahl der Bittakte in einer Pause	Damit wird festgelegt, dass beim Start einer Meldung eine Pause gesendet wird, wenn die RTS-Einschaltverzögerung (sofern eingerichtet) abgelaufen ist und CTS aktiv ist. Wie viele Bittakte eine Pause darstellen, während der die Übertragung angehalten ist, können Sie einstellen. Die Voreinstellung ist 12 und das Maximum ist 65535 bis zu einem Grenzwert von acht Sekunden.
Leerzeile nach Pause senden Leerzeile nach Pause	Gibt an, dass eine Leerzeile vor Meldungsbeginn gesendet wird. Sie wird nach der Pause gesendet, sofern eine Pause konfiguriert ist. Der Parameter "Leerzeile nach Pause" gibt an, wie viele Bittakte einer Leerzeile entsprechen. Die Voreinstellung ist 12 und das Maximum ist 65535 bis zu einem Grenzwert von acht Sekunden.

13.3.3.3 Empfangsparameter konfigurieren

In der Gerätekonfiguration der CPU legen Sie fest, wie Daten über eine Kommunikationsschnittstelle empfangen und wie Meldungsbeginn und Meldungsende erkannt werden sollen. Sie geben diese Parameter in den Eigenschaften "Konfiguration Empfangsmeldung" für die ausgewählte Schnittstelle an.



Sie können ferner die Meldungsparameter im Anwenderprogramm mit der Anweisung `Receive_Config` (Seite 970) dynamisch konfigurieren oder ändern.

Hinweis

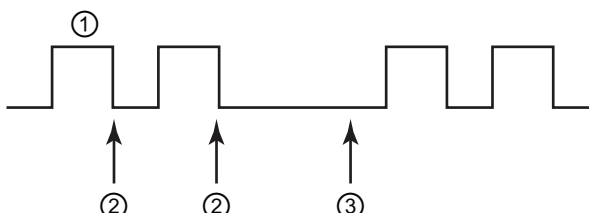
Parameterwerte, die mit `Receive_Config` im Anwenderprogramm eingegeben wurden, überschreiben die Eigenschaften "Konfiguration Empfangsmeldung". Parameter, die mit der Anweisung `RCV_CFG` eingegeben wurden, werden bei einer Abschaltung oder einem Wechsel in STOP nicht in der CPU gespeichert.

Bedingungen für den Meldungsbeginn

Sie können festlegen, wie die Kommunikationsschnittstelle den Meldungsbeginn erkennen soll. Startzeichen und Meldungsinhalt werden im Empfangspuffer abgelegt, bis eine konfigurierte Endebedingung erfüllt ist.

Sie können mehrere Startbedingungen angeben. Wenn Sie mehrere Startbedingungen angeben, müssen alle Startbedingungen erfüllt sein, damit die Meldung als gestartet betrachtet wird. Wenn Sie z. B. eine Zeit für den Leitungleerlauf und ein bestimmtes Startzeichen konfigurieren, wartet das CM bzw. CB zunächst auf die Erfüllung der Zeit für den Leitungleerlauf und sucht dann nach dem spezifischen Startzeichen. Wenn ein anderes Zeichen

empfangen wird (nicht das angegebene Startzeichen), startet das CM bzw. CB die Suche nach dem Meldungsstart erneut, indem es auf die Zeit für den Leitungslieferlauf wartet.

Parameter	Definition
Start mit beliebigem Zeichen	Die Bedingung "Beliebiges Zeichen" gibt an, dass durch erfolgreichen Empfang eines beliebigen Zeichens ein Meldungsbeginn angezeigt wird. Dieses Zeichen ist das erste Zeichen in einer Meldung.
Leitungspause	Die Bedingung "Leitungspause" gibt an, dass eine Operation zum Meldungsempfang beginnt, nachdem ein Pausezeichen empfangen wurde.
Leitung im Leerlauf	<p>Die Bedingung "Leitung im Leerlauf" gibt an, dass der Meldungsempfang beginnt, wenn die Empfangsleitung für die Anzahl der angegebenen Bitzeiten im Leerlauf war. Nachdem diese Bedingung aufgetreten ist, beginnt der Empfang einer Meldung.</p>  <p>① Zeichen ② Startet die Zeit für den Leitungslieferlauf neu ③ Leitungslieferlauf wird erkannt und Meldungsempfang gestartet</p>
Sonderbedingung: Meldungsbeginn durch einzelnes Zeichen erkennen	Gibt an, dass ein bestimmtes Zeichen den Beginn einer Meldung anzeigt. Dieses Zeichen ist dann das erste Zeichen in einer Meldung. Jedes Zeichen, das vor diesem bestimmten Zeichen empfangen wird, wird ignoriert. Das Standardzeichen ist STX.
Sonderbedingung: Meldungsbeginn durch Zeichenfolge erkennen	<p>Gibt an, dass eine bestimmte Zeichenfolge aus bis zu vier konfigurierten Folgen den Beginn einer Meldung anzeigt. Sie können bis zu fünf Zeichen für jede Zeichenfolge eingeben. Für jede Zeichenposition kann ein bestimmtes Hex-Zeichen oder das Ignorieren des Zeichens in der Zeichenfolge (Platzhalterzeichen) eingegeben werden. Das letzte spezifische Zeichen einer Zeichenfolge beendet die Startbedingungsfolge.</p> <p>Eingehende Zeichenfolgen werden mit den konfigurierten Startbedingungen verglichen, bis eine Startbedingung erfüllt ist. Ist die Startbedingung erfüllt, beginnt die Überwachung der Endbedingung.</p> <p>Sie können bis zu vier spezifische Zeichenfolgen konfigurieren. Sie verwenden eine Startbedingung mit mehreren Folgen, wenn verschiedene Zeichenfolgen den Beginn einer Meldung anzeigen können. Die Meldung beginnt, wenn eine der eingerichteten Zeichenfolgen erfüllt ist.</p>

Die Startbedingungen werden in der folgenden Reihenfolge geprüft:

- Leerzeile
- Zeilenumbruch
- Zeichen oder Zeichenfolgen

Wenn mehrere Startbedingungen geprüft werden und eine der Bedingungen nicht erfüllt wird, beginnt das CM bzw. CB die Prüfung erneut mit der ersten geforderten Bedingung. Nachdem das CM bzw. CB ermittelt hat, dass die Startbedingungen erfüllt wurden, beginnt es mit dem Auswerten der Endbedingungen.

Beispielkonfiguration: Meldungsbeginn bei einer von zwei Zeichenfolgen

Beachten Sie die folgende Konfiguration der Bedingungen für den Meldungsbeginn:

Erkenne Nachrichtenbeginn mit einer Zeichenkette

Anzahl der zu definierenden Zeichenfolgen:

5-Zeichen-Sequenz zum Nachrichtenbeginn

Nachrichtenbeginn Sequenz 1

Prüfe Zeichen 1

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 2

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 3

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 4

Zeichenwert (HEX):

Zeichenwert (ASCII):

Prüfe Zeichen 5

Zeichenwert (HEX):

Zeichenwert (ASCII):

Nachrichtenbeginn Sequenz 2

Prüfe Zeichen 1

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 2

Zeichenwert (HEX): 6A

Zeichenwert (ASCII): |

Prüfe Zeichen 3

Zeichenwert (HEX): 6A

Zeichenwert (ASCII): |

Prüfe Zeichen 4

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Prüfe Zeichen 5

Zeichenwert (HEX): 0

Zeichenwert (ASCII): BELIEBIG

Bei dieser Konfiguration ist die Startbedingung erfüllt, wenn eines der folgenden Bitmuster vorliegt:

- Empfang einer Zeichenfolge mit 5 Zeichen, bei der das erste Zeichen 0x6A und das fünfte Zeichen 0x1C ist. Die Zeichen 2, 3 und 4 können hierbei beliebig sein. Nach dem Empfang des fünften Zeichens beginnt die Überwachung auf die Endbedingung.
- Empfang von zwei aufeinanderfolgenden Zeichen 0x6A nach einem beliebigen anderen Zeichen. In diesem Fall beginnt die Überwachung auf die Endbedingung nach dem Empfang der zweiten 0x6A-Folge (3 Zeichen). Das Zeichen vor dem ersten 0x6A ist in der Startbedingung enthalten.

Beispiele für Zeichenfolgen, die diese Startbedingung erfüllen:

- <beliebiges Zeichen> 6A 6A
- 6A 12 14 18 1C
- 6A 44 A5 D2 1C

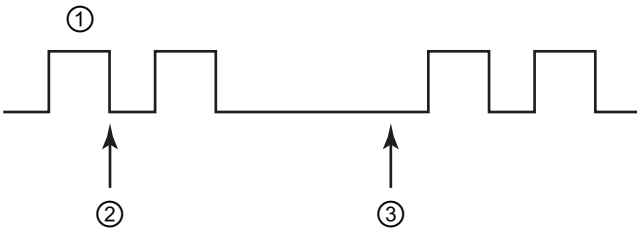
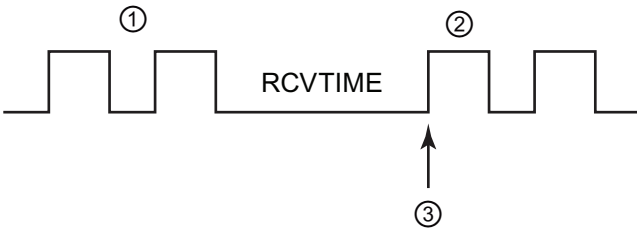
Bedingungen für das Meldungsende

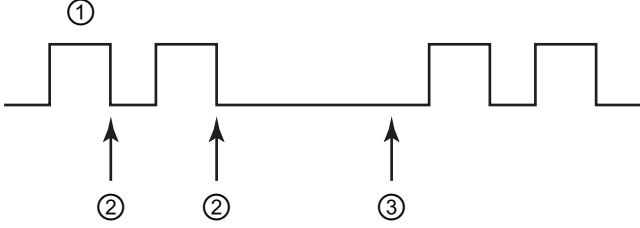
Sie legen ferner fest, wie die Kommunikationsschnittstelle das Meldungsende erkennen soll. Sie können mehrere Endbedingungen für Meldungen einrichten. Die Meldung endet, sobald eine der eingerichteten Bedingungen erfüllt ist.

Sie können z. B. eine Endbedingung mit einem Meldungs-Timeout von 300 ms, einem Zeichenabstands-Timeout von 40 Bitzeiten und eine maximale Länge von 50 Bytes angeben. Die Meldung endet, wenn der Meldungsempfang länger als 300 ms dauert oder wenn der Abstand

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

zwischen zwei beliebigen Zeichen 40 Bitzeiten überschreitet oder wenn 50 Bytes empfangen wurden.

Parameter	Definition
Meldungsende erkennen durch Meldungs-Timeout	<p>Das Meldungsende ist erreicht, wenn die konfigurierte Wartezeit auf das Meldungsende abgelaufen ist. Die Timeout-Zeit der Meldung beginnt, wenn eine Startbedingung erfüllt wurde. Die Voreinstellung ist 200 ms, der Wertebereich ist 0 bis 65535 ms.</p>  <p>① Empfangene Zeichen ② Bedingung für den Meldungsbeginn erfüllt: Meldungs-Timer beginnt zu laufen ③ Meldungs-Timer läuft ab und beendet die Meldung</p>
Meldungsende erkennen durch Antwort-Timeout	<p>Das Meldungsende ist erreicht, wenn die konfigurierte Wartezeit auf eine Antwort erreicht ist, bevor eine gültige Startzeichenfolge empfangen wurde. Die Timeout-Zeit für die Antwort beginnt, wenn eine Übertragung endet und das CM bzw. CB die Empfangsanweisung startet. Die Voreinstellung für die Timeout-Zeit der Antwort beträgt 200 ms, der Wertebereich ist 0 bis 65535 ms. Wenn nicht innerhalb des von RCVTIME angegebenen Zeitraums für die Antwort ein Zeichen empfangen wird, wird der entsprechenden Anweisung RCV_PTP ein Fehler gemeldet. Das Antwort-Timeout definiert keine spezifische Endebedingung. Das Timeout gibt nur an, dass ein Zeichen innerhalb der angegebenen Zeit erfolgreich empfangen werden muss. Sie müssen eine andere Endebedingung konfigurieren, um das tatsächliche Ende einer Meldung anzugeben.</p>  <p>① Gesendete Zeichen ② Empfangene Zeichen ③ Erstes Zeichen muss zu diesem Zeitpunkt erfolgreich empfangen worden sein.</p>

Parameter	Definition
Meldungsende durch Zeichenabstand erkennen	<p>Das Meldungsende ist erreicht, wenn die konfigurierte maximale Zeit zwischen zwei beliebigen, aufeinanderfolgenden Zeichen einer Meldung abgelaufen ist. Die Voreinstellung für den Zeichenabstand ist 12 Bittakte und die maximale Zahl ist 65535 Bittakte, bis maximal acht Sekunden.</p>  <p>① Empfangene Zeichen ② Startet den Timer für die Zeit zwischen den Zeichen neu ③ Der Timer für die Zeit zwischen den Zeichen läuft ab und beendet die Meldung.</p>
Meldungsende durch Empfang einer festen Anzahl von Zeichen erkennen	<p>Das Meldungsende ist erreicht, wenn die konfigurierte Anzahl von Zeichen empfangen wurde. Der gültige Bereich für die feste Länge liegt zwischen 1 und 4096.</p> <p>Beachten Sie, dass diese Bedingung für das Meldungsende bei der S7-1200 erst ab CPUs der Version V4.0 gültig ist.</p>
Meldungsende durch maximale Meldungslänge erkennen	<p>Das Meldungsende ist erreicht, wenn die konfigurierte maximale Zeichenzahl empfangen wurde. Der gültige Bereich für die maximale Länge liegt zwischen 1 und 1024.</p> <p>Diese Bedingung kann dafür genutzt werden, den Fehler "Überlauf Meldungspuffer" zu verhindern. Wird diese Endebedingung mit Timeout-Endebedingungen verknüpft und tritt die Timeout-Endebedingung auf, werden alle gültigen empfangenen Zeichen ausgegeben, auch wenn die maximale Länge nicht erreicht wurde. So können Protokolle unterschiedlicher Länge unterstützt werden, auch wenn nur die maximale Länge bekannt ist.</p>
Meldungslänge aus Meldung lesen	<p>Die Meldungslänge ist in der Meldung selbst angegeben. Das Meldungsende ist erreicht, wenn eine Meldung mit der angegebenen Länge empfangen wurde. Wie die Meldungslänge angegeben und ausgewertet wird, ist nachstehend beschrieben.</p>
Meldungsende mit einem Zeichen erkennen	<p>Das Meldungsende ist erreicht, wenn ein bestimmtes Zeichen empfangen wird.</p>
Meldungsende mit einer Zeichenfolge erkennen	<p>Das Meldungsende ist erreicht, wenn eine bestimmte Zeichenfolge empfangen wird. Sie können eine Zeichenfolge mit bis zu 5 Zeichen angeben. Für jede Zeichenposition kann ein bestimmtes Hex-Zeichen oder dass das Zeichen in der Zeichenfolge ignoriert werden soll eingegeben werden.</p> <p>Führende Zeichen, die ignoriert werden, gehören nicht zur Endebedingung. Nachgestellte Zeichen, die ignoriert werden, gehören zur Endebedingung.</p>

Beispielkonfiguration: Meldung mit einer Zeichenfolge beenden

Beachten Sie die folgende Konfiguration der Bedingungen für das Meldungsende:

Die Endebedingung ist in diesem Fall erfüllt, wenn zwei aufeinanderfolgende Zeichen 0x6A, gefolgt von zwei beliebigen Zeichen, empfangen werden. Das Zeichen vor 0x6A 0x6A gehört nicht zur Endezeichenfolge. Nach 0x6A 0x6A sind zwei Zeichen erforderlich, um die Endezeichenfolge abzuschließen. Die von Zeichen 4 und 5 empfangenen Werte sind irrelevant, sie müssen jedoch empfangen werden, damit die Endebedingung erfüllt ist.

Hinweis

Soll Ihre Zeichenfolge das Ende der Meldung anzeigen, setzen Sie sie an die letzte Zeichenposition. Im Beispiel oben würde 0x6A an den Zeichenpositionen 4 und 5 konfiguriert werden, wenn 0x6A 0x6A die Meldung ohne nachgestellte Zeichen beenden soll.

Angabe der Meldungslänge in der Meldung

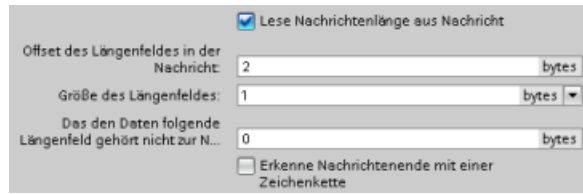
Bei der Auswahl der Sonderbedingung, bei der die Meldungslänge in der Meldung enthalten ist, müssen drei Parameter angegeben werden, die Informationen über die Meldungslänge enthalten.

Die tatsächliche Meldungsstruktur hängt vom verwendeten Protokoll ab. Die drei Parameter sind:

- n: Zeichenposition (Basis 1) innerhalb der Meldung, an der die Längenangabe beginnt
- Längenwert: Anzahl der Bytes (eins, zwei oder vier) der Längenangabe
- Länge m: Anzahl der Zeichen nach der Längenangabe, die nicht in der Längenzählung berücksichtigt werden

Die Endezeichen müssen nicht aufeinander folgen. Der Wert "Länge m" kann genutzt werden, um die Länge eines Prüfsummenfelds anzugeben, dessen Größe nicht im Längenfeld enthalten ist.

Diese Felder erscheinen in der Konfiguration des Meldungsempfangs in den Geräteeigenschaften:



Beispiel 1: Es wird eine Meldung betrachtet, die nach dem folgenden Protokoll strukturiert ist:

STX	Len (n)	Zeichen 3 bis 14, nach Länge gezählt											
		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

Die Parameter für die Länge des Meldungsempfangs konfigurieren Sie wie folgt:

- n = 2 (Meldungslänge startet in Byte 2.)
- Längenwert = 1 (Meldungslänge ist in einem Byte enthalten.)
- Länge m = 0 (Auf die Längenangabe folgen keine weiteren Zeichen, die nicht in der Längenzählung berücksichtigt werden. Nach der Längenangabe folgen zwölf Zeichen.)

In diesem Beispiel werden die Zeichen von 3 bis einschließlich 14 von Len (n) gezählt.

Beispiel 2: Es wird eine weitere Meldung betrachtet, die nach dem folgenden Protokoll strukturiert ist:

SD1	Len (n)	Len (n)	SD2	Zeichen 5 bis 10, nach Länge gezählt						FCS	ED
				DA	SA	FA	Dateneinheit=3 Byte				
1	2	3	4	5	6	7	8	9	10	11	12
xx	0x06	0x06	xx	xx	xx	xx	xx	xx	xx	xx	xx

Die Parameter für die Länge des Meldungsempfangs konfigurieren Sie wie folgt:

- n = 3 (Meldungslänge startet in Byte 3.)
- Längenwert = 1 (Meldungslänge ist in einem Byte enthalten.)
- Länge m = 3 (Auf die Längenangabe folgen drei Zeichen, die nicht in der Längenzählung berücksichtigt werden. Im Protokoll dieses Beispiels werden Zeichen SD2, FCS und ED bei der Längenzählung nicht beachtet. Die sechs anderen Zeichen werden in der Längenzählung berücksichtigt; die Gesamtzahl der Zeichen nach der Längenangabe ist somit neun.)

In diesem Beispiel werden die Zeichen von 5 bis einschließlich 10 von Len (n) gezählt.

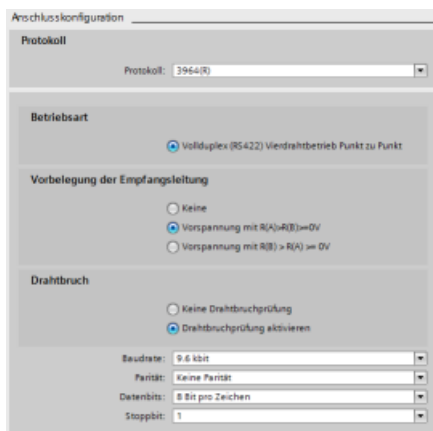
13.3.4 Konfigurieren der 3964(R)-Kommunikation

13.3.4.1 Konfigurieren der 3964(R)-Kommunikationsports

Zur Konfiguration der Kommunikationsschnittstellen für die 3964(R)-Kommunikation sind die folgenden Vorgehensweisen möglich:

- Konfigurieren Sie die Portparameter in der Gerätekonfiguration in STEP 7. Die CPU speichert die Einstellungen in der Gerätekonfiguration und übernimmt die Einstellungen nach einem Neustart.
- Legen Sie die Portparameter über die Anweisung Port_Config (Seite 965) fest. Die von den Anweisungen festgelegten Porteinstellungen sind gültig, wenn sich die CPU in der Betriebsart RUN befindet. Die Porteinstellungen kehren nach einem Neustart zu den Einstellungen der Gerätekonfiguration zurück.

Nach dem Hinzufügen der Kommunikationsschnittstellen zur Gerätekonfiguration (Seite 139) parametrieren Sie die Kommunikationsschnittstellen durch Auswahl eines der CMs in Ihrem Baugruppenträger.



Im Inspektorfenster werden im Register "Eigenschaften" die Parameter des ausgewählten CMs angezeigt. Wählen Sie "Portkonfiguration", um die folgenden Parameter zu bearbeiten:

- Protokoll: 3964(R)
- Betriebszustand (nur Modul CM 1241 (RS-422/485))
- Ausgangszustand Empfangsleitung (nur Modul CM 1241 (RS-422/485))
- Drahtbruch (nur Modul CM 1241 (RS-422/485))
- Baudrate
- Parität
- Datenbits
- Stoppbits

Parameter	Definition
Protokoll	3964R oder frei programmierbar. Wählen Sie 3964R, um den Port für die 3964(R)-Kommunikation zu konfigurieren.
Betriebszustand*	Vollduplex (RS-422), Vierdrahtbetrieb, Punkt-zu-Punkt. (Aktiviert)
Ausgangszustand Empfangsleitung*	Aktivieren Sie eine der folgenden Optionen: <ul style="list-style-type: none"> • Keiner • Vorspannung mit $R(A) > R(B) \geq 0V$ • Vorspannung mit $R(B) > R(A) \geq 0V$

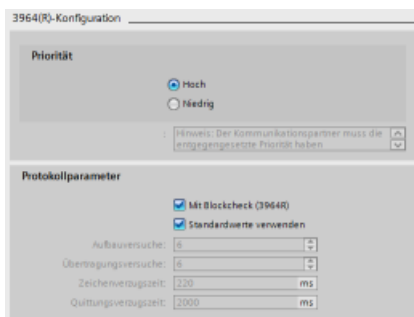
Parameter	Definition
Drahtbruch*	Aktivieren Sie eine der folgenden Optionen: <ul style="list-style-type: none"> Keine Drahtbruchprüfung Drahtbruchprüfung aktivieren
Baudrate	Die Voreinstellung für die Baudrate ist 9,6 kBit/s. Gültige Werte sind: 300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s, 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s und 115,2 kBit/s.
Parität	Die Voreinstellung für die Parität ist "Keine". Gültige Werte sind: Keine Parität, gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0), und beliebige Parität (Paritätsbit für Übertragung auf 0 setzen; Paritätsfehler beim Empfang ignorieren).
Datenbits pro Zeichen	Die Anzahl der Datenbits in einem Zeichen. Gültig sind die Einstellungen 7 oder 8.
Anzahl Stoppbits	Die Anzahl der Stoppbits kann eins oder zwei sein. Die Voreinstellung ist eins.

* Nur Modul CM 1241 (RS-422/485)

13.3.4.2 Konfigurieren von 3964(R)-Priorität und Protokollparametern

Zur Konfiguration der Kommunikationsschnittstellen für die 3964(R)-Kommunikation sind die folgenden Vorgehensweisen möglich:

- Klicken Sie in der Gerätekonfiguration der Kommunikationsschnittstelle auf "Konfiguration 3964(R)", um die Priorität festzulegen und die Protokollparameter zu konfigurieren. Die CPU speichert die Einstellungen in der Gerätekonfiguration und übernimmt die Einstellungen nach einem Neustart.
- Legen Sie mit der Anweisung P3964_Config (Seite 975) die Priorität fest und konfigurieren Sie die Protokollparameter. Die von den Anweisungen festgelegten Werte sind gültig, wenn sich die CPU in der Betriebsart RUN befindet. Die Werte kehren nach einem Neustart zu den Einstellungen der Gerätekonfiguration zurück.



Im Inspektorfenster werden im Register "Eigenschaften" die Parameter des ausgewählten CMs angezeigt. Wählen Sie "Konfiguration 3964(R)", um die folgenden Parameter zu bearbeiten:

- Priorität (hoch oder niedrig)
- Protokollparameter
 - Mit Blockprüfung (3964R)
 - Standardwerte verwenden
 - Verbindungsversuche
 - Übertragungsversuche
 - Zeichenverzugszeit
 - Quittierungsverzögerung

Parameter	Definition
Priorität	Hoch oder niedrig: Das CM hat entweder hohe oder niedrige Priorität und der Kommunikationspartner muss die entgegengesetzte Priorität haben.
Mit Blockprüfung (3964)	Wenn ausgewählt, wird die Übertragungssicherheit bei der 3964(R)-Kommunikation durch Einschließen eines Blockprüfzeichens (BCC) angewendet. Ist die Option nicht ausgewählt, wird bei der Übertragungssicherheit kein Blockprüfzeichen eingeschlossen.
Standardwerte verwenden	Wenn ausgewählt, verwendet 3964(R) für die folgenden Protokollparameter Standardwerte: <ul style="list-style-type: none"> • Verbindungsversuche • Übertragungsversuche • Zeichenverzugszeit • Quittierungsverzögerung Ist die Option nicht ausgewählt, können Sie für jeden dieser Parameter Werte konfigurieren.
Verbindungsversuche	Anzahl der Verbindungsversuche (Standardwert: 6 Verbindungsversuche) 1 bis 255
Übertragungsversuche	Anzahl der Übertragungsversuche (Standardwert: 6 Verbindungsversuche) 1 bis 255
Zeichenverzugszeit	Einstellung der Zeichenverzugszeit (abhängig von der festgelegten Datenübertragungsrate) (Standardwert: 220 ms) 1 ms bis 65535 ms
Quittierungsverzögerung	Einstellung der Quittierungsverzugszeit (abhängig von der festgelegten Datenübertragungsrate) (Standardwert: 2000 ms, wenn die Blockprüfung aktiviert ist; 550 ms, wenn die Blockprüfung nicht aktiviert ist) 1 ms bis 65535 ms

Hinweis

Mit Ausnahme der Priorität müssen die Protokolleinstellungen für das CM-Modul und den Kommunikationspartner die gleichen sein.

13.3.5 Punkt-zu-Punkt-Anweisungen

13.3.5.1 Gemeinsame Parameter für Punkt-zu-Punkt-Operationen

Tabelle 13-3 Allgemeine Eingangsparameter der PTP-Anweisungen

Parameter	Beschreibung
REQ	Viele der PtP-Anweisungen haben einen Eingang REQ, der die Anweisung bei einer steigenden Flanke (0 nach 1) initiiert. Der Eingang REQ muss während der Ausführung einer Anweisung 1 (WAHR) sein, doch der Eingang REQ kann so lange wie gewünscht WAHR bleiben. Die Anweisung stößt erst dann eine andere Anweisung an, wenn sie mit Eingang REQ = FALSCH aufgerufen wird, so dass die Anweisung den Zustand des Eingangs REQ zurücksetzen kann. Dies ist erforderlich, damit die Anweisung die steigende Flanke zum Starten der nächsten Anweisung erkennen kann. Wenn Sie eine PtP-Anweisung in Ihr Programm einfügen, werden Sie von STEP 7 aufgefordert, den Instanz-DB anzugeben. Verwenden Sie für jeden PtP-Anweisungsaufwurf einen eindeutigen DB. Dadurch wird sichergestellt, dass jede Anweisung die Eingänge wie REQ ordnungsgemäß verarbeitet.
PORT	Eine Portadresse wird während der Konfiguration des Kommunikationsgeräts zugewiesen. Nach der Konfiguration kann aus der Klappliste des Parameters ein symbolischer Name für den Standardport ausgewählt werden. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Konstanten" der PLC-Variablen-Tabelle zugewiesen.
Auflösung von Bitzeiten	Für verschiedene Parameter wird die Anzahl der Bitzeiten bei der konfigurierten Baudrate angegeben. Durch Angabe des Parameters in Bitzeiten ist der Parameter unabhängig von der Baudrate. Alle Parameter mit Einheiten von Bitzeiten können mit einer maximalen Anzahl von 65535 angegeben werden. Der maximale Zeitraum jedoch, der von einem CM oder CB gemessen werden kann, beträgt 8 Sekunden.

Die Ausgangsparameter DONE, NDR, ERROR und STATUS der PtP-Anweisungen zeigen den Ausführungsstatus der PtP-Funktionen an.

Tabelle 13-4 Ausgangsparameter DONE, NDR, ERROR und STATUS

Parameter	Datentyp	Standard	Beschreibung
DONE	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehler abgeschlossen wurde; andernfalls FALSCH.
NDR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die angeforderte Aktion fehlerfrei abgeschlossen und neue Daten empfangen wurden; andernfalls FALSCH.

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Parameter	Datentyp	Standard	Beschreibung
ERROR	Bool	FALSCH	Wird eine Ausführung lang auf WAHR gesetzt, um anzuzeigen, dass die letzte Anforderung mit Fehlern abgeschlossen wurde, der entsprechende Fehlercode befindet sich in STATUS; andernfalls FALSCH.
STATUS	Word	0	<p>Ergebniszustand:</p> <ul style="list-style-type: none"> • Wird das Bit DONE oder NDR gesetzt, wird STATUS auf 0 oder auf einen Informationscode gesetzt. • Wird das Bit ERROR gesetzt, wird STATUS auf einen Fehlercode gesetzt. • Wird keines der oben aufgeführten Bits gesetzt, kann die Anweisung Statusergebnisse zurückgeben, die den aktuellen Zustand der Funktion beschreiben. <p>Der Wert in STATUS wird während der Ausführung der Funktion gehalten.</p>

Hinweis

Beachten Sie, dass die Parameter DONE, NDR und ERROR nur eine Ausführung lang gesetzt sind. Ihre Programmlogik muss temporäre Ausgangszustandswerte in Datenpuffern speichern, damit Sie in nachfolgenden Programmzyklen Zustandsänderungen erkennen können.

Tabelle 13-5 Gemeinsame Bedingungs-codes

STATUS (W#16#....)	Beschreibung
0000	Kein Fehler
7000	Funktion nicht besetzt
7001	Funktion ist beim ersten Aufruf aktiv.
7002	Funktion ist bei Folgeaufrufen aktiv (Abfragen nach dem ersten Aufruf).
8x3A	Unzulässiger Pointer in Parameter x
8070	Gesamter interner Speicher ausgelastet, zu viele Anweisungen gleichzeitig in Bearbeitung
8080	Portnummer ist unzulässig.
8081	Zeitüberschreitung, Modulfehler oder anderer interner Fehler
8082	Parametrierung fehlgeschlagen, weil Parametrierung im Hintergrund läuft.
8083	<p>Pufferüberlauf:</p> <p>Das CM oder CB hat eine Empfangsmeldung mit einer Länge, die größer ist als vom Längenparameter zugelassen, zurückgegeben.</p>
8090	Interner Fehler: Falsche Meldungslänge, falsches Submodul oder unzulässige Meldung Wenden Sie sich an den Kunden-Support.
8091	Interner Fehler: Falsche Version in Parametrierungsmeldung Wenden Sie sich an den Kunden-Support.
8092	Interner Fehler: Falsche Datensatzlänge in Parametrierungsmeldung Wenden Sie sich an den Kunden-Support.

Tabelle 13-6 Gemeinsame Fehlerklassen

Beschreibung der Klasse	Fehlerklassen	Beschreibung
Portkonfiguration	16#81Ax	Für die Beschreibung häufiger Fehler in der Schnittstellenkonfiguration
Sendekonfiguration	16#81Bx	Für die Beschreibung häufiger Fehler in der Sendekonfiguration
Empfangskonfiguration	16#81Cx 16#82Cx	Für die Beschreibung häufiger Fehler in der Empfangskonfiguration
Sende-Laufzeit	16#81Dx	Für die Beschreibung häufiger Laufzeitfehler beim Senden
Empfangs-Laufzeit	16#81Ex	Für die Beschreibung häufiger Laufzeitfehler beim Empfangen
Signalverarbeitung	16#81Fx	Für die Beschreibung häufiger Fehler in Verbindung mit der Signalverarbeitung
Pointer-Fehler	16#8p01 bis 16#8p51	Für ANY-Pointer-Fehler, wobei "p" die Parameternummer der Anweisung ist
Embedded-Protokoll-Fehler	16#848x 16#858x	Für Embedded-Protokoll-Fehler

13.3.5.2 Port_Config (Kommunikationsparameter dynamisch konfigurieren)

Tabelle 13-7 Anweisung Port_Config (Portkonfiguration)

KOP/FUP	SCL	Beschreibung
<pre> *Port_Config_DB* Port_Config - EN ENO - - REQ DONE - - PORT ERROR - - PROTOCOL STATUS - - BAUD - PARITY - DATABITS - STOPBITS - FLOWCTRL - XONCHAR - XOFFCHAR - WAITTIME - MODE - LINE_PRE - BRK_DET </pre>	<pre> "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_); </pre>	<p>Mit Port_Config können Sie Portparameter wie die Baudrate über Ihr Programm ändern.</p> <p>Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit Port_Config in Ihrem Programm die Konfiguration ändern.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Die CPU speichert die von Ihnen in der Anweisung Port_Config festgelegten Werte nicht dauerhaft. Die in der Gerätekonfiguration gespeicherten Parameter werden von der CPU wiederhergestellt, wenn diese von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere

Informationen finden Sie unter Kommunikationsports konfigurieren (Seite 947) und Flusskontrolle verwalten (Seite 949).

Tabelle 13-8 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke dieses Eingangs. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
PROTOCOL	IN	UInt	0 - Frei programmierbare Kommunikation (Standardwert) 1. Protokoll 3964(R)
BAUD	IN	UInt	Baudrate des Ports (Standardwert: 6): 1 = 300 Baud, 2 = 600 Baud, 3 = 1200 Baud, 4 = 2400 Baud, 5 = 4800 Baud, 6 = 9600 Baud, 7 = 19200 Baud, 8 = 38400 Baud, 9 = 57600 Baud, 10 = 76800 Baud, 11 = 115200 Baud
PARITY	IN	UInt	Parität des Ports (Standardwert: 1): 1 = Keine Parität, 2 = Gerade Parität, 3 = Ungerade Parität, 4 = Mark-Parität, 5 = Space-Parität
DATABITS	IN	UInt	Bits pro Zeichen (Standardwert: 1): 1 = 8 Datenbits, 2 = 7 Datenbits
STOPBITS	IN	UInt	Stoppbits (Standardwert: 1): 1 = 1 Stoppbit, 2 = 2 Stoppbits
FLOWCTRL*	IN	UInt	Flusskontrolle (Standardwert: 1): 1 = Keine Flusskontrolle, 2 = XON/XOFF, 3 = Hardware-RTS immer EIN, 4 = Hardware-RTS geschaltet
XONCHAR ¹	IN	Char	Gibt das Zeichen an, das als XON-Zeichen dient. Dies ist typischerweise ein DC1-Zeichen (16#11). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 16#11)
XOFFCHAR ¹	IN	Char	Gibt das Zeichen an, das als XOFF-Zeichen dient. Dies ist typischerweise ein DC3-Zeichen (16#13). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 16#13)
WAITTIME ¹	IN	UInt	Gibt an, wie lange nach dem Empfang eines XON-Zeichens auf ein XOFF-Zeichen gewartet werden soll bzw. wie lange nach Aktivierung von RTS auf das CTS-Signal gewartet werden soll (0 bis 65535 ms). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 2000)
MODE ²	IN	UInt	Gibt die gewählte Betriebsart des Moduls an. <ul style="list-style-type: none"> • 0 = Vollduplex (RS232) • 1 = Vollduplex (RS-422), Vierdrahtbetrieb (Punkt-zu-Punkt), Sender immer aktiviert • 2 = Vollduplex (RS-422), Vierdrahtbetrieb (Multipoint-Master), Sender immer aktiviert • 3 = Vollduplex (RS-422), Vierdrahtbetrieb (Multipoint-Slave), Sender beim Senden aktiviert • 4 = Halbduplex (RS-485), Zweidrahtbetrieb

Parameter und Datentyp		Datentyp	Beschreibung
LINE_PRE	IN	UInt	Gibt den inaktiven Leitungszustand (Leerlauf) an. Für RS422- und RS485-Module wird der inaktive Leitungszustand durch Anlegen einer Vorspannung an die Signale R(A) und R(B) hergestellt. Die folgenden Einstellungen sind möglich: <ul style="list-style-type: none"> 0 = Keine Vorspannung (keine Voreinstellung) (Standard) 1 = Vorspannung $R(A) > R(B) \geq 0V$; nur RS-422 2 = Vorspannung $R(B) > R(A) \geq 0V$; RS-422 und RS-485
BRK_DET	IN	UInt	Aktiviert/deaktiviert Kommunikationsleitungsbruchererkennung. Bei aktivierter Leitungsbruchererkennung meldet das Modul einen Fehler, wenn das Kommunikationskabel nicht am Modul angeschlossen ist. Im RS422 Point-to-Point-Betrieb ist die Leitungsbruchererkennung nur möglich, wenn die Voreinstellung der Empfangsleitung mit einer Vorspannung verwendet wird, sodass $R(A) > R(B) \geq 0V$ ist. <ul style="list-style-type: none"> 0 = Keine Leitungsbruchererkennung (Standard) 1 = Leitungsbruchererkennung aktiviert
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

¹ Gilt nicht, wenn Protokoll = 1 (Protokoll 3964(R))

² Nur die Betriebsarten 0 und 1 sind gültig, wenn Protokoll = 1 (Protokoll 3964(R)). Dies ist abhängig davon, ob Ihr CM-Modul ein RS-232-Modul oder ein RS-422-Modul ist.

Tabelle 13-9 Bedingungscode

STATUS (W#16#....)	Beschreibung
81A0	Angegebenes Protokoll ist nicht vorhanden.
81A1	Angegebene Baudrate ist nicht vorhanden.
81A2	Angegebene Paritätsoption ist nicht vorhanden.
81A3	Angegebene Anzahl Datenbits ist nicht vorhanden.
81A4	Angegebene Anzahl Stopbits ist nicht vorhanden.
80A5	Angegebene Art der Flusskontrolle ist nicht vorhanden.
81A6	Wartezeit ist 0 und Flusskontrolle ist aktiviert
81A7	XON und XOFF sind unzulässige Werte (z. B. der gleiche Wert)
81A8	Fehler im Baustein-Header (zum Beispiel falscher Bausteintyp oder falsche Bausteinlänge)
81A9	Rekonfiguration abgelehnt, weil ein Konfigurierungsvorgang läuft
81AA	Ungültiger RS422/RS485-Betrieb
81AB	Ungültige Voreinstellung der Empfangsleitung für Leitungsbruchererkennung
81AC	Ungültige RS232-Pausenverarbeitung
8280	Negative Quittierung beim Lesen aus Modul
8281	Negative Quittierung beim Schreiben in Modul
8282	DP-Slave oder Modul nicht verfügbar

13.3.5.3 Send_Config (Parameter für die serielle Kommunikation dynamisch konfigurieren)

Tabelle 13-10 Anweisung Send_Config (Sendekonfiguration)

KOP/FUP	SCL	Beschreibung
<pre> "Send_Config_DB" Send_Config - EN ENO - - REQ DONE - - PORT ERROR - - RTSONLY STATUS - - RTSOFFDLY - RTSOFFDLY - BREAK - IDLELINE - USR_END - APP_END </pre>	<pre> "Send_Config_DB" (REQ:=_bool_in_, PORT:=_word_in_, RTSONLY:=_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_); </pre>	<p>Send_Config ermöglicht das dynamische Konfigurieren serieller Übertragungsparameter für eine PtP-Kommunikationsschnittstelle. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn Send_Config ausgeführt wird.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit Send_Config in Ihrem Programm die Konfiguration ändern.

Die CPU speichert die von Ihnen in der Anweisung Send_Config festgelegten Werte nicht dauerhaft. Die in der Gerätekonfiguration gespeicherten Parameter werden von der CPU wiederhergestellt, wenn diese von RUN in STOP wechselt oder neu eingeschaltet wird. Siehe Sendeparameter konfigurieren (Seite 950).

Tabelle 13-11 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN PORT	Nachdem Sie ein CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
RTSONDLY	IN UInt	Anzahl der Millisekunden, die nach Aktivierung von RTS gewartet werden soll, bevor eine Übertragung von Tx-Daten durchgeführt wird. Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich liegt zwischen 0 und 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
RTSOFFDLY	IN UInt	Anzahl der Millisekunden, die nach der Übertragung von Tx-Daten gewartet werden soll, bevor RTS deaktiviert wird: Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich liegt zwischen 0 und 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
BREAK	IN UInt	Dieser Parameter gibt an, dass beim Start jeder Meldung für die angegebene Anzahl von Bitzeiten eine Pause gesendet wird. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 12)

Parameter und Datentyp		Datentyp	Beschreibung
IDLELINE	IN	UInt	Dieser Parameter gibt an, dass vor dem Start jeder Meldung für die angegebene Anzahl von Bitzeiten die Leitung im Leerlauf bleibt. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
USR_END*	IN	STRING[2]	Gibt Anzahl und Art der Zeichen im Abschlusstrennzeichen an. Das Abschlusstrennzeichen ist in den Sendepuffer eingebettet (nur Zeichen) und kennzeichnet das Ende der gesendeten Meldung (Zeichen werden bis zum Abschlusstrennzeichen übertragen). Das Abschlusstrennzeichen wird am Ende der Meldung angehängt. <ul style="list-style-type: none"> STRING[2,0,xx,yy] – Abschlusstrennzeichen nicht benutzt (Standard) STRING[2,1,xx,yy] – Abschlusstrennzeichen besteht aus einem Zeichen STRING[2,2,xx,yy] – Abschlusstrennzeichen besteht aus zwei Zeichen Entweder USR_END oder APP_END müssen eine Länge von null haben.
APP_END*	IN	STRING[5]	Gibt Anzahl und Art der Zeichen an, die an die gesendete Meldung anzuhängen sind (nur die Zeichen werden angehängt). STRING[5,0,aa,bb,cc,dd,ee] – Endezeichen nicht benutzt (Standard) <ul style="list-style-type: none"> STRING[5,1,aa,bb,cc,dd,ee] – Ein Endezeichen senden STRING[5,2,aa,bb,cc,dd,ee] – Zwei Endezeichen senden STRING[5,3,aa,bb,cc,dd,ee] – Drei Endezeichen senden STRING[5,4,aa,bb,cc,dd,ee] – Vier Endezeichen senden STRING[5,5,aa,bb,cc,dd,ee] – Fünf Endezeichen senden
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

* Für CM und CB 1241 nicht unterstützt; für den Parameter ist eine leere Zeichenkette ("") zu verwenden.

Tabelle 13-12 Bedingungscode

STATUS (W#16#....)	Beschreibung
81B0	Konfiguration eines Sendealarms ist nicht zulässig. Wenden Sie sich an den Customer Support.
81B1	Pausenzeit ist größer als der maximal zulässige Wert.
81B2	Leerlaufzeit ist größer als der maximal zulässige Wert.
81B3	Fehler im Baustein-Header, zum Beispiel falscher Bausteintyp oder falsche Bausteinlänge
81B4	Rekonfiguration abgelehnt, weil ein Konfigurierungsvorgang läuft
81B5	Angegebene Anzahl der Abschlusstrennzeichen ist größer als zwei oder Anzahl der Endezeichen ist größer als fünf
81B6	Sendekonfiguration abgelehnt, wenn in der Firmware eingebettete Protokolle betroffen sind
8280	Negative Quittierung beim Lesen aus Modul
8281	Negative Quittierung beim Schreiben in Modul
8282	DP-Slave oder Modul nicht verfügbar

13.3.5.4 Receive_Config (Parameter für den seriellen Empfang dynamisch konfigurieren)

Tabelle 13-13 Anweisung Receive_Config (Empfangskonfiguration)

KOP/FUP	SCL	Beschreibung
<pre> Receive_Config - EN ENO - - REQ DONE - - PORT ERROR - - Receive_ STATUS - - Conditions </pre>	<pre> "Receive_Config_DB" (REQ:=_bool_in_, PORT:=_uint_in_, Receive_Conditions:=_struct _in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Receive_Config führt die dynamische Konfiguration serieller Empfangsparameter für eine PtP-Kommunikationsschnittstelle durch. Diese Anweisung konfiguriert die Bedingungen, die den Beginn und das Ende einer empfangenen Meldung kennzeichnen. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn Receive_Config ausgeführt wird.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Kommunikationsanschlusses in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit Receive_Config in Ihrem Programm die Konfiguration ändern.

Die CPU speichert die von Ihnen in der Anweisung Receive_Config festgelegten Werte nicht dauerhaft. Die in der Gerätekonfiguration gespeicherten Parameter werden von der CPU wiederhergestellt, wenn diese von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere Informationen dazu finden Sie unter "Empfangsparameter konfigurieren (Seite 952)".

Tabelle 13-14 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
CONDITIONS	IN	CONDITIONS	Die Datenstruktur von CONDITIONS gibt die Anfangs- und Endbedingungen der Meldung wie im Folgenden beschrieben an.
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Startbedingungen für die Anweisung Receive_P2P

Die Anweisung Receive_P2P verwendet die von der Anweisung Receive_Config angegebene Konfiguration, um Anfang und Ende von Punkt-zu-Punkt-Kommunikationsmeldungen zu bestimmen. Der Meldungsbeginn wird von den Startbedingungen festgelegt. Der Meldungsbeginn kann anhand von einer oder mehreren Startbedingungen ermittelt werden. Sind mehrere Startbedingungen angegeben, müssen alle Bedingungen erfüllt sein, damit die Meldung gestartet wird.

Eine Beschreibung der Bedingungen für den Meldungsbeginn finden Sie unter "Empfangsparameter konfigurieren (Seite 952)".

Datentypstruktur des Parameters CONDITIONS, Teil 1 (Startbedingungen)

Tabelle 13-15 Struktur von CONDITIONS für Startbedingungen

Parameter und Datentyp	Datentyp	Beschreibung
STARTCOND	IN	UInt Angabe der Startbedingung (Standardwert: 1) <ul style="list-style-type: none"> • 01H - Startzeichen • 02H - Beliebige Zeichen • 04H - Leitungspause • 08H - Leitung im Leerlauf • 10H - Zeichenfolge 1 • 20H - Zeichenfolge 2 • 40H - Zeichenfolge 3 • 80H - Zeichenfolge 4
IDLETIME	IN	UInt Die Anzahl der erforderlichen Bitzeiten für Timeout des Leitungsleerlaufs. (Standardwert: 40). Nur in Verbindung mit der Bedingung "Leitung im Leerlauf". 0 bis 65535
STARTCHAR	IN	Byte Das Startzeichen für die Bedingung "Startzeichen". (Standardwert: B#16#2)
STRSEQ1CTL	IN	Byte Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/vergleichen: (Standardwert: B#16#0) Dies sind die Aktivierungsbits für jedes Zeichen der Startzeichenfolge. <ul style="list-style-type: none"> • 01H - Zeichen 1 • 02H - Zeichen 2 • 04H - Zeichen 3 • 08H - Zeichen 4 • 10H - Zeichen 5 Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zeichenfolge jedes Zeichen eine Übereinstimmung darstellt.
STRSEQ1	IN	Char[5] Zeichenfolge 1, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ2CTL	IN	Byte Zeichenfolge 2, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0)
STRSEQ2	IN	Char[5] Zeichenfolge 2, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ3CTL	IN	Byte Zeichenfolge 3, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0

Parameter und Datentyp		Datentyp	Beschreibung
STRSEQ3	IN	Char[5]	Zeichenfolge 3, Startzeichen (5 Zeichen). Standardwert: 0
STRSEQ4CTL	IN	Byte	Zeichenfolge 4, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0
STRSEQ4	IN	Char[5]	Zeichenfolge 4, Startzeichen (5 Zeichen), Standardwert: 0

Beispiel

Sehen Sie sich die folgende empfangene hexadezimal-codierte Meldung an: "**68** 10 aa **68** bb 10 aa 16". Die konfigurierten Startzeichenfolgen finden Sie in der folgenden Tabelle. Startzeichenfolgen werden ausgewertet, nachdem das erste Zeichen 68H erfolgreich empfangen wurde. Nach erfolgreichem Empfang des vierten Zeichens (das zweite 68H) ist die Startbedingung 1 erfüllt. Wenn die Startbedingungen erfüllt sind, beginnt die Auswertung der Endbedingungen.

Die Verarbeitung der Startzeichenfolge kann aufgrund verschiedener Fehler bei Parität, Framing oder Zeitabständen zwischen den Zeichen beendet werden. Diese Fehler führen dazu, dass die Meldung nicht empfangen wird, weil die Startbedingung nicht erfüllt wurde.

Tabelle 13-16 Startbedingungen

Startbedingung	Erstes Zeichen	Erstes Zeichen +1	Erstes Zeichen +2	Erstes Zeichen +3	Erstes Zeichen +4
1	68H	xx	xx	68H	xx
2	10H	aaH	xx	xx	xx
3	dcH	aaH	xx	xx	xx
4	e5H	xx	xx	xx	xx

Endbedingungen für die Anweisung Receive_P2P

Das Ende einer Meldung wird von den angegebenen Endbedingungen festgelegt. Das Ende einer Meldung wird durch das erste Auftreten einer oder mehrerer konfigurierter Endbedingungen festgelegt. Im Abschnitt "Bedingungen für den Meldungsbeginn" unter "Empfangsparameter konfigurieren (Seite 952)" werden die Endbedingungen beschrieben, die Sie für die Anweisung Receive_Config konfigurieren können.

Sie können die Endbedingungen entweder in den Eigenschaften der Kommunikationsschnittstelle in der Gerätekonfiguration oder über die Anweisung Receive_Config konfigurieren. Immer wenn die CPU von STOP in RUN wechselt, werden die Empfangsparameter (Start- und Endbedingungen) wieder auf die Einstellungen der Gerätekonfiguration gesetzt. Wenn das STEP 7 Anwenderprogramm Receive_Config ausführt, werden die Einstellungen auf die Bedingungen von Receive_Config gesetzt.

Datentypstruktur des Parameters CONDITIONS, Teil 2 (Endebedingungen)

Tabelle 13-17 Struktur von CONDITIONS für Endebedingungen

Parameter	Parametertyp	Datentyp	Beschreibung
ENDCOND	IN	UInt 0	Dieser Parameter gibt die Bedingung für das Meldungsende an: <ul style="list-style-type: none"> • 01H - Antwortzeit • 02H - Meldungszeit • 04H - Zeichenabstand • 08H - Maximale Länge • 10H - N + LEN + M • 20H - Zeichenfolge
MAXLEN	IN	UInt 1	Maximale Meldungslänge: Wird nur verwendet, wenn die Endebedingung "Maximale Länge" ausgewählt ist. 1 bis 1024 Bytes
N	IN	UInt 0	Byteposition des Längenfilds in der Meldung. Wird nur bei der Endebedingung N + LEN + M verwendet. 1 bis 1022 Bytes
LENGTHSIZE	IN	UInt 0	Größe des Bytefelds (1, 2 oder 4 Byte). Wird nur bei der Endebedingung N + LEN + M verwendet.
LENGTHM	IN	UInt 0	Geben Sie die Anzahl der Zeichen nach dem Längenfild an, die nicht im Wert des Längenfilds enthalten sind. Diese Angabe wird nur bei der Endebedingung N + LEN + M verwendet. 0 bis 255 Bytes
RCVTIME	IN	UInt 200	Geben Sie an, wie lange auf das erste empfangene Zeichen gewartet werden soll. Die Empfangsanweisung wird mit einem Fehler beendet, wenn nicht innerhalb der angegebenen Zeit ein Zeichen erfolgreich empfangen wird. Diese Angabe wird nur bei der Bedingung "Antwortzeit" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden) Bei diesem Parameter handelt es sich nicht um eine Endebedingung, weil die Auswertung beendet wird, wenn das erste Zeichen einer Antwort empfangen wird. Es handelt sich nur in dem Sinn um eine Endebedingung, als dass eine Empfangsanweisung beendet wird, weil bei Erwartung einer Antwort keine Antwort empfangen wird. Sie müssen eine getrennte Endebedingung definieren.
MSGTIME	IN	UInt 200	Geben Sie an, wie lange nach dem Empfang des ersten Zeichens auf den vollständigen Empfang der gesamten Meldung gewartet werden soll. Dieser Parameter wird nur verwendet, wenn die Bedingung "Timeout der Meldung" ausgewählt ist. (0 bis 65535 ms)
CHARGAP	IN	UInt 12	Geben Sie die Anzahl der Bitzeiten zwischen den Zeichen an. Wenn die Anzahl der Bitzeiten zwischen den Zeichen den angegebenen Wert überschreitet, ist die Endebedingung erfüllt. Diese Angabe wird nur bei der Bedingung "Zeichenabstand" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden)

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Parameter	Parametertyp	Datentyp	Beschreibung
ENDSEQ1CTL	IN	Byte B#16#0	Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/ vergleichen: Dies sind die Aktivierungsbits für jedes Zeichen der Ende- zeichenfolge. Zeichen 1 ist Bit 0, Zeichen 2 ist Bit 1, ..., Zeichen 5 ist Bit 4. Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zei- chenfolge jedes Zeichen eine Übereinstimmung darstellt.
ENDSEQ1	IN	Char[5] 0	Zeichenfolge 1, Startzeichen (5 Zeichen)

Tabelle 13-18 Bedingungs-codes

STATUS (W#16#....)	Beschreibung
81C0	Unzulässige Startbedingung ausgewählt
81C1	Unzulässige Endebedingung ausgewählt, keine Endebedingung ausgewählt
81C2	Empfangsalarm aktiviert und dies ist nicht möglich.
81C3	Endebedingung "Maximale Länge" ist aktiviert und die maximale Länge ist 0 oder > 1024.
81C4	Berechnete Länge ist aktiviert und N ist >= 1023.
81C5	Berechnete Länge ist aktiviert und Länge ist nicht 1, 2 oder 4.
81C6	Berechnete Länge ist aktiviert und der Wert von M ist > 255.
81C7	Berechnete Länge ist aktiviert und die berechnete Länge ist > 1024.
81C8	Timeout der Antwort ist aktiviert und das Antwort-Timeout ist null.
81C9	Timeout für den Zeichenabstand ist aktiviert und das Timeout ist null.
81CA	Timeout für den Leitungsleerlauf ist aktiviert und das Timeout ist null.
81CB	Endezeichenfolge ist aktiviert, doch alle Zeichen sind "nicht relevant".
81CC	Startzeichenfolge (eine von 4) ist aktiviert, doch alle Zeichen sind "nicht relevant".
81CD	Fehler Überschreibschutz, Empfangsmeldung ungültig
81CE	Fehler Pufferverarbeitung für Empfangsmeldungen beim Übergang von STOP zu RUN ungültig
81CF	Fehler im Baustein-Header, zum Beispiel falscher Bausteintyp oder falsche Bausteinlänge
8281	Negative Quittierung beim Schreiben in Modul
8282	DP-Slave oder Modul nicht verfügbar
82C0	Rekonfiguration abgelehnt, weil ein Konfigurierungsvorgang läuft
82C1	Der angegebene Wert für die Anzahl der Meldungen, die das Modul puffern kann, ist größer als der maximal zulässige Wert.
82C2	Empfangskonfiguration abgelehnt, wenn in der Firmware eingebettete Protokolle betroffen sind
8351	Datentyp für diesen Variant-Pointer nicht zulässig

13.3.5.5 P3964_Config (3964(R)-Protokoll konfigurieren)

Tabelle 13-19 Anweisung P3964_Config (3964_Config (3964(R)-Protokoll konfigurieren)

KOP/FUP	SCL	Beschreibung
<pre> *P3964_Config_ DB* P3964_Config - EN ENO - REQ DONE - PORT ERROR - BCC STATUS - Priority - CharacterDelay - Time - AcknDelayTime - BuildupAttempt - s - RepetitionAtte - mpts </pre>	<pre> "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_); </pre>	<p>Mit P3964_Config können Sie während der Laufzeit Priorität und Protokollparameter ändern.</p> <p>Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Mit der Anweisung P3964_Config können Sie in Ihrem Programm die Konfiguration ändern.</p>

¹ STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.

Die CPU speichert die von Ihnen in der Anweisung P3964_Config festgelegten Werte nicht dauerhaft. Die in der Gerätekonfiguration gespeicherten Parameter werden von der CPU wiederhergestellt, wenn diese neu eingeschaltet wird. Weitere Informationen finden Sie unter Konfigurieren der Kommunikationspriorität und der Protokollparameter für 3964(R) (Seite 961).

Tabelle 13-20 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
PORT	IN	UInt
BCC	IN	USInt
Priority	IN	UInt
CharacterDelay-Time	IN	UInt
AcknDelayTime	IN	UInt

Beschreibung der Parameter:

- REQ:** Aktiviert die Konfigurationsänderung bei steigender Flanke dieses Eingangs. (Standardwert: Falsch)
- PORT:** Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
- BCC:** Aktiviert/deaktiviert die Verwendung der Blockprüfung
 - 0 = ohne Blockprüfung
 - 1 = mit Blockprüfung
- Priority:** Auswahl der Priorität
 - 0 = niedrige Priorität
 - 1 = hohe Priorität
 Die Priorität des CMs muss der Priorität des Kommunikationspartners entgegengesetzt sein.
- CharacterDelay-Time:** Einstellung der Zeichenverzugszeit (abhängig von der festgelegten Datenübertragungsrate) (Standardwert: 220 ms)
1 ms bis 65535 ms
- AcknDelayTime:** Einstellung der Quittierungsverzugszeit (abhängig von der festgelegten Datenübertragungsrate) (Standardwert: 2000 ms)
1 ms bis 65535 ms

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Parameter und Datentyp	Datentyp	Beschreibung	
BuildupAttempts	IN	UInt	Anzahl der Verbindungsversuche (Standardwert: 6 Verbindungsversuche) 1 bis 255
RepetitionAttempts	IN	UInt	Anzahl der Übertragungsversuche (Standardwert: 6 Verbindungsversuche) 1 bis 255
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 13-21 Bedingungscode

STATUS (W#16#....)	Beschreibung
16#8380	Parametrierfehler: Ungültiger Wert für "Zeichenverzugszeit".
16#8381	Parametrierfehler: Ungültiger Wert für "Antwort-Timeout".
16#8382	Parametrierfehler: Ungültiger Wert für "Priorität".
16#8383	Parametrierfehler: Ungültiger Wert für "Blockprüfung".
16#8384	Parametrierfehler: Ungültiger Wert für "Verbindungsversuche".
16#8385	Parametrierfehler: Ungültiger Wert für "Übertragungsversuche".
16#8386	Laufzeitfehler: Anzahl der Verbindungsversuche überschritten
16#8387	Laufzeitfehler: Anzahl der Übertragungsversuche überschritten
16#8388	Laufzeitfehler: Fehler am "Blockprüfzeichen" Der intern berechnete Wert des Blockprüfzeichens stimmt nicht mit dem am Verbindungsende vom Partner empfangenen Blockprüfzeichen überein.
16#8389	Laufzeitfehler: Ungültiges Zeichen empfangen beim Warten auf freien Empfangspuffer.
16#838A	Laufzeitfehler: Logischer Fehler beim Empfangen. Nach dem Empfang von DLE wurde ein weiteres zufälliges Zeichen (nicht DLE oder ETX) empfangen.
16#838B	Laufzeitfehler: Zeichenverzugszeit überschritten
16#838C	Laufzeitfehler: Wartezeit für freien Empfangspuffer hat begonnen
16#838D	Laufzeitfehler: Telegrammwiederholung beginnt nicht innerhalb von 4 s nach NAK
16#838E	Laufzeitfehler: Im Leerlauf wurde mindestens ein Zeichen empfangen (nicht NAK oder STX).
16#838F	Laufzeitfehler: Initialisierungskonflikt - Beide Partner haben hohe Priorität eingestellt
16#8391	Parametrierfehler: 3964-Konfigurationsdaten zurückgewiesen, weil die frei programmierbare Kommunikation eingestellt ist

13.3.5.6 Send_P2P (Sendepufferdaten übertragen)

Tabelle 13-22 Anweisung Send_P2P (Punkt-zu-Punkt-Daten senden)

KOP/FUP	SCL	Beschreibung
<pre> "Send_P2P_DB" Send_P2P - EN ENO - - REQ DONE - - PORT ERROR - - BUFFER STATUS - - LENGTH </pre>	<pre> "Send_P2P_DB" (REQ:=_bool_in_, PORT:=_word_in_, BUFFER:=_variant_in_, LENGTH:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Send_P2P startet die Übertragung der Daten und überträgt den zugewiesenen Puffer zur Kommunikationsschnittstelle. Das Programm der CPU wird weiterhin ausgeführt, während das CM oder CB die Daten mit der zugewiesenen Baudrate sendet. Es darf zu jeder Zeit nur eine Sendeanweisung anstehen. Das CM oder CB gibt einen Fehler aus, wenn eine zweite Anweisung Send_P2P ausgeführt wird, während das CM oder CB bereits eine Meldung sendet.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-23 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Freigabe der angeforderten Übertragung bei steigender Flanke dieses Freigabeeingangs. Dadurch wird der Inhalt des Puffers zur Punkt-zu-Punkt-Kommunikationsschnittstelle übertragen. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Sendepuffers. (Standardwert: 0) Hinweis: Boolesche Daten oder Boolesche Felder werden nicht unterstützt.
LENGTH	IN	UInt	Übertragene Rahmenlänge in Byte (Standardwert: 0) Verwenden Sie beim Senden einer komplexen Struktur immer die Länge 0. Wenn die Länge 0 ist, überträgt die Anweisung den gesamten Rahmen.
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

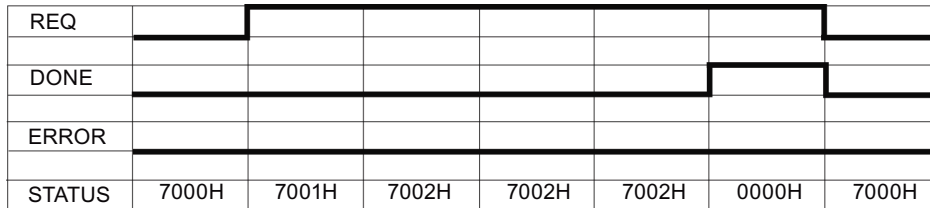
Wenn eine Sendeanweisung in Bearbeitung ist, sind die Ausgänge DONE und ERROR im Zustand FALSCH. Nach dem Ende einer Sendeanweisung wird einer der Ausgänge DONE oder ERROR auf WAHR gesetzt, um den Zustand der Sendeanweisung zu melden. Während DONE oder ERROR im Zustand WAHR ist, ist der Ausgang STATUS gültig.

Die Anweisung gibt den Status 16#7001 aus, wenn die Kommunikationsschnittstelle die Sendedaten annimmt. Nachfolgende Ausführungen von Send_P2P geben den Wert 16#7002

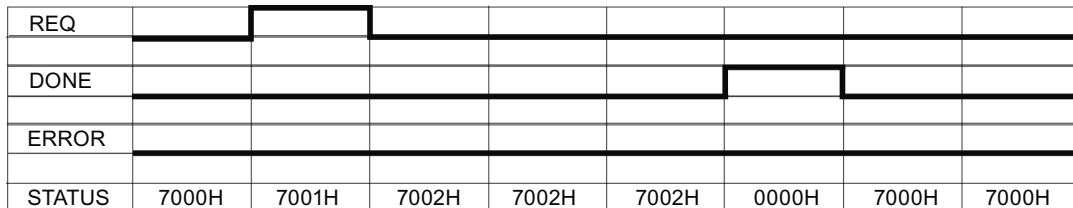
13.3 Punkt-zu-Punkt-Kommunikation (PtP)

aus, wenn das CM oder CB immer noch sendet. Nach dem Ende der Sendeanweisung gibt das CM oder CB den Status 16#0000 für die Sendeanweisung aus (sofern kein Fehler aufgetreten ist). Nachfolgende Ausführungen von Send_P2P mit REQ = 0 geben den Status 16#7000 (frei) aus.

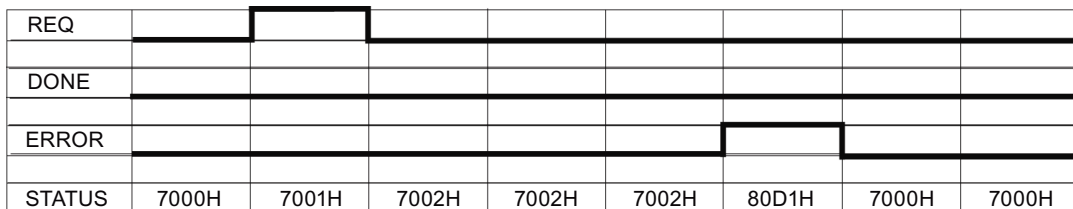
Das folgende Diagramm zeigt die Beziehung zwischen den Ausgangswerten und REQ. Hierbei wird vorausgesetzt, dass die Anweisung regelmäßig aufgerufen wird, um den Status des Sendevorgangs zu prüfen. In der folgenden Abbildung wird davon ausgegangen, dass die Anweisung in jedem Zyklus aufgerufen wird (dargestellt durch die STATUS-Werte).



Die folgende Abbildung zeigt, wie die Parameter DONE und STATUS nur einen Zyklus lang gültig sind, wenn an der REQ-Leitung (einen Zyklus lang) ein Impuls anliegt, um die Sendeanweisung anzustoßen.



Die folgende Abbildung zeigt die Beziehung der Parameter DONE, ERROR und STATUS im Fehlerfall.



Die Werte DONE, ERROR und STATUS sind nur solange gültig, bis Send_P2P erneut mit dem gleichen Instanz-DB ausgeführt wird.

Tabelle 13-24 Bedingungs-codes

STATUS (W#16#...)	Beschreibung
81D0	Neue Anforderung bei aktivem Sender
81D1	Sendung abgebrochen, weil kein CTS innerhalb der Wartezeit
81D2	Sendung abgebrochen, weil kein DSR vom DCE-Gerät
81D3	Sendung wegen Überlauf der Warteschlange abgebrochen (mehr als 1024 Bytes senden)
81D5	Umgekehrtes Biassignal (Drahtbruch)
81D6	Sendeanforderung abgelehnt, weil im Sendepuffer kein Abschlusstrennzeichen gefunden wurde.

STATUS (W#16#....)	Beschreibung
81D7	Interner Fehler / Synchronisationsfehler zwischen FB und CM
81D8	Sendeversuch abgelehnt, weil der Anschluss nicht konfiguriert ist.
81DF	CM hat Schnittstelle zum FB aus einem der folgenden Gründe zurückgesetzt <ul style="list-style-type: none"> • Modul neu gestartet (Aus-/Einschalten) • CPU hat einen Haltepunkt erreicht • Modul wurde neu parametrier Das Modul zeigt den entsprechenden Code in jedem Fall im Status-Parameter an. Das Modul setzt Status und Error nach dem ersten empfangenen Datensatz für SEND_P2P zurück.
8281	Negative Quittierung beim Schreiben in Modul
8282	DP-Slave oder Modul nicht verfügbar
8301	Unzulässige Syntax-ID an einem ANY-Pointer
8322	Bereichslängenfehler beim Lesen eines Parameters
8324	Bereichsfehler beim Lesen eines Parameters
8328	Ausrichtungsfehler beim Lesen eines Parameters
8332	Der Parameter enthält eine DB-Nummer, die höher als die maximal zulässige Nummer ist (DB-Nummernfehler).
833A	Der DB für den Parameter BUFFER ist nicht vorhanden.

Hinweis

Einstellen der maximalen Länge des Datensatzes für die Profibus-Kommunikation

Wenn das Kommunikationsmodul CM1243-5 als Profibus-Master zur Steuerung eines ET200SP- oder ET200MP-Profibus-Geräts dient, das wiederum ein RS232-, RS422- oder RS485-Punkt-zu-Punkt-Modul verwendet, müssen Sie die Datenbausteinvariable "max_record_len" wie folgt explizit auf den Wert 240 setzen:

Setzen Sie nach Ausführung einer Kommunikationsanweisung wie z. B. Port_Config, Send_Config oder Receive_Config den Parameter max_record_len im Instanz-DB (beispielsweise "Send_P2P_DB".max_record_len) auf 240.

Die explizite Zuweisung des Werts für max_record_len ist nur bei Profibus-Verbindungen erforderlich. Bei Profinet-Verbindungen wird bereits ein gültiger Wert für max_record_len verwendet.

Interaktion der Parameter LENGTH und BUFFER

Die Mindestdatengröße, die von der Anweisung SEND_P2P gesendet werden kann, ist ein Byte. Der Parameter BUFFER legt die Größe der zu sendenden Daten fest. Sie können für den Parameter BUFFER weder den Datentyp Bool noch Arrays vom Typ Bool verwenden.

Sie können den Parameter LENGTH jederzeit auf 0 setzen und damit sicherstellen, dass SEND_P2P die gesamte Datenstruktur des Parameters BUFFER sendet. Wenn Sie nur einen Teil

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

der Datenstruktur des Parameters BUFFER senden möchten, können Sie LENGTH wie folgt setzen:

Tabelle 13-25 Parameter LENGTH und BUFFER

LENGTH	BUFFER	Beschreibung
= 0	Nicht verwendet	Die vollständigen Daten werden wie vom Parameter BUFFER definiert gesendet. Wenn LENGTH = 0, brauchen Sie die Anzahl der übertragenen Bytes nicht anzugeben.
> 0	Elementarer Datentyp	Der LENGTH-Wert muss die Bytezahl dieses Datentyps enthalten. Beispiel: Bei einem Word-Wert muss LENGTH zwei sein. Bei einem Dword- oder Real-Wert muss LENGTH vier sein. Andernfalls erfolgt keine Übertragung, und der Fehler 8088H wird ausgegeben.
	Struktur	Der LENGTH-Wert kann eine Bytezahl enthalten, die kleiner als die vollständige Bytelänge der Struktur ist. In diesem Fall werden von der Anweisung nur die ersten n Bytes der Struktur aus BUFFER, gesendet, wobei n = LENGTH ist. Weil die interne Byteanordnung einer Struktur nicht immer bestimmt werden kann, kann dies zu unerwarteten Ergebnissen führen. Verwenden Sie dann eine LENGTH = 0, um die vollständige Struktur zu senden.
	Array	Der LENGTH-Wert muss eine Bytezahl kleiner als oder gleich der vollständigen Bytelänge des Arrays enthalten, wobei es sich um ein Vielfaches der Bytezahl des Datenelements handeln muss. Beispiel: Der Parameter LENGTH eines Arrays vom Typ Word muss ein Vielfaches von zwei sein und bei einem Array vom Typ Real ein Vielfaches von vier. Wenn LENGTH angegeben ist, überträgt die Anweisung die Anzahl der Arrayelemente, die dem LENGTH-Wert in Bytes entspricht. Wenn beispielsweise BUFFER ein Array mit 15 Dword-Elementen (insgesamt 60 Bytes) enthält und Sie geben LENGTH = 20 an, werden die ersten fünf Dword-Elemente aus dem Array übertragen. Der LENGTH-Wert muss ein Vielfaches der Bytezahl der Datenelemente sein. Andernfalls ist STATUS = 8088H, ERROR = 1 und keine Übertragung erfolgt.
	String	Der Parameter LENGTH enthält die Anzahl der zu sendenden Zeichen. Nur die Zeichen des String werden übertragen. Die Bytes mit der maximalen und der tatsächlichen Länge des String werden nicht gesendet.

13.3.5.7 Receive_P2P (Meldungsempfang aktivieren)

Tabelle 13-26 Anweisung Receive_P2P (Punkt-zu-Punkt-Daten empfangen)

KOP/FUP	SCL	Beschreibung
	<pre>"Receive_P2P_DB" (PORT:=_word_in_, BUFFER:=_variant_in_, NDR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, LENGTH=>_uint_out_);</pre>	<p>Receive_P2P prüft die Meldungen, die im CM oder CB empfangen wurden. Wenn eine Meldung verfügbar ist, wird sie vom CM oder CB zur CPU übertragen. Ein Fehler gibt den entsprechenden STATUS-Wert aus.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-27 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Empfangspuffers. Dieser Puffer muss groß genug sein, um die maximale Meldungslänge zu empfangen. Boolesche Daten oder Boolesche Felder werden nicht unterstützt. (Standardwert: 0)
NDR	OUT	Bool	Eine Ausführung lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde.
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die Anweisung mit Fehler beendet wurde.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
LENGTH	OUT	UInt	Länge der ausgegebenen Meldung in Byte (Standardwert: 0)

Der STATUS-Wert ist gültig, wenn NDR oder ERROR im Zustand WAHR ist. Der STATUS-Wert liefert den Grund für die Beendigung der Empfangsoperation im CM oder CB. Dies ist typischerweise ein positiver Wert, der angibt, dass die Empfangsoperation erfolgreich war und dass der Empfangsvorgang normal beendet wurde. Ist der STATUS-Wert negativ (das höchstwertige Bit des Hexadezimalwerts ist gesetzt), wurde die Empfangsoperation wegen einer Fehlerbedingung wie Paritäts-, Framing- oder Überlauffehler beendet.

Jede PtP-Kommunikationsschnittstelle kann maximal 1024 Bytes puffern. Hierbei kann es sich um eine große oder mehrere kleinere Meldungen handeln. Sind mehrere Meldungen im CM oder CB verfügbar, gibt die Anweisung Receive_P2P die älteste verfügbare Meldung aus. Eine anschließend ausgeführte Anweisung Receive_P2P gibt die zweitälteste Meldung aus.

Tabelle 13-28 Bedingungs-codes

STATUS (W#16#...)	Beschreibung
0000	Kein Puffer vorhanden
0094	Meldung wurde beendet, weil die maximale Zeichenlänge empfangen wurde
0095	Meldung wurde wegen Meldungs-Timeout beendet
0096	Meldung wurde wegen Zeichenabstands-Timeout beendet
0097	Meldung wurde wegen Antwort-Timeout beendet
0098	Meldung wurde beendet, weil die Längenbedingung "N+LEN+M" erfüllt war
0099	Meldung wurde beendet, weil die Endezeichenfolge erfüllt war
8085	Parameter LENGTH hat einen Wert 0 oder ist größer als 1 kB.
8088	Parameter LENGTH oder die empfangene Länge ist größer als der in BUFFER angegebene Bereich oder die empfangene Länge ist größer als der in BUFFER angegebene Bereich.
8090	Fehlerhafte Konfigurationsmeldung, falsche Meldungslänge, falsches Submodul, unzulässige Meldung
81E0	Meldung wurde beendet, weil der Empfangspuffer voll ist
81E1	Meldung wurde wegen Paritätsfehler beendet
81E2	Meldung wurde wegen Framingfehler beendet
81E3	Meldung wurde wegen Überlauffehler beendet
81E4	Meldung wurde beendet, weil die berechnete Länge die Puffergröße überschreitet

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

STATUS (W#16#...)	Beschreibung
81E5	Umgekehrtes Biassignal (Drahtbruch)
81E6	Meldungswarteschlange voll. Dieser Fehler wird ohne Daten gemeldet. Tritt er auf, schaltet das Modul zwischen einer fehlerfreien Datenübertragung und dem Fehler hin und her.
81E7	Interner Fehler, Synchronisationsfehler zwischen Anweisung und CM: wird bei einem Sequenzfehler gesetzt
81E8	Meldung beendet, Zeitüberschreitung zwischen Zeichen erreicht, bevor Meldungsende erkannt wurde
81E9	Modbus-CRC-Fehler erkannt (nur für Module, die die CRC-Generierung/Prüfung für das Modbus-Protokoll unterstützen)
81EA	Modbus-Telegramm zu kurz (nur für Module, die die CRC-Generierung/Prüfung für das Modbus-Protokoll unterstützen)
81EB	Meldung abgebrochen, da maximale Meldungsgröße überschritten
8201	Unzulässige Syntax-ID an einem ANY-Pointer
8223	Bereichslängenfehler beim Schreiben eines Parameters. Der Parameter liegt ganz oder teilweise außerhalb eines Adressbereichs oder die Länge eines Bitbereichs ist kein Vielfaches von 8 bei einem ANY-Pointer.
8225	Bereichsfehler beim Schreiben eines Parameters. Der Parameter liegt in einem Bereich, der für die Systemfunktion nicht zulässig ist.
8229	Ausrichtungsfehler beim Schreiben eines Parameters. Der referenzierte Parameter befindet sich an einer Bitadresse, die nicht gleich 0 ist.
8230	Der Parameter befindet sich in einem schreibgeschützten globalen DB
8231	Der Parameter befindet sich in einem schreibgeschützten Instanz-DB
8232	Der Parameter enthält eine DB-Nummer, die höher als die maximal zulässige Bausteinnummer ist (DB-Nummernfehler).
823A	Der DB für den Parameter BUFFER ist nicht vorhanden.
8280	Negative Quittierung beim Lesen aus Modul
8282	DP-Slave oder Modul nicht verfügbar

13.3.5.8 Receive_Reset (Empfangspuffer löschen)

Tabelle 13-29 Anweisung Receive_Reset (Empfänger zurücksetzen)

KOP/FUP	SCL	Beschreibung
<pre> *Receive_Reset_ DB Receive_Reset - EN ENO - - REQ DONE - - PORT ERROR - STATUS - </pre>	<pre> "Receive_Reset_DB" (REQ:=_bool_in_, PORT:=_word_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_); </pre>	<p>Receive_Reset löscht den Empfangspuffer im CM oder CB.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-30 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Aktiviert das Löschen des Empfangspuffers bei steigender Flanke dieses Freigabeingangs (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
DONE	OUT	Bool	Einen Zyklus lang WAHR bedeutet, dass die letzte Anforderung fehlerfrei ausgeführt wurde.
ERROR	OUT	Bool	WAHR bedeutet, dass die letzte Anforderung mit Fehlern ausgeführt wurde. Ist dieser Ausgang WAHR, enthält Ausgang STATUS die zugehörigen Fehlercodes.
STATUS	OUT	Word	Fehlercode (Standardwert: 0)

13.3.5.9 Signal_Get (RS-232-Signale abfragen)

Tabelle 13-31 Anweisung Signal_Get (RS232-Signale abrufen)

KOP/FUP	SCL	Beschreibung
<pre> *Signal_Get_DB* Signal_Get - EN ENO - REQ NDR - PORT ERROR - STATUS DTR DSR RTS CTS DCD RING </pre>	<pre> "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_); </pre>	<p>Signal_Get liest die aktuellen Zustände der RS232-Kommunikationssignale.</p> <p>Diese Funktion gilt nur beim RS232-CM.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-32 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	RS232-Signalzustände werden bei der steigenden Flanke an diesem Eingang abgerufen (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen.
NDR	OUT	Bool	Einen Zyklus lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die Anweisung mit Fehler beendet wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Parameter und Datentyp	Datentyp	Beschreibung	
DTR	OUT	Bool	Datenterminal bereit, Modul bereit (Ausgang). Standardwert: False
DSR	OUT	Bool	Datensatz bereit, Kommunikationsteilnehmer bereit (Eingang). Standardwert: False
RTS	OUT	Bool	Sendeanforderung, Modul sendebereit (Ausgang). Standardwert: False
CTS	OUT	Bool	Sendebereit, Kommunikationsteilnehmer kann Daten empfangen (Eingang). Standardwert: False
DCD	OUT	Bool	Datenträger erkannt, Signalpegel empfangen (immer False, nicht unterstützt)
RING	OUT	Bool	Rufanzeige, Meldung eines eingehenden Rufs (immer False, nicht unterstützt)

Tabelle 13-33 Bedingungs-codes

STATUS (W#16#....)	Beschreibung
81F0	CM oder CB ist RS485 und es sind keine Signale verfügbar
81F4	Fehler im Baustein-Header, zum Beispiel falscher Bausteintyp oder falsche Bausteinlänge
8280	Negative Quittierung beim Lesen aus Modul
8282	DP-Slave oder Modul nicht verfügbar

13.3.5.10 Signal_Set (RS-232-Signale festlegen)

Tabelle 13-34 Anweisung Signal_Set (RS232-Signale setzen)

KOP/FUP	SCL	Beschreibung
<pre> *Signal_Set_DB* Signal_Set - EN ENO - - REQ DONE - - PORT ERROR - - SIGNAL STATUS - - RTS - DTR - DSR </pre>	<pre> "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_); </pre>	<p>Signal_Set setzt die Zustände der RS232-Kommunikationssignale. Diese Funktion gilt nur beim RS232-CM.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-35 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Die Anweisung zum Setzen der RS232-Signale wird bei der steigenden Flanke an diesem Eingang gestartet (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen. (Standardwert: 0)

Parameter und Datentyp		Datentyp	Beschreibung
SIGNAL	IN	Byte	Gibt die festzusetzenden Signale an: (mehrere zulässig). Standardwert: 0 <ul style="list-style-type: none"> • 01H = RTS • 02H = DTR • 04H = DSR
RTS	IN	Bool	Sendeanforderung, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch), Standardwert: Falsch
DTR	IN	Bool	Datenterminal bereit, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch). Standardwert: Falsch
DSR	IN	Bool	Datensatz bereit (gilt nur für Schnittstellentyp DCE), nicht verwendet.
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 13-36 Bedingungscode

STATUS (W#16#....)	Beschreibung
81F0	CM oder CB ist RS485 und es können keine Signale gesetzt werden
81F1	Signale können wegen Hardwareflusskontrolle nicht gesetzt werden
81F2	DSR kann nicht gesetzt werden, weil das Modul ein DTE-Gerät ist
81F3	DTR kann nicht gesetzt werden, weil das Modul ein DCE-Gerät ist
81F4	Fehler im Baustein-Header, zum Beispiel falscher Bausteintyp oder falsche Bausteinlänge
8280	Negative Quittierung beim Lesen aus Modul
8281	Negative Quittierung beim Schreiben in Modul
8282	DP-Slave oder Modul nicht verfügbar

13.3.5.11 Get_Features

Tabelle 13-37 Anweisung Get_Features (Erweiterte Funktionen abrufen)

KOP/FUP	SCL	Beschreibung
<pre> *Get_Features_ DB* Get_Features - EN ENO - - REQ NDR - - PORT ERROR - STATUS - MODBUS_CRC - DIAG_ALARM - SUPPLY_VOLT - </pre>	<pre> "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_); </pre>	Die Anweisung Get_Features liest die erweiterten Funktionsmöglichkeiten eines Moduls.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

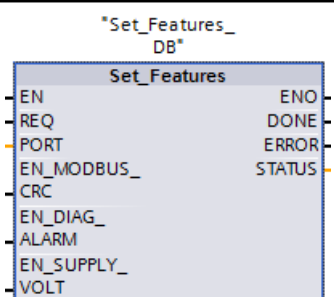
Tabelle 13-38 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
NDR	OUT	Bool	Zeigt an, dass neue Daten bereit sind.
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
MODBUS_CRC*	OUT	Bool	MODBUS CRC-Generierung und -Prüfung
DIAG_ALARM*	OUT	Bool	Erzeugung Diagnosealarm
SUPPLY_VOLT*	OUT	Bool	Diagnose für Ausfall Versorgungsspannung L+ verfügbar

*Get_Features gibt WAHR (1) zurück, wenn die Funktion verfügbar ist und FALSCH (0), wenn die Funktion nicht verfügbar ist

13.3.5.12 Set_Features

Tabelle 13-39 Anweisung Set_Features (Erweiterte Funktionen setzen)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Set_Features definiert die erweiterten Funktionen, die ein Modul unterstützt.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-40 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
EN_MODBUS_CRC	IN	Bool	MODBUS CRC-Generierung und -Prüfung aktivieren: <ul style="list-style-type: none"> • 0: CRC-Berechnung AUS (Standard) • 1: CRC-Berechnung EIN Hinweis: Dieser Parameter wird nur von CMs der Version 2.1, CPUs der Version 4.1 mit CBs und CM-PtP-Modulen für dezentrale E/A unterstützt.
EN_DIAG_ALARM	IN	Bool	Diagnosealarm aktivieren <ul style="list-style-type: none"> • 0: Diagnosealarm AUS • 1: Diagnosealarm EIN (Standard)
EN_SUPPLY_VOLT	IN	Bool	Diagnose für Ausfall Versorgungsspannung L+ aktivieren: <ul style="list-style-type: none"> • 0: Diagnose Versorgungsspannung deaktiviert (Standard) • 1: Diagnose Versorgungsspannung aktiviert
DONE	OUT	Bool	Gibt an, dass Set_Features ausgeführt ist
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

13.3.6 Programmieren der PtP-Kommunikation

STEP 7 bietet erweiterte Anweisungen, mit denen das Anwenderprogramm die Punkt-zu-Punkt-Kommunikation über ein im Anwenderprogramm konzipiertes und vorgegebenes Protokoll durchführen kann. Diese Anweisungen lassen sich in zwei Kategorien einteilen:

- Konfigurationsanweisungen
- Kommunikationsanweisungen

Konfigurationsanweisungen

Bevor das Anwenderprogramm die PtP-Kommunikation starten kann, müssen die Kommunikationsschnittstelle und die Parameter zum Senden und Empfangen der Daten konfiguriert werden.

Die Schnittstellenkonfiguration und die Meldungskonfiguration können für jedes CM oder CB in der Gerätekonfiguration oder mit den folgenden Anweisungen Ihres Anwenderprogramms durchgeführt werden:

- Port_Config (Seite 965)
- Send_Config (Seite 968)
- Receive_Config (Seite 970)

Kommunikationsanweisungen

Mit den Anweisungen für die Punkt-zu-Punkt-Kommunikation kann das Anwenderprogramm Meldungen an die Kommunikationsschnittstellen senden und von diesen Meldungen empfangen. Beachten Sie für weitere Informationen zum Übertragen von Daten mit diesen Anweisungen den Abschnitt zur Datenkonsistenz (Seite 187).

Alle PtP-Anweisungen funktionieren asynchron. Mit Hilfe einer Abfragearchitektur kann das Anwenderprogramm den Sende- und Empfangsstatus feststellen. Send_P2P und Receive_P2P können gleichzeitig ausgeführt werden. Die Kommunikationsmodule und das Kommunikationsboard puffern die Sende- und Empfangsmeldungen je nach Bedarf bis zu einer maximalen Puffergröße von 1024 Bytes.

Die CMs und das CB senden und empfangen Meldungen an die bzw. von den Kommunikationsteilnehmern. Das Meldungsprotokoll befindet sich in einem Puffer, der von einer bestimmten Kommunikationsschnittstelle empfangen oder an diese gesendet wird. Puffer und Port sind Parameter der Sende- und Empfangsanweisungen:

- Send_P2P (Seite 977)
- Receive_P2P (Seite 980)

Mit zusätzlichen Anweisungen kann der Empfangspuffer zurückgesetzt und es können spezielle RS232-Signale abgefragt und gesetzt werden:

- Receive_Reset (Seite 982)
- Signal_Get (Seite 983)
- Signal_Set (Seite 984)

13.3.6.1 Abfragearchitektur

Das STEP 7-Anwenderprogramm muss die Punkt-zu-Punkt-Anweisungen der S7-1200 zyklisch/ regelmäßig aufrufen, um empfangene Meldungen abzufragen. Durch Abfragen des Sendevorgangs ermittelt das Anwenderprogramm, wann die Übertragung beendet ist.

Abfragearchitektur: Master

Die typische Sequenz für einen Master ist wie folgt:

1. Die Anweisung Send_P2P (Seite 977) veranlasst eine Übertragung zum CM oder CB.
2. Die Anweisung Send_P2P wird in aufeinanderfolgenden Zyklen ausgeführt, um den Status des Übertragungsvorgangs abzufragen.
3. Wenn die Anweisung Send_P2P meldet, dass die Übertragung beendet ist, kann der Anwendercode den Empfang der Antwort vorbereiten.

4. Die Anweisung Receive-P2P (Seite 980) wird wiederholt ausgeführt, um auf eine Antwort abzufragen. Wenn das CM oder CB eine Antwortmeldung erfasst hat, kopiert die Anweisung Receive_P2P die Antwort in die CPU und meldet, dass neue Daten empfangen wurden.
5. Das Anwenderprogramm kann die Antwort verarbeiten.
6. Zurück zu Schritt 1 und Wiederholung des Zyklus.

Abfragearchitektur: Slave

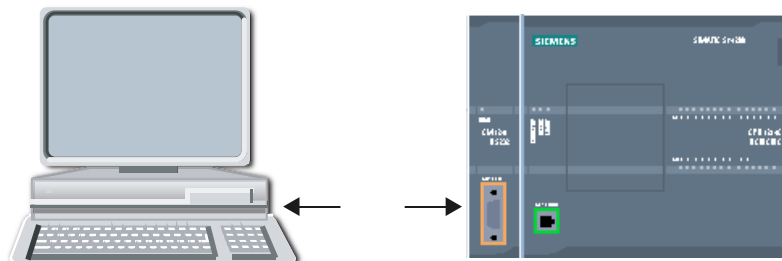
Die typische Sequenz für einen Slave ist wie folgt:

1. Das Anwenderprogramm führt die Anweisung Receive_P2P in jedem Zyklus aus.
2. Wenn das CM oder CB eine Anforderung empfangen hat, meldet die Anweisung Receive_P2P, dass neue Daten bereit sind, und die Anforderung wird in die CPU kopiert.
3. Das Anwenderprogramm verarbeitet die Anforderung und erzeugt eine Antwort.
4. Mit der Anweisung Send_P2P wird die Antwort an den Master zurückgesendet.
5. Führen Sie Send_P2P wiederholt aus, um sicherzustellen, dass der Sendevorgang stattfindet.
6. Zurück zu Schritt 1 und Wiederholung des Zyklus.

Der Slave muss dafür sorgen, dass Receive_P2P oft genug aufgerufen wird, um eine Übertragung vom Master empfangen zu können, bevor dieser beim Warten auf eine Antwort wegen Zeitüberschreitung den Vorgang abbricht. Um diese Aufgabe zu erfüllen, kann das Anwenderprogramm RCV_PTP aus einem Zyklus-OB heraus aufrufen, dessen Zykluszeit ausreichend lang ist, um eine Übertragung vom Master vor dem Ablauf der Timeout-Einstellung zu empfangen. Wird die Zykluszeit für den OB so eingestellt, dass zwei Ausführungen innerhalb der Timeout-Einstellung des Masters erfolgen, kann das Anwenderprogramm alle Übertragungen ohne Verlust empfangen.

13.3.7 Beispiel: Punkt-zu-Punkt-Kommunikation

In diesem Beispiel kommuniziert eine S7-1200 CPU mit einem PC mit einem Terminalemulator über ein CM 1241 RS232-Modul. Die Punkt-zu-Punkt-Konfiguration und das STEP 7-Programm in diesem Beispiel zeigen, wie die CPU eine Meldung vom PC empfangen und das Echo der Meldung an den PC zurückgeben kann.



Sie müssen die Kommunikationsschnittstelle des CM 1241 RS232-Moduls an die RS232-Schnittstelle des PC anschließen. Dies ist üblicherweise COM1. Weil es sich bei beiden Ports um Datenendgeräte (Data Terminal Equipment, DTE) handelt, müssen Sie die Empfangs- und

13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Sendepins (2 und 3) schalten, wenn Sie die beiden Ports anschließen. Hierzu gehen Sie auf eine der folgenden Arten vor:

- Verwenden Sie einen NULL-Modemadapter, um die Pins 2 und 3 zu tauschen, zusammen mit einem herkömmlichen RS232-Kabel.
- Verwenden Sie ein NULL-Modemkabel, bei dem die Pins 2 und 3 bereits getauscht sind. Sie erkennen ein NULL-Modemkabel üblicherweise an den zwei 9-poligen D-Buchsen.

13.3.7.1 Kommunikationsmodul konfigurieren

Sie können das CM 1241 in der Gerätekonfiguration in STEP 7 oder mit Anweisungen im Anwenderprogramm konfigurieren. In diesem Beispiel wird die Gerätekonfiguration verwendet.

- Schnittstellenkonfiguration: Klicken Sie auf den Kommunikationsport des CM in der Gerätekonfiguration und konfigurieren Sie den Port wie folgt:



Hinweis

Die Konfigurationseinstellungen für "Betriebsart" und "Ausgangszustand Empfangsleitung" gelten nur für das Modul CM 1241 (RS422/RS485). Die anderen CM 1241-Module haben diese Portkonfigurationseinstellungen nicht. Weitere Informationen finden Sie unter RS422 und RS485 konfigurieren (Seite 993).

- Konfiguration der Sendemeldung: Übernehmen Sie die Voreinstellung für die Konfiguration zum Senden von Meldungen. Bei Meldungsbeginn wird keine Pause gesendet.

- Beginn des Meldungsempfangs konfigurieren: Konfigurieren Sie das CM 1241 so, dass der Meldungsempfang beginnt, wenn die Kommunikationsleitung mindestens 50 Bitzeiten (ca. 5 ms bei 9600 Baud = $50 * 1/9600$) inaktiv ist:

> Nachrichtenbeginn

Mit beliebigem Zeichen beginnen
 Mit spezieller Bedingung beginnen
 Erkenne Nachrichtenbeginn bei Line Break
 Erkenne Nachrichtenbeginn bei Idle Line

Dauer von Idle Line: Bitzeiten

Erkenne Nachrichtenbeginn mit individuellem Zeichen
 Erkenne Nachrichtenbeginn mit einer Zeichenkette

Nachrichtenbeginn-Zeichen (HEX):

Nachrichtenbeginn-Zeichen (ASCII):

Anzahl der zu definierenden Zeichenfolgen:

- Ende des Meldungsempfangs konfigurieren: Konfigurieren Sie das CM 1241 so, dass der Meldungsempfang beendet wird, wenn maximal 100 Byte oder ein Zeilenschaltungszeichen (10 dezimal oder a hexadezimal) empfangen wurden. Als Endezeichenfolge sind maximal fünf Endezeichen in Folge zulässig. An der fünften Stelle der Folge befindet sich das Zeilenschaltungszeichen. Die vorhergehenden vier Endezeichen der Folge sind "nicht relevant" oder nicht ausgewählte Zeichen. Das CM 1241 wertet die Zeichen "nicht relevant" nicht aus, erwartet jedoch ein Zeilenschaltungszeichen mit vorhergehender 0 oder weiteren Zeichen "nicht relevant" zur Kennzeichnung des Meldungsendes.

> Nachrichtenende

Bedingungen zum Nachrichtenende definieren

Erkenne Nachrichtenende durch Nachrichten-Zeitüberschreitung
 Nachrichten-Zeitüberschreitung: ms

Erkenne Nachrichtenende durch Antwort-Zeitüberschreitung
 Antwort-Zeitüberschreitung: ms

Erkenne Nachrichtenende durch Zeitüberschreitung innerhalb der Zeichen
 Zeichenlücken-Zeitüberschreitung: Bitzeiten

Erkenne Nachrichtenende durch maximale Länge
 Maximale Länge der Nachricht: bytes

Lese Nachrichtenlänge aus Nachricht

Offset des Längenfeldes in der Nachricht: bytes

Größe des Längenfeldes: bytes

Das den Daten folgende Längenfeld gehört nicht zur N...: bytes

Erkenne Nachrichtenende mit einer Zeichenkette

5-Zeichen-Sequenz zum Nachrichtenende

<input type="checkbox"/> Frühe Zeichen 1
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 2
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 3
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input type="checkbox"/> Frühe Zeichen 4
Zeichenwert (HEX): 0
Zeichenwert (ASCII): BELIEBIG
<input checked="" type="checkbox"/> Frühe Zeichen 5
Zeichenwert (HEX): A
Zeichenwert (ASCII): LF

13.3.7.2 Betriebsarten RS422 und RS485

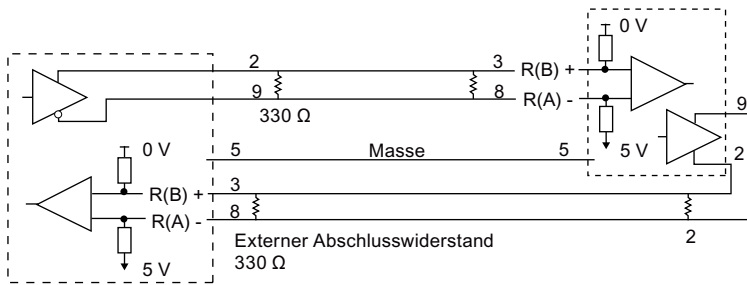
RS422 konfigurieren

Im RS422-Modus gibt es, abhängig von Ihrer Netzwerkkonfiguration, drei Betriebsarten. Wählen Sie je nach den Geräten in Ihrem Netzwerk eine dieser Betriebsarten aus. Die verschiedenen Einstellungen für "Ausgangszustand Empfangsleitung" beziehen sich auf die im Folgenden dargestellten Fälle.

- Vollduplex (RS422), Vierdrahtmodus (Punkt-zu-Punkt-Verbindung): Wählen Sie diese Option aus, wenn Ihr Netzwerk zwei Geräte umfasst. Für "Ausgangszustand Empfangsleitung":
 - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
 - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
 - Wählen Sie "Vorspannung in Sperrichtung", um interne Vorspannung und Abschluss zu verwenden und die Kabelbrucherkennung für beide Geräte zu aktivieren (Fall 1).
- Vollduplex (RS422), Vierdrahtmodus (Multipoint-Master): Wählen Sie diese Option für das Mastergerät aus, wenn Ihr Netzwerk einen Master und mehrere Slaves umfasst. Für "Ausgangszustand Empfangsleitung":
 - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
 - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
 - Kabelbrucherkennung ist in dieser Betriebsart nicht möglich.
- Vollduplex (RS422), Vierdrahtmodus (Multipoint-Slave): Wählen Sie diese Option für alle Slavegeräte aus, wenn Ihr Netzwerk einen Master und mehrere Slaves umfasst. Für "Ausgangszustand Empfangsleitung":
 - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 3).
 - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 2).
 - Wählen Sie "Vorspannung in Sperrichtung", um interne Vorspannung und Abschluss zu verwenden und die Kabelbrucherkennung für die Slaves zu aktivieren (Fall 1).

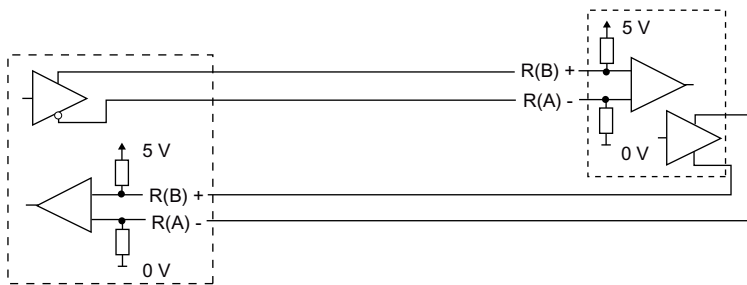
Fall 1: RS422 mit Kabelbrucherkennung

- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Vorspannung in Sperrichtung (Vorspannung mit $R(A) > R(B) > 0 V$)
- Kabelbruch: Kabelbrucherkennung aktiviert (Sender immer aktiv)



Fall 2: RS422 ohne Kabelbrucherkennung, Vorspannung in Vorwärtsrichtung

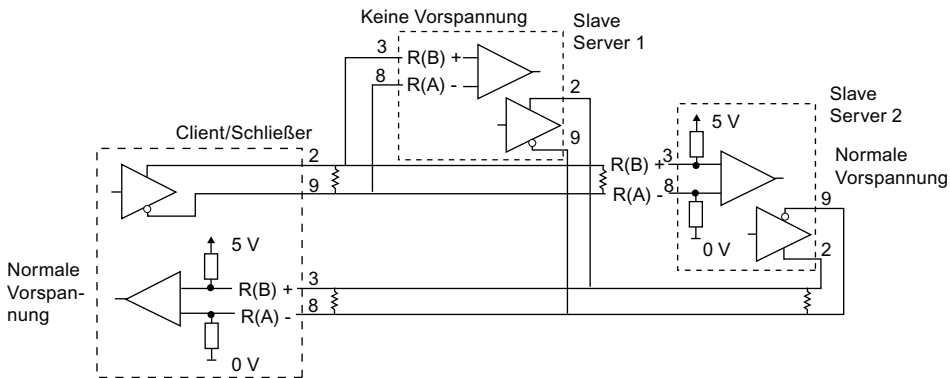
- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Vorspannung in Vorwärtsrichtung (Vorspannung mit $R(B) > R(A) > 0 V$)
- Kabelbruch: Keine Kabelbrucherkennung (Sender nur beim Senden aktiviert)



Fall 3: RS422: Keine Kabelbrucherkennung, keine Vorspannung

- Betriebsart: RS422
- Ausgangszustand Empfangsleitung: Keine Vorspannung
- Kabelbruch: Keine Kabelbrucherkennung (Sender nur beim Senden aktiviert)

Vorspannung und Abschluss werden vom Anwender an den Endknoten des Netzwerks bereitgestellt.



RS485 konfigurieren

Im RS485-Modus gibt es nur eine Betriebsart. Die verschiedenen Einstellungen für "Ausgangszustand Empfangsleitung" beziehen sich auf die im Folgenden dargestellten Fälle.

- Halbduplex (RS485), Zweidrahtmodus. Für "Ausgangszustand Empfangsleitung":
 - Wählen Sie "Ohne", wenn Sie die Vorspannung und den Abschluss vorgeben (Fall 5).
 - Wählen Sie "Vorspannung in Vorwärtsrichtung", um interne Vorspannung und Abschluss zu verwenden (Fall 4).

Fall 4: RS485: Vorspannung in Vorwärtsrichtung

- Betriebsart: RS485
- Ausgangszustand Empfangsleitung: Vorspannung in Vorwärtsrichtung (Vorspannung mit $R(B) > R(A) > 0 \text{ V}$)



Fall 5: RS485: Keine Vorspannung (externe Vorspannung)

- Betriebsart: RS485
- Ausgangszustand Empfangsleitung: Keine Vorspannung (externe Vorspannung erforderlich)



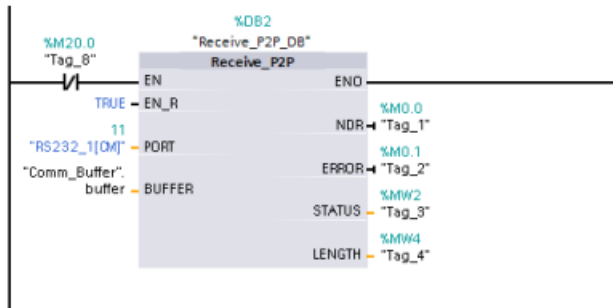
13.3.7.3 STEP 7-Programm programmieren

Das Beispielprogramm verwendet einen globalen Datenbaustein für den Kommunikationspuffer, eine Anweisung RCV_PTP (Seite 1128) für den Empfang von Daten vom Terminalemulator und eine Anweisung SEND_PTP (Seite 1125) zum Zurücksenden des Pufferechos an den Terminalemulator. Um das Beispiel zu programmieren, fügen Sie die Datenbausteinkonfiguration und den Hauptprogrammbaustein OB 1 wie im Folgenden beschrieben ein.

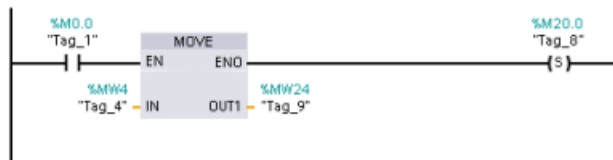
13.3 Punkt-zu-Punkt-Kommunikation (PtP)

Globaler Datenbaustein "Comm_Buffer": Erstellen Sie einen globalen Datenbaustein (DB) und nennen Sie ihn "Comm_Buffer". Erstellen Sie einen Wert im Datenbaustein, nämlich "buffer", mit dem Datentyp "Array [0 .. 99] of byte".

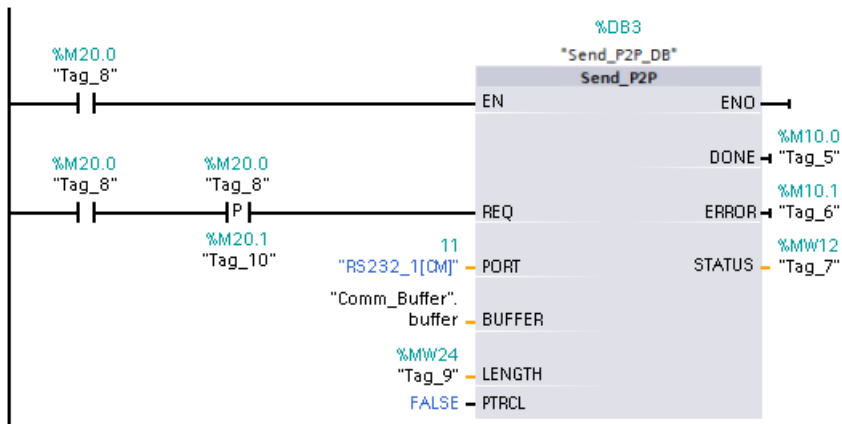
Netzwerk 1: Aktivieren Sie die Anweisung RCV_PTP immer, wenn SEND_PTP nicht aktiv ist. Tag_8 in MW20.0 zeigt in Netzwerk 4 an, wenn der Sendevorgang abgeschlossen ist und wenn das Kommunikationsmodul damit für den Meldungsempfang bereit ist.



Netzwerk 2: Erstellen Sie mit dem von der Anweisung RCV_PTP gesetzten NDR Wert (Tag_1 in M0.0) eine Kopie von der empfangenen Anzahl Bytes und setzen Sie einen Merker (Tag_8 in M20.0), um die Anweisung SEND_PTP auszulösen.



Netzwerk 3: Aktivieren Sie die Anweisung SEND_PTP, wenn der Merker M20.0 gesetzt ist. Mit diesem Merker setzen Sie auch den Eingang REQ einen Zyklus lang auf WAHR. Der Eingang REQ teilt der Anweisung SEND_PTP mit, dass eine neue Anforderung zu übertragen ist. Der Eingang REQ darf nur während einer Ausführung von SEND_PTP auf WAHR gesetzt sein. Die Anweisung SEND_PTP wird in jedem Zyklus ausgeführt, bis die Übertragung beendet ist. Die Übertragung ist beendet, wenn das letzte Byte der Meldung vom CM 1241 übertragen wurde. Wenn die Übertragung beendet ist, wird der Ausgang DONE (Tag_5 in M10.0) während einer Ausführung von SEND_PTP auf WAHR gesetzt.



Netzwerk 4: Überwachen Sie den Ausgang DONE von SEND_PTP und setzen Sie den Übertragungsmerker (Tag_8 in M20.0) zurück, wenn der Übertragungsvorgang beendet ist. Wenn der Übertragungsmerker zurückgesetzt wird, wird die Anweisung RCV_PTP in Netzwerk 1 aktiviert, um die nächste Meldung zu empfangen.



13.3.7.4 Terminalemulator konfigurieren

Sie müssen den Terminalemulator einrichten, um das Beispielprogramm auszuführen. Sie können nahezu jeden Terminalemulator an Ihrem PC verwenden, z. B. HyperTerminal. Achten Sie darauf, dass der Terminalemulator ausgeschaltet ist, bevor Sie die Einstellungen wie folgt ändern:

1. Legen Sie fest, dass der Terminalemulator den RS232-Anschluss am PC verwendet (normalerweise COM1).
2. Konfigurieren Sie den Port für 9600 Baud, 8 Datenbits, keine Parität, 1 Stoppbit und keine Flusskontrolle.
3. Ändern Sie die Einstellungen des Terminalemulators, um ein ANSI-Terminal zu emulieren.
4. Konfigurieren Sie die ASCII-Einrichtung des Terminalemulators so, dass nach jeder Zeile eine Zeilenschaltung gesendet wird (nachdem der Anwender die Eingabetaste drückt).
5. Geben Sie ein lokales Echo der Zeichen zurück, so dass der Terminalemulator anzeigt, was eingegeben wird.

13.3.7.5 Beispielprogramm ausführen

Um das Beispielprogramm auszuführen, gehen Sie folgendermaßen vor:

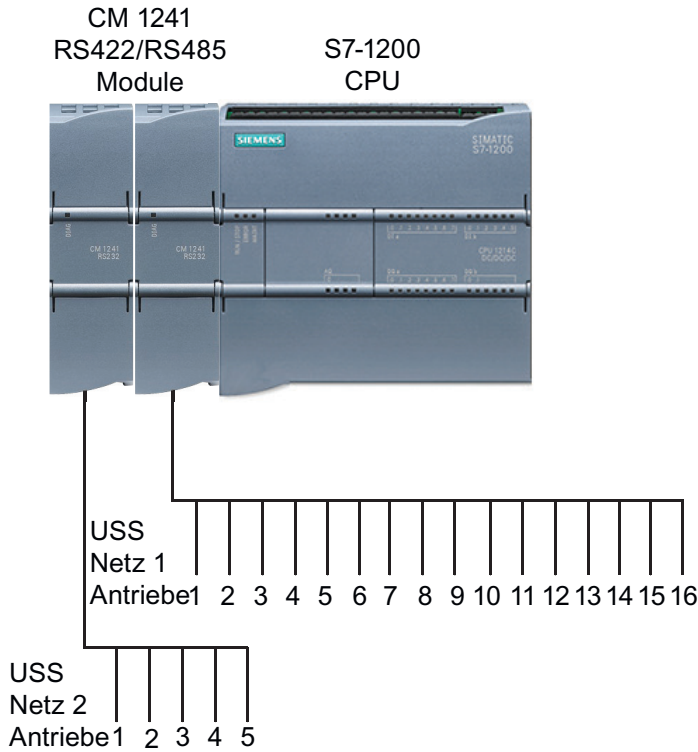
1. Laden Sie das STEP 7-Programm in die CPU und achten Sie darauf, dass diese im Betriebszustand RUN ist.
2. Klicken Sie im Terminalemulator auf die Schaltfläche zum Verbinden, um die Konfigurationsänderungen zu übernehmen, und öffnen Sie eine Terminalsitzung für das CM 1241.
3. Geben Sie am PC Zeichen ein und drücken Sie die Eingabetaste.

Der Terminalemulator sendet die Zeichen zum CM 1241 und zur CPU. Das CPU-Programm gibt dann ein Echo der Zeichen zum Terminalemulator zurück.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Die USS-Anweisungen steuern den Betrieb von Motorantrieben, die das Protokoll der universellen seriellen Schnittstelle (USS) unterstützen. Mit den USS-Anweisungen können Sie über RS485-Verbindungen mit mehreren Antrieben mit dem CM 1241 RS485-Kommunikationsmodul oder einem CB 1241 RS485-Kommunikationsboard kommunizieren. In einer S7-1200 CPU können bis zu drei CM 1241 RS422/RS485-Module und ein CB 1241 RS485-Board eingebaut werden. Jeder RS485-Port kann bis zu sechzehn Antriebe betreiben.

Das USS-Protokoll nutzt ein Master/Slave-Netzwerk für die Kommunikation über einen seriellen Bus. Der Master verwendet einen Adressparameter, um eine Meldung an einen ausgewählten Slave zu senden. Ein Slave selbst kann niemals senden, ohne dafür zuvor eine Anforderung zu erhalten. Die direkte Meldungsübertragung zwischen den einzelnen Slaves ist nicht möglich. Die USS-Kommunikation funktioniert im Halbduplex-Betrieb. Die folgende USS-Abbildung zeigt ein Netzwerkdiagramm für eine Beispielanwendung eines Antriebs.



USS-Kommunikation über PROFIBUS oder PROFINET

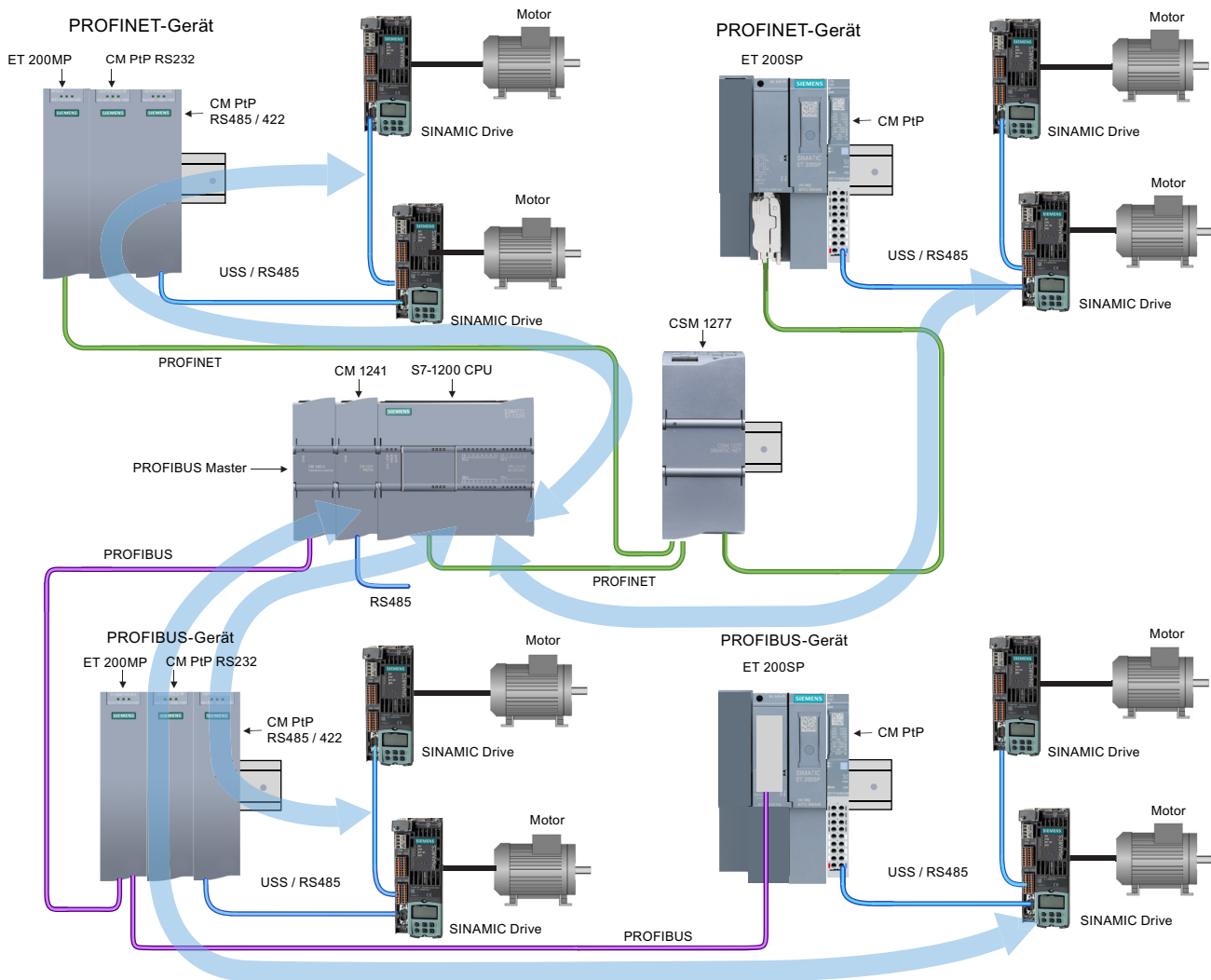
Ab Version V4.1 der S7-1200 CPU in Verbindung mit STEP 7 V13 SP1 wird die USS-Fähigkeit der CPU zur Nutzung eines dezentralen PROFINET- oder PROFIBUS-Peripheriebaugruppenträgers für die Kommunikation mit verschiedenen Geräten (RFID-Lesegeräten, GPS und anderen) erweitert:

- PROFINET (Seite 598): Die Ethernet-Schnittstelle der S7-1200 CPU wird mit einem PROFINET Schnittstellenmodul verbunden. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.
- PROFIBUS (Seite 782): Ein PROFIBUS-Kommunikationsmodul wird an der linken Seite des Baugruppenträgers mit der S7-1200 CPU eingesteckt. Das PROFIBUS-Kommunikationsmodul wird mit einem Baugruppenträger verbunden, der ein PROFIBUS-Schnittstellenmodul enthält. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.

Die S7-1200 unterstützt daher zwei Sätze PtP-Anweisungen:

- Ältere USS-Anweisungen (Seite 1137): Diese USS-Anweisungen wurden vor der Version V4.0 der S7-1200 benutzt und ermöglichen nur die serielle Kommunikation mit Hilfe eines Kommunikationsmoduls CM 1241 oder eines Kommunikationsboards CB 1241.
- USS-Anweisungen (Seite 1004): Diese USS-Anweisungen stellen alle Funktionen der alten Anweisungen bereit und zusätzlich die Möglichkeit zur Verbindung mit der dezentralen PROFINET- und PROFIBUS-Peripherie. Diese USS-Anweisungen ermöglichen das Konfigurieren der Kommunikation zwischen den PtP-Kommunikationsmodulen im Baugruppenträger mit der dezentralen Peripherie und den PtP-Geräten. S7-1200 CM 1241-Module benötigen mindestens die Firmwareversion V2.1, um diese USS-Anweisungen verwenden zu können.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)



Der blaue Pfeil kennzeichnet den bidirektionalen Kommunikationsfluss zwischen den Geräten.

Hinweis

Ab der Version V4.1 der S7-1200 können PtP-Anweisungen für alle Arten der PtP-Kommunikation verwendet werden: seriell, seriell über PROFINET und seriell über PROFIBUS. STEP 7 stellt die alten PtP-Anweisungen nur zur Unterstützung vorhandener Programme bereit. Die alten Anweisungen funktionieren noch bei allen S7-1200 CPUs. Es ist nicht nötig, ältere Programme auf die neuen Anweisungen zu konvertieren.

13.4.1 Version der USS-Anweisungen auswählen

Es gibt zwei Versionen der USS-Anweisungen in STEP 7:

- Version 2.0 (alte Anweisungen) war anfangs in STEP 7 Basic/Professional V13 verfügbar.
- Version 2.1 und höher ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Sie können nicht beide Versionen der Anweisungen bei demselben Modul verwenden, Sie können jedoch für zwei verschiedene Module unterschiedliche Versionen der Anwendungen einsetzen.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.

USS communication		V2.1
USS_Port_Scan	Communication via US...	V2.1
USS_Drive_Control	Data exchange with th...	V2.0
USS_Read_Param	Read data from drive	V2.1
USS_Write_Param	Change data in drive	V1.4

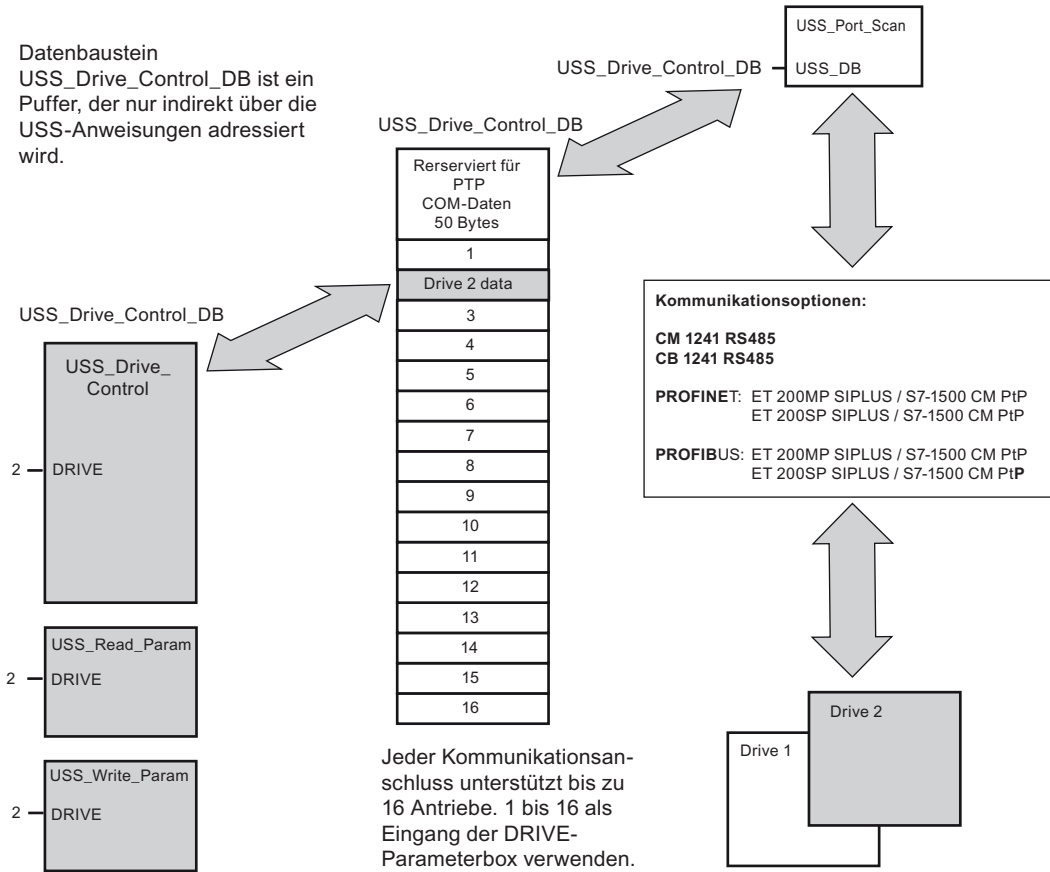
Um die Version einer USS-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine USS-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten USS-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer USS-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmierer angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines USS-FBs oder FCs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der USS-Anweisung anzuzeigen.

13.4.2 Voraussetzungen für den Einsatz des USS-Protokolls

Die vier USS-Anweisungen nutzen zwei Funktionsbausteine (FB) und zwei Funktionen (FC) für die Unterstützung des USS-Protokolls. Für jedes USS-Netzwerk wird ein Instanz-Datenbaustein (DB) USS_Port_Scan verwendet. Der Instanz-Datenbaustein USS_Port_Scan enthält temporäre Speicher und Puffer für alle Antriebe im USS-Netzwerk. Die USS-Anweisungen nutzen die Informationen in diesem Datenbaustein gemeinsam.



Alle Antriebe (max. 16), die an einen RS485-Port angeschlossen sind, sind Teil desselben USS-Netzwerks. Alle Antriebe, die an einen anderen RS485-Port angeschlossen sind, sind Teil eines anderen USS-Netzwerks. Jedes USS-Netzwerk wird mithilfe eines eindeutigen Datenbausteins verwaltet. Alle Anweisungen, die zu einem USS-Netzwerk gehören, müssen diesen Datenbaustein gemeinsam nutzen. Dies umfasst alle Anweisungen USS_Drive_Control, USS_Port_Scan, USS_Read_Param und USS_Write_Param für die Steuerung aller Antriebe in einem USS-Netzwerk.

Die Anweisung USS_Drive_Control ist ein Funktionsbaustein (FB). Wenn Sie die Anweisung USS_Drive_Control in den Programmiereditor einfügen, werden Sie im Dialog "Aufrufoptionen" aufgefordert, einen DB für diesen FB zuzuweisen. Wenn es sich um die erste Anweisung USS_Drive_Control in diesem Programm für dieses USS-Netzwerk handelt, können Sie die DB-Standardzuweisung übernehmen (oder ggf. den Namen ändern), und der neue DB wird für Sie erstellt. Wenn es sich jedoch nicht um die erste Anweisung USS_Drive_Control für diesen Kanal handelt, müssen Sie im Dialog "Aufrufoptionen" in der Klappliste den DB auswählen, der diesem USS-Netzwerk bereits zuvor zugewiesen wurde.

Die Anweisung `USS_Port_Scan` ist ein Funktionsbaustein (FB) und steuert die Kommunikation zwischen der CPU und den Antrieben über den Punkt-zu-Punkt(PtP)-RS485-Kommunikationsport. Bei jedem Aufruf dieses FB wird eine Kommunikation mit einem Antrieb bearbeitet. Ihr Programm muss diesen FB schnell genug aufrufen, so dass die Antriebe keine Zeitüberschreitung melden. Dieser FB kann aus dem Zyklus-OB des Hauptprogramms oder aus einem beliebigen Alarm-OB aufgerufen werden.

Anweisungen `USS_Read_Param` und `USS_Write_Param` sind beide Funktionen (FCs). Wenn Sie diese FCs im Editor einfügen, wird kein DB zugewiesen. Stattdessen müssen Sie dem Eingang `USS_DB` dieser Anweisungen den jeweiligen DB zuweisen. Doppelklicken Sie auf das Parameterfeld und klicken Sie dann auf das Symbol, um die verfügbaren DBs anzuzeigen.

In der Regel wird der FB `USS_Port_Scan` aus einem Weckalarm-OB aufgerufen. Stellen Sie die Zykluszeit des Weckalarm-OBs etwa auf die Hälfte eines Mindestaufrufintervalls ein (beispielsweise sollte für die Kommunikation mit 1200 Baud eine Zykluszeit von maximal 350 ms verwendet werden).

Der Funktionsbaustein `USS_Drive_Control` gibt Ihrem Programm Zugriff auf einen angegebenen Antrieb im USS-Netzwerk. Seine Ein- und Ausgänge entsprechen den Zuständen und den Bedienfunktionen des Antriebs. Sind 16 Antriebe im Netzwerk vorhanden, so muss `USS_Drive_Control` in Ihrem Programm mindestens 16 mal aufgerufen werden, also jeweils einmal für jeden Antrieb. Wie schnell diese Bausteine aufgerufen werden, hängt von der erforderlichen Geschwindigkeit für die Steuerung des Antriebsbetriebs ab.

Sie können den FB `USS_Drive_Control` nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.

VORSICHT

Beim Aufruf von USS-Anweisungen aus OBs zu beachten

Rufen Sie `USS_Drive_Control`, `USS_Read_Param` und `USS_Write_Param` nur aus einem Zyklus-OB des Hauptprogramms auf. Der FB `USS_Port_Scan` kann aus einem beliebigen OB aufgerufen werden, üblicherweise wird er aus einem Weckalarm-OB aufgerufen.

Verwenden Sie die Anweisungen `USS_Drive_Control`, `USS_Read_Param` und `USS_Write_Param` nicht in einem OB mit einer höheren Priorität als die entsprechende Anweisung `USS_Port_Scan`. Fügen Sie beispielsweise `USS_Port_Scan` nicht in einen OB des Hauptprogramms und `USS_Read_Param` in einen Weckalarm-OB ein. Wird die Unterbrechung der Ausführung von `USS_Port_Scan` nicht verhindert, kann es zu unerwarteten Fehlern kommen, die zu Verletzungen führen können.

Mit den Funktionen `USS_Read_Param` und `USS_Write_Param` werden die Betriebsparameter des entfernten Antriebs gelesen und geschrieben. Diese Parameter steuern die interne Funktionsweise des Antriebs. Eine Definition dieser Parameter finden Sie im Handbuch des Antriebs. Ihr Programm kann eine beliebige Anzahl dieser Funktionen enthalten, es kann jedoch immer nur eine Lese- oder Schreibanforderung für einen Antrieb aktiv sein. Sie können die FCs `USS_Read_Param` und `USS_Write_Param` nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.

Zeit für die Kommunikation mit dem Antrieb berechnen

Die Kommunikation mit dem Antrieb läuft asynchron zum Zyklus der S71200 ab. Die S7-1200 durchläuft üblicherweise mehrere Zyklen, bevor die Kommunikation mit einem Antrieb beendet ist.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Das Intervall USS_Port_Scan ist die Zeit, die für eine Transaktion des Antriebs erforderlich ist. Die folgende Tabelle zeigt die Mindestintervalle für USS_Port_Scan für jede Baudrate der Kommunikation. Wenn Sie den FB USS_Port_Scan häufiger aufrufen, als es das USS_Port_Scan-Intervall vorgibt, wird die Anzahl der Transaktionen nicht erhöht. Das Timeout-Intervall des Antriebs ist die Zeitdauer, die für eine Transaktion zur Verfügung steht, wenn zur Fertigstellung der Transaktion aufgrund von Kommunikationsfehlern 3 Versuche nötig sind. Standardmäßig führt die Bibliothek für das USS-Protokoll bei jeder Transaktion bis zu 2 Wiederholungen durch.

Tabelle 13-41 Zeitbedarf berechnen

Baudrate	Berechnetes Mindestintervall für Aufruf von USS_Port_Scan (ms)	Intervall-Timeout für Antriebsmeldung pro Antrieb (ms)
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 USS-Anweisungen

13.4.3.1 USS_Port_Scan (Kommunikation über USS-Netzwerk bearbeiten)

Tabelle 13-42 Anweisung USS_Port_Scan

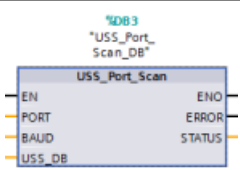
KOP/FUP	SCL	Beschreibung
	<pre> USS_Port_Scan(PORT:= _uint_in_, BAUD:= _dint_in_, ERROR=> _bool_out_, STATUS=> _word_out_, USS_DB:= _fbtref_inout_); </pre>	<p>Die Anweisung USS_Port_Scan bearbeitet die Kommunikation über ein USS-Netzwerk.</p>

Tabelle 13-43 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
PORT	IN	Port
BAUD	IN	DInt

Parameter und Datentyp		Datentyp	Beschreibung
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_Drive_Control in Ihr Programm eingefügt wird.
ERROR	OUT	Bool	Wenn WAHR, weist dieser Ausgang darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt des Ergebnis des Zyklus oder der Initialisierung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

Normalerweise ist nur eine Anweisung USS_Port_Scan pro PtP-Kommunikationsport im Programm vorhanden und jeder Aufruf dieses Funktionsbausteins (FB) steuert eine Übertragung zu oder von einem einzigen Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Instanz-DB nutzen.

Ihr Programm muss die Anweisung USS_Port_Scan so oft ausführen, dass keine Zeitüberschreitung im Antrieb auftritt. USS_Port_Scan wird üblicherweise aus einem Weckalarm-OB aufgerufen, um Zeitüberschreitungen der Antriebe zu verhindern und die letzten USS-Datenaktualisierungen für Aufrufe von USS_Drive_Control verfügbar zu halten.

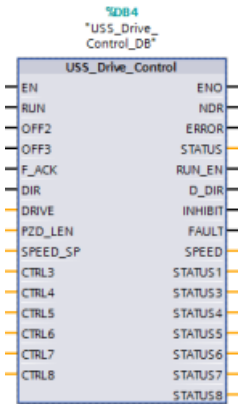
Hinweis

Wenn Sie die Bibliothek für das USS-Protokoll und die Anweisung USS_Port_Scan mit einem CB 1241 verwenden, müssen Sie für die Datenbausteinvariable LINE_PRE den Wert 0 festlegen (kein Ausgangszustand). Der Standardwert 2 für die Datenbausteinvariable LINE_PRE führt dazu, dass von der Anweisung USS_Port_Scan der Fehlerwert 16#81AB zurückgegeben wird. Die Datenbausteinvariable LINE_PRE finden Sie im Datenbaustein der Anweisung USS_Port_Scan (üblicherweise mit der Bezeichnung USS_Port_Scan_DB).

Achten Sie darauf, dass der Anfangswert von LINE_PRE in 0 (null) geändert wird.

13.4.3.2 USS_Drive_Control (Daten mit Antrieb tauschen)

Tabelle 13-44 Anweisung USS_Drive_Control

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Die Anweisung USS_Drive_Control tauscht Daten mit einem Antrieb aus, indem Anfragemeldungen erzeugt und die Antwortmeldungen des Antriebs ausgewertet werden. Es sollte für jeden Antrieb ein eigener Funktionsbaustein verwendet werden, jedoch müssen alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, den gleichen Instanz-Datenbaustein verwenden. Sie müssen den DB-Namen eingeben, wenn Sie die erste Anweisung USS_Drive_Control einfügen. Dann verweisen Sie auf diesen DB, der beim Einfügen der ersten Anweisung angelegt wurde.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>

¹ KOP und FUP: Erweitern Sie die Box, um alle Parameter anzuzeigen. Klicken Sie dazu auf den unteren Bereich der Box. Die Parameteranschlüsse, die grau dargestellt sind, sind optional, eine Parametrierung ist nicht erforderlich.

Tabelle 13-45 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
RUN	IN	Bool	Startbit des Antriebs: Ist dieser Parameter WAHR, so ermöglicht dieser Eingang den Betrieb des Antriebs mit der voreingestellten Drehzahl. Wenn RUN im Betrieb des Antriebs nach Falsch wechselt, läuft der Motor bis zum Stillstand aus. Dieses Verhalten unterscheidet sich von der Abschaltung der Spannungsversorgung (OFF2) und vom Bremsen des Motors (OFF3).
OFF2	IN	Bool	Bit "Zum Stillstand auslaufen": Ist dieser Parameter FALSCH, so veranlasst dieses Bit das Auslaufen des Antriebs, ohne zu bremsen.
OFF3	IN	Bool	Schnelles Stoppbit: Ist dieser Parameter FALSCH, so verursacht dieses Bit einen schnellen Halt durch Abbremsen des Antriebs.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Parameter und Datentyp		Datentyp	Beschreibung
F_ACK	IN	Bool	Fehlerquittierungsbit: Mit diesem Bit wird das Fehlerbit eines Antriebs zurückgesetzt. Das Bit wird nach dem Löschen des Fehlers gesetzt und der Antrieb erkennt damit, dass der vorherige Fehler nicht mehr gemeldet werden muss.
DIR	IN	Bool	Richtungssteuerung des Antriebs: Dieses Bit wird gesetzt, wenn der Antrieb in Vorwärtsrichtung laufen soll (wenn SPEED_SP positiv ist).
DRIVE	IN	USInt	Adresse des Antriebs: Dieser Eingang ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PZD_LEN	IN	USInt	Wortlänge: Dies ist die Anzahl der PZD-Datenwörter. Gültige Werte sind 2, 4, 6 oder 8 Wörter. Der Standardwert ist 2.
SPEED_SP	IN	Real	Drehzahlsollwert: Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Frequenz. Ein positiver Wert bedeutet, dass der Antrieb vorwärts läuft (wenn DIR wahr ist). Gültig ist der Bereich von 200,00 bis -200,00.
CTRL3	IN	Word	Steuerwort 3: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL4	IN	Word	Steuerwort 4: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL5	IN	Word	Steuerwort 5: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL6	IN	Word	Steuerwort 6: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL7	IN	Word	Steuerwort 7: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL8	IN	Word	Steuerwort 8: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
NDR	OUT	Bool	Neue Daten bereit: Ist dieser Parameter wahr, so meldet das Bit, dass am Ausgang Daten einer neuen Kommunikationsanforderung bereitstehen.
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist dies darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_Port_Scan gemeldet.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt das Ergebnis des Zyklus an. Dies ist kein vom Antrieb ausgegebenes Zustandswort.
RUN_EN	OUT	Bool	Betrieb freigegeben: Dieses Bit meldet, ob der Antrieb läuft.
D_DIR	OUT	Bool	Antriebsrichtung: Dieses Bit meldet, ob der Antrieb vorwärts läuft.
INHIBIT	OUT	Bool	Antrieb gesperrt: Dieses Bit meldet den Zustand des Sperrbits für den Antrieb.
FAULT	OUT	Bool	Antriebsfehler: Dieses Bit meldet, dass im Antrieb ein Fehler aufgetreten ist. Sie müssen die Störung beheben und Bit F_ACK setzen, um dieses Bit zu löschen.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Parameter und Datentyp		Datentyp	Beschreibung
SPEED	OUT	Real	Istwert Antriebsdrehzahl (skalierter Wert von Zustandswort 2 des Antriebs): Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Drehzahl.
STATUS1	OUT	Word	Zustandswort 1 des Antriebs: Dieser Wert enthält feste Zustandsbits eines Antriebs.
STATUS3	OUT	Word	Zustandswort 3 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS4	OUT	Word	Zustandswort 4 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS5	OUT	Word	Zustandswort 5 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS6	OUT	Word	Zustandswort 6 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS7	OUT	Word	Zustandswort 7 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS8	OUT	Word	Zustandswort 8 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.

Wenn die erste Ausführung von USS_Drive_Control erfolgt, wird der von der USS-Adresse (Parameter DRIVE) angegebene Antrieb im Instanz-DB initialisiert. Nach dieser Initialisierung können nachstehende Anweisungen USS_Port_Scan die Kommunikation mit dem Antrieb an dieser Antriebsnummer beginnen.

Wenn Sie die Antriebsnummer ändern, muss die CPU zunächst in STOP und dann wieder in RUN versetzt werden, damit der Instanz-DB initialisiert wird. Die Eingangsparameter werden im USS-Sendepuffer konfiguriert und die Ausgänge werden, sofern vorhanden, aus einem "vorherigen" gültigen Antwortpuffer gelesen. Während der Ausführung der Anweisung USS_Drive_Control findet keine Datenübertragung statt. Nach der Ausführung von USS_Port_Scan kommunizieren die Antriebe. USS_Drive_Control konfiguriert nur die zu sendenden Meldungen und wertet Daten aus, die möglicherweise mit einer vorherigen Anforderung empfangen wurden.

Sie können die Drehrichtung des Antriebs entweder über den Eingang DIR (Bool) oder über das Vorzeichen (positiv oder negativ) am Eingang SPEED_SP (Real) steuern. Die folgende Tabelle erläutert, wie diese Eingänge zusammen funktionieren, um die Drehrichtung des Antriebs zu bestimmen, vorausgesetzt der Motor dreht vorwärts.

Tabelle 13-46 Interaktion der Parameter SPEED_SP und DIR

SPEED_SP	DIR	Drehrichtung des Antriebs
Wert > 0	0	Rückwärts
Wert > 0	1	Vorwärts
Wert < 0	0	Vorwärts
Wert < 0	1	Rückwärts

13.4.3.3 USS_Read-Param (Parameter aus dem Antrieb auslesen)

Tabelle 13-47 Anweisung USS_Read_Param

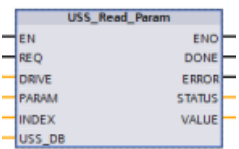
KOP/FUP	SCL	Beschreibung
	<pre>USS_Read_Param(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, VALUE=>_variant_out_, USS_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung USS_Read_Param liest einen Parameter aus einem Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationssport zugewiesen sind, müssen den gleichen Datenbaustein verwenden. USS_Read_Param muss aus einem Zyklus-OB des Hauptprogramms aufgerufen werden.</p>

Tabelle 13-48 Datentypen für die Parameter

Parametertyp		Datentyp	Beschreibung
REQ	IN	Bool	Sendeanforderung: Ist REQ WAHR, so wird eine neue Leseanforderung benötigt. Dies wird ignoriert, wenn die Anforderung für diesen Parameter bereits ansteht.
DRIVE	IN	USInt	Adresse des Antriebs: DRIVE ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PARAM	IN	UInt	Parameternummer: PARAM gibt an, welcher Antriebsparameter geschrieben wird. Der Bereich für diesen Parameter liegt zwischen 0 und 2047. Bei einigen Antrieben kann das höchstwertige Byte auf PARAM-Werte größer als 2047 zugreifen. Weitere Informationen für den Zugriff auf einen erweiterten Bereich finden Sie im Handbuch zu Ihrem Antrieb.
INDEX	IN	UInt	Parameterindex: INDEX gibt an, in welchen Antriebsparameterindex geschrieben werden soll. Es handelt sich um einen 16-Bit-Wert, bei dem das niederwertigste Byte der tatsächliche Indexwert ist, mit einem Bereich von (0 bis 255). Das höchstwertige Byte kann ebenfalls von dem Antrieb verwendet werden und ist antriebspezifisch. Weitere Informationen finden Sie im Handbuch Ihres Antriebs.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_Drive_Control in Ihr Programm eingefügt wird.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Dies ist der Wert des Parameters, der gelesen wurde und er ist nur gültig, wenn das Bit DONE wahr ist.
DONE ¹	OUT	Bool	Ist dieser Parameter WAHR, so steht am Ausgang VALUE der zuvor angeforderte Wert des Leseparameters an. Dieses Bit wird gesetzt, wenn die Anweisung USS_Drive_Control die Leseantwort des Antriebs erkennt. Dieses Bit wird zurückgesetzt, wenn: Sie die Antwortdaten über eine andere Abfrage USS_Read_Param anfordern oder beim zweiten der nächsten beiden Aufrufe von USS_Drive_Control.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Parametertyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_Port_Scan gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Leseanforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

¹ Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf eines USS_Read_Param- oder USS_Write_Param -FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

13.4.3.4 USS_Write_Param (Parameter im Antrieb ändern)

Hinweis

EEPROM-Schreibanweisungen (für den EEPROM in einem USS-Antrieb)

Übertreiben Sie die Verwendung der EEPROM-Schreiboperation nicht. Halten Sie die Anzahl der EEPROM-Schreiboperationen möglichst gering, um die Lebensdauer des EEPROM zu verlängern.

Tabelle 13-49 Anweisung USS_Write_Param

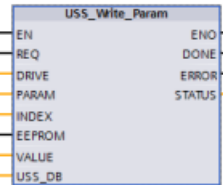
KOP/FUP	SCL	Beschreibung
	<pre> USS_Write_Param(REQ:=_bool_in_ _/, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, EEPROM:=_bool_in_, VALUE:=_variant_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, USS_DB:=_fbtref_inout_); </pre>	<p>Die Anweisung USS_Write_Param ändert einen Parameter im Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Datenbaustein nutzen.</p> <p>USS_Write_Param muss aus dem Zyklus-OB eines Hauptprogramms aufgerufen werden.</p>

Tabelle 13-50 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Sendeanforderung: Ist REQ WAHR, so wird eine neue Schreibanforderung benötigt. Dies wird ignoriert, wenn die Anforderung für diesen Parameter bereits ansteht.
DRIVE	IN	USInt	Adresse des Antriebs: DRIVE ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.

13.4 Kommunikation über die universelle serielle Schnittstelle (USS)

Parameter und Datentyp		Datentyp	Beschreibung
PARAM	IN	UInt	Parameter Nummer: PARAM gibt an, welcher Antriebsparameter geschrieben wird. Der Bereich für diesen Parameter liegt zwischen 0 und 2047. Bei einigen Antrieben kann das höchstwertige Byte auf PARAM-Werte größer als 2047 zugreifen. Weitere Informationen für den Zugriff auf einen erweiterten Bereich finden Sie im Handbuch zu Ihrem Antrieb.
INDEX	IN	UInt	Parameterindex: INDEX gibt an, in welchen Antriebsparameterindex geschrieben werden soll. Es handelt sich um einen 16-Bit-Wert, bei dem das niederwertigste Byte der tatsächliche Indexwert ist, mit einem Bereich von (0 bis 255). Das höchstwertige Byte kann ebenfalls von dem Antrieb verwendet werden und ist antriebsspezifisch. Weitere Informationen finden Sie im Handbuch Ihres Antriebs.
EEPROM	IN	Bool	Im EEPROM des Antriebs speichern: Wenn WAHR, wird die Transaktion eines Parameters zum Schreiben in den Antrieb im EEPROM des Antriebs gespeichert. Wenn FALSCH, so wird der geschriebene Wert nur temporär gespeichert und geht beim nächsten Einschalten des Antriebs verloren.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Wert des Parameters, in den geschrieben werden soll. Er muss beim Zustandswechsel von REQ gültig sein.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_Drive_Control in Ihr Programm eingefügt wird.
DONE ¹	OUT	Bool	Wenn WAHR, weist DONE darauf hin, dass der Eingang VALUE in den Antrieb geschrieben wurde. Dieses Bit wird gesetzt, wenn die Anweisung USS_Drive_Control die Schreibantwort des Antriebs erkennt. Dieses Bit wird zurückgesetzt, entweder wenn Sie die Antwortdaten über eine weitere Abfrage USS_Drive_Control anfordern, oder beim zweiten der nächsten beiden Aufrufe von USS_Drive_Control.
ERROR	OUT	Bool	Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_Port_Scan gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Schreibenanforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

¹ Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf eines USS_Read_Param- oder USS_Write_Param-FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

13.4.4 USS-Zustandcodes

Statuscodes der USS-Anweisung werden im Ausgang STATUS der USS-Funktionen ausgegeben.

Tabelle 13-51 STATUS-Codes ¹

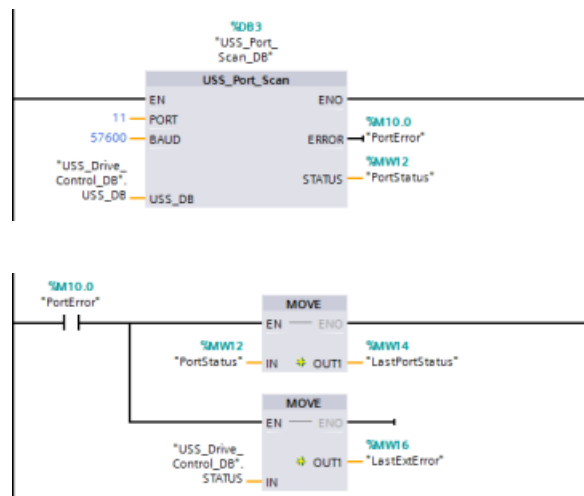
STATUS (W#16#...)	Beschreibung
0000	Kein Fehler
8180	Die Länge der Antwort des Antriebs entsprach nicht den vom Antrieb empfangenen Zeichen. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8181	Der Parameter VALUE gehört nicht zum Datentyp Wort, Real oder Doppelwort.
8182	Der Anwender hat einen Parameterwert vom Typ Wort eingegeben und die Antwort vom Antrieb im Format Doppelwort oder Real empfangen.
8183	Der Anwender hat einen Parameterwert vom Typ Doppelwort oder Real eingegeben und die Antwort vom Antrieb im Format Wort empfangen.
8184	Das Antworttelegramm des Antriebs hatte eine falsche Prüfsumme. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8185	Unzulässige Antriebsadresse (gültiger Adressbereich für Antriebe: 1 bis16)
8186	Der Drehzahl Sollwert liegt außerhalb des gültigen Bereichs (gültiger Sollwertbereich für die Drehzahl: -200 % bis 200 %).
8187	Die falsche Antriebsnummer hat auf die gesendete Anforderung geantwortet. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8188	Unzulässige PZD-Wortlänge angegeben (gültiger Bereich = 2, 4, 6 oder 8 Wörter)
8189	Unzulässige Baudrate angegeben.
818A	Der Anforderungskanal für Parameter wird von einer anderen Anforderung für diesen Antrieb verwendet.
818B	Der Antrieb hat nicht auf Anforderungen und Wiederholungen reagiert. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818C	Der Antrieb hat einen erweiterten Fehler zu einer Parameteranforderung ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818D	Der Antrieb hat einen Fehler "Unzulässiger Zugriff" zu einer Parameteranforderung ausgegeben. Weitere Informationen dazu, weshalb der Parameterzugriff begrenzt sein kann, finden Sie im Handbuch zu Ihrem Antrieb.
818E	Der Antrieb wurde nicht initialisiert. Dieser Fehlercode wird an USS_Read_Param oder USS_Write_Param ausgegeben, wenn die Anweisung USS_Drive_Control nicht mindestens einmal für diesen Antrieb aufgerufen wurde. Dadurch wird verhindert, dass die Initialisierung im ersten Zyklus von USS_Drive_Control eine anstehende Anforderung zum Lesen oder Schreiben von Parametern überschreibt, weil dabei der Antrieb als neuer Eintrag initialisiert wird. Um diesen Fehler zu beheben, rufen Sie die Anweisung USS_Drive_Control für diesen Antrieb auf.
80Ax-80Fx	Spezifische Fehler, die von den von der USS-Bibliothek aufgerufenen FBs für die PtP-Kommunikation zurückgegeben werden - Diese Fehlercodes werden von der USS-Bibliothek nicht verändert und sind in den Beschreibungen der PtP-Anweisung definiert.

¹ Neben den oben aufgeführten Fehlern der USS-Anweisungen können auch die zugrunde liegenden PtP-Kommunikationsanweisungen (Seite 963) Fehler zurückgeben.

Für verschiedene STATUS-Codes werden weitere Informationen in der Variable "USS_Extended_Error" des Instanz-DB USS_Drive_Control zur Verfügung gestellt. Für die STATUS-Codes hexadezimal 8180, 8184, 8187 und 818B, enthält USS_Extended_Error die Antriebsnummer des Antriebs, bei dem der Kommunikationsfehler auftrat. Für STATUS-Code hexadezimal 818C enthält USS_Extended_Error einen Antriebsfehlercode, der bei der Verwendung einer Anweisung USS_Read_Param oder USS_Write_Param vom Antrieb ausgegeben wird.

Beispiel: Meldung von Kommunikationsfehlern

Kommunikationsfehler (STATUS = 16#818B) werden nur bei der Anweisung USS_Port_Scan und nicht bei der Anweisung USS_Drive_Control gemeldet. Beispiel: Wenn das Netzwerk nicht ordnungsgemäß beendet wird, kann ein Antrieb in RUN wechseln, doch die Anweisung USS_Drive_Control zeigt an allen Ausgangsparametern 0 an. In diesem Fall können Sie den Kommunikationsfehler nur über die Anweisung USS_Port_Scan erkennen. Weil dieser Fehler nur einen Zyklus lang sichtbar ist, müssen Sie entsprechende Erfassungslgik einfügen. Dies wird in dem folgenden Beispiel dargestellt. In diesem Beispiel werden, wenn das Fehlerbit der Anweisung USS_Port_Scan WAHR ist, die Werte STATUS und USS_Extended_Error im Speicherbereich der Merker abgelegt. Die Antriebsnummer wird in der Variablen USS_Extended_Error abgelegt, wenn der Wert des STATUS-Codes hexadezimal 8180, 8184, 8187 oder 818B ist.



Netzwerk 1 Der Portzustand "PortStatus" und die erweiterten Fehlercode-werte "USS_Drive_Control_DB".USS_Extended_Error sind nur einen Programmzyklus lang gültig. Die Werte müssen zur späteren Bearbeitung erfasst werden.

Netzwerk 2 Der "PortError"-Kontakt löst die Speicherung des "PortStatus"-Werts in "LastPortStatus" und des "USS_Drive_Control_DB".USS_Extended_Error-Werts in "LastExtError" aus.

Lese- und Schreibzugriff auf die internen Parameter eines Antriebs

USS-Antriebe unterstützen Lese- und Schreibzugriff auf die internen Parameter eines Antriebs. Diese Funktion ermöglicht die dezentrale Steuerung und Konfiguration des Antriebs. Zugriffe der Antriebsparameter können aufgrund von Fehlern wie Wert außerhalb des Bereichs oder unzulässige Anforderungen in der aktuellen Betriebsart des Antriebs fehlschlagen. Der Antrieb erzeugt einen Fehlercode, der in der Variablen "USS_Extended_Error" ausgegeben wird. Dieser Fehlercode gilt nur für die letzte Ausführung der Anweisung USS_Read_Param oder USS_Write_Param. Der Fehlercode des Antriebs wird in der Variablen "USS_Extended_Error" abgelegt, wenn der Wert von STATUS code hexadezimal 818C ist. Der Fehlercodewert von USS_Extended_Error richtet sich nach dem Antriebsmodell. Eine Beschreibung der erweiterten Fehlercodes von Lese- und Schreibfunktionen für Parameter finden Sie im Handbuch des Antriebs.

13.4.5 Allgemeine Voraussetzungen für die USS-Antriebseinrichtung

Für die USS-Antriebseinrichtung gelten die folgenden allgemeinen Voraussetzungen:

- Für die Antriebe muss die Verwendung von 4 PKW-Wörtern eingerichtet werden.
- Die Antriebe können für 2, 4, 6 oder 8 PZD-Wörter konfiguriert werden.
- Die Anzahl der PZD-Wörter im Antrieb muss dem Eingang PZD_LEN der Anweisung USS_Drive_Control des Antriebs entsprechen.
- Die Baudrate aller Antriebe muss dem Eingang BAUD der Anweisung USS_Port_Scan entsprechen.
- Der Antrieb muss für die Fernsteuerung eingerichtet werden.
- Für den Frequenzsollwert an der COM-Verbindung des Antriebs muss USS festgelegt werden.
- Für die Antriebsadresse muss 1 bis 16 festgelegt sein. Diese Adresse muss dem Eingang DRIVE am Baustein USS_Drive_Control des Antriebs entsprechen.
- Für die Richtungssteuerung des Antriebs muss die Verwendung der Polarität des Antriebssollwerts eingerichtet werden.
- Das RS485-Netzwerk muss ordnungsgemäß abgeschlossen sein.

13.4.6 Beispiel: Allgemeine USS-Antriebsverbindung und -einrichtung

MicroMaster-Antrieb anschließen

Diese Informationen zu SIEMENS MicroMaster-Antriebe dienen als Beispiel. Bei anderen Antrieben finden Sie die Einrichtungsanleitung im Handbuch des Antriebs.

Wenn Sie einen MicroMaster-Antrieb der Serie 4 (MM4) anschließen möchten, stecken Sie die Enden des RS485-Kabels in die beiden schraubenlosen Druckklemmen für den USS-Betrieb. Die

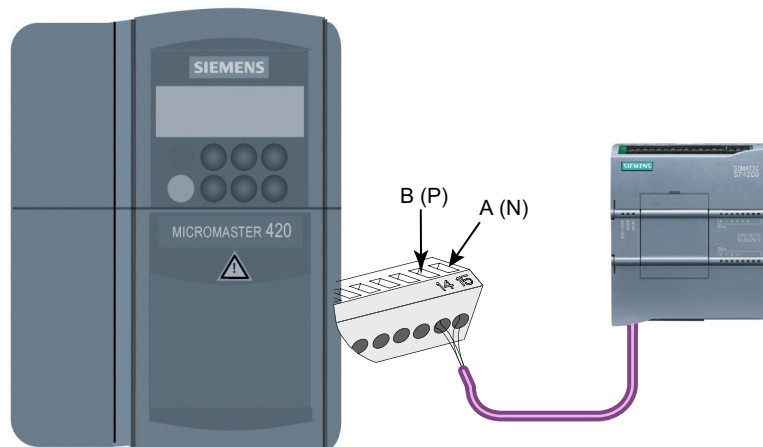
S7-1200 kann mit dem herkömmlichen PROFIBUS-Kabel und den Steckverbindern an den MicroMaster-Antrieb angeschlossen werden.

⚠ VORSICHT

Wenn Sie Geräte miteinander verbinden, die nicht die gleiche Bezugsspannung haben, kann dies unerwünschte Ströme im Verbindungskabel hervorrufen.

Diese unerwünschten Ströme können Kommunikationsfehler verursachen oder Sachschaden in den Geräten hervorrufen. Stellen Sie sicher, dass alle Geräte, die Sie über ein Kommunikationskabel miteinander verbinden, entweder den gleichen Bezugsleiter im Stromkreis haben oder elektrisch getrennt sind, damit keine unerwünschten Ströme auftreten. Die Schirmung muss mit Masse oder Pin 1 des 9-poligen Steckverbinders verbunden werden. Die Verdrahtungsklemme 2-0 V des MicroMaster-Antriebs muss mit Masse verbunden werden.

Stecken Sie die beiden Drähte am gegenüberliegenden Ende des RS485-Kabels in den Klemmenblock des MM4-Antriebs. Zum Herstellen der Kabelverbindung am MM4-Antrieb entfernen Sie die Abdeckung(en) des Antriebs, damit Sie Zugriff auf die Klemmenblöcke haben. Im Benutzerhandbuch des MM4-Antriebs finden Sie eine ausführliche Beschreibung, wie Sie die Abdeckung(en) an Ihrem spezifischen Antrieb entfernen.



Die Anschlüsse am Klemmenblock sind durchnummeriert. Verwenden Sie auf der Seite der S7-1200 einen PROFIBUS-Anschlussstecker und schließen Sie die Klemme A des Kabels an Klemme 15 des Antriebs (bei einem MM420) oder an Klemme 30 des Antriebs (bei einem MM440) an. Schließen Sie die Klemme B (P) A (N) des Kabelverbinders an Klemme 14 (bei einem MM420) oder an Klemme 29 (bei einem MM440) an.

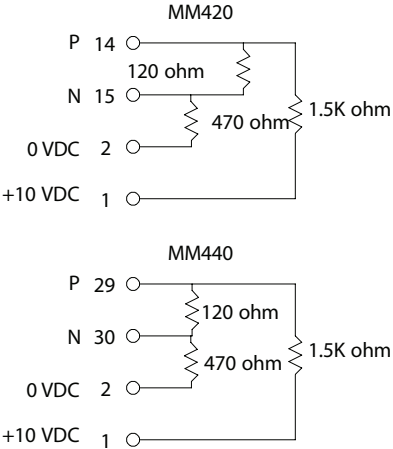
Handelt es sich bei der S7-1200 um einen abschließenden Teilnehmer im Netz oder um eine Punkt-zu-Punkt-Verbindung, müssen Sie die Klemmen A1 und B1 (nicht A2 und B2) des Steckverbinders verwenden, weil diese Klemmen die Abschlusseinstellungen ermöglichen (z. B. beim DP-Steckverbinder 6ES7972-0BA40-0X40).

⚠ VORSICHT

Bauen Sie die Abdeckungen des Antriebs sorgfältig ein, bevor Sie das Gerät wieder mit Spannung versorgen

Achten Sie darauf, dass Sie die Abdeckungen des Antriebs sorgfältig wieder einbauen, bevor Sie das Gerät mit Spannung versorgen.

Ist der Antrieb als abschließender Teilnehmer im Netz konfiguriert, müssen außerdem Abschlusswiderstände mit den entsprechenden Klemmen verdrahtet werden. Diese Abbildung zeigt Beispiele für die Anschlüsse des MM4-Antriebs für den Abschluss.



Einrichten des MM4-Antriebs

Bevor Sie einen Antrieb an die S7-1200 anschließen, müssen Sie sicherstellen, dass der Antrieb über folgende Systemparameter verfügt. Sie stellen die Parameter mit der Tastatur des Antriebs ein:

1. Setzen Sie den Antrieb auf die Werkseinstellungen zurück (optional).	P0010 = 30 P0970 = 1
Wenn Sie Schritt 1 überspringen, müssen Sie darauf achten, dass diese Parameter auf die angegebenen Werte gesetzt sind:	USS PZD-Länge = P2012 Index 0 = (2, 4, 6 oder 8) USS PKW-Länge = P2013 Index 0 = 4
2. Aktivieren Sie den Lese-/Schreibzugriff für alle Parameter (Expertenmodus).	P0003 = 3
3. Prüfen Sie die Motoreinstellungen Ihres Antriebs. Die Einstellungen richten sich nach dem jeweiligen Motor. Damit Sie die Parameter P304, P305, P307, P310 und P311 einstellen können, müssen Sie zunächst den Parameter P010 auf 1 setzen (Modus für Schnellinbetriebsetzung). Nachdem Sie die Parameter eingestellt haben, setzen Sie den Parameter P010 auf 0. Die Parameter P304, P305, P307, P310 und P311 können nur im Modus für Schnellinbetriebsetzung geändert werden.	P0304 = Motornennspannung (V) P0305 = Motornennstrom (A) P0307 = Motornennleistung (W) P0310 = Motornennfrequenz (Hz) P0311 = Motornennzahl
4. Stellen Sie den lokalen/entfernten Modus ein.	P0700 Index 0 = 5
5. Stellen Sie den Frequenzsollwert der COM-Verbindung auf USS ein.	P1000 Index 0 = 5
6. Hochlaufzeit (optional) Zeit in Sekunden, die der Motor braucht, um auf maximale Frequenz zu beschleunigen.	P1120 = (0 bis 650,00)

7. Auslaufzeit (optional) Zeit in Sekunden, die der Motor braucht, um bis zum vollständigen Halt abzubremsen.	P1121 = (0 bis 650,00)
8. Stellen Sie die Referenzfrequenz der seriellen Verbindung ein:	P2000 = (1 bis 650 Hz)
9. Stellen Sie die USS-Normalisierung ein:	P2009 Index 0 = 0
10. Stellen Sie die Baudrate der seriellen Schnittstelle RS485 ein:	P2010 Index 0 = 4 (2400 Baud) 5 (4800 Baud) 6 (9600 Baud) 7 (19200 Baud) 8 (38400 Baud) 9 (57600 Baud) 12 (115200 Baud)
11. Geben Sie die Slave-Adresse ein. Jeder Antrieb (maximal 31) kann über den Bus betrieben werden.	P2011 Index 0 = (0 bis 31)
12. Legen Sie den Timeout-Wert für die serielle Verbindung fest. Dies ist der maximal zulässige Zeitraum zwischen zwei eingehenden Datentelegrammen. Diese Funktion schaltet den Inverter bei Kommunikationsausfall aus. Die Zeit wird gemessen, nachdem ein gültiges Telegramm empfangen wurde. Wird kein weiteres Datentelegramm innerhalb der angegebenen Zeit empfangen, schaltet der Inverter ab und zeigt Fehlercode F0070 an. Wenn Sie den Wert auf Null setzen, wird die Steuerung ausgeschaltet.	P2014 Index 0 = (0 bis 65.535 ms) 0 = Timeout deaktiviert
13. Übertragen Sie die Daten vom RAM zum EEPROM:	P0971 = 1 (Übertragung starten). Speichern Sie die Änderungen an den Parametereinstellungen im EEPROM.

13.5 Modbus-Kommunikation

13.5.1 Überblick zur Kommunikation mittels Modbus RTU und Modbus TCP

Modbus-Funktionscodes

- Eine CPU, die als Modbus RTU-Master (oder Modbus TCP-Client) betrieben wird, kann Daten und E/A-Zustände in einem dezentralen Modbus RTU-Slave (bzw. Modbus TCP-Server) lesen und schreiben. Dezentrale Daten können von Ihrer Programmlogik gelesen und dann verarbeitet werden.
- Eine CPU, die als Modbus RTU-Slave (oder Modbus TCP-Server) betrieben wird, ermöglicht es einem übergeordneten Gerät, Daten und E/A-Zustände im Speicher der CPU zu lesen und zu schreiben. Ein RTU-Master (oder Modbus TCP-Client) kann neue Werte in den für Ihre Programmlogik verfügbaren Speicher einer Slave/Server-CPU schreiben.


 WARNUNG
<p>Sicherheitsrisiken durch physische Netzwerkangriffe vermeiden</p> <p>Wenn sich ein Angreifer physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.</p> <p>Einige andere Arten der Kommunikation (E/A-Austausch über PROFIBUS, PROFINET, AS-i oder einen anderen E/A-Bus, GET/PUT, T-Block und Kommunikationsmodule (CM)) haben keine Sicherheitsfunktionen. Sie müssen diese Arten der Kommunikation durch Begrenzung des physischen Zugriffs schützen. Wenn sich ein Angreifer über diese Arten der Kommunikation physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.</p> <p>Wenn Sie diese Arten der Kommunikation nicht schützen, kann das zu Tod oder schwerem Personenschaden führen.</p> <p>Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren "Operational Guidelines für Industrial Security" auf der Website "Service & Support" von Siemens.</p>

Tabelle 13-52 Funktionen zum Lesen von Daten: Dezentrale E/A und Programmdateien lesen

Modbus-Funktionscode	Funktionen zum Lesen des Slave (Server) - Standardadressierung
01	Ausgangsbits lesen: 1 bis 2000 Bits pro Anforderung
02	Eingangsbits lesen: 1 bis 2000 Bits pro Anforderung
03	Halteregister lesen: 1 bis 125 Bits pro Anforderung
04	Eingangswörter lesen: 1 bis 125 Bits pro Anforderung

Tabelle 13-53 Funktionen zum Schreiben von Daten: Dezentrale E/A schreiben und Programmdateien ändern

Modbus-Funktionscode	Funktionen zum Schreiben in den Slave (Server) - Standardadressierung
05	Ein Ausgangsbit schreiben: 1 Bit pro Anforderung
06	Ein Halteregister schreiben: 1 Wort pro Anforderung
15	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bits pro Anforderung
16	Ein oder mehrere Halteregister schreiben: 1 bis 123 Bits pro Anforderung

- Die Modbus-Funktionscodes 08 und 11 bieten Diagnoseinformationen für die Kommunikation des Slavegeräts.
- Modbus-Funktionscode 0 sendet eine Broadcast-Meldung an alle Slaves (ohne Slaveantwort). Die Broadcast-Funktion ist bei Modbus TCP nicht verfügbar, weil die Kommunikation verbindungs-basiert abläuft.
- Modbus-Funktionscode 23 kann ein oder mehrere Halteregister schreiben und lesen: 1 bis 121/125 (Schreiben/Lesen) Wörter pro Anforderung. Dieser Funktionscode steht nur für Modbus TCP zur Verfügung.

Tabelle 13-54 Stationsadressen im Modbus-Netzwerk

Station		Adresse
RTU-Station	Standardstationsadresse	1 bis 247
	Erweiterte Stationsadresse	1 bis 65535
TCP-Station	Stationsadresse	IP-Adresse und Portnummer

Modbus-Speicheradressen

Die tatsächlich verfügbare Anzahl von Modbus-Speicheradressen richtet sich nach der CPU-Variante, nach dem verfügbaren Arbeitsspeicher und danach, wie viel CPU-Speicher durch andere Programmdaten belegt ist. Die folgende Tabelle zeigt den Nennwert des Adressbereichs.

Tabelle 13-55 Modbus-Speicheradressen

Station		Adressbereich
RTU-Station	Adresse im Standardspeicher	10 K
	Adresse im erweiterten Speicher	64 K
TCP-Station	Adresse im Standardspeicher	10 K

Modbus RTU-Kommunikation

Modbus RTU (Remote Terminal Unit) ist ein Standardprotokoll für die Kommunikation im Netzwerk und verwendet die elektrische RS232- oder RS485-Verbindung für die serielle Datenübertragung zwischen Modbus-Geräten im Netzwerk. Eine CPU mit einem RS232 oder RS485 CM oder einem RS485 CB können Sie um PtP-Netzwerkports (Punkt zu Punkt) erweitern.

Modbus RTU nutzt ein Master/Slave-Netzwerk, in dem die gesamte Kommunikation von einem einzigen Master-Gerät ausgelöst wird, während die Slaves lediglich auf die Anforderung des Masters reagieren können. Der Master sendet eine Anforderung an eine Slave-Adresse und nur die Slave-Adresse antwortet auf den Befehl.

Modbus TCP-Kommunikation

Modbus TCP (Transmission Control Protocol) ist ein Standardprotokoll für die Kommunikation im Netzwerk und verwendet den PROFINET-Anschluss an der CPU für die TCP/IP-Kommunikation. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.

Modbus TCP nutzt Open User Communication (OUC)-Verbindungen als Modbus-Kommunikationspfad. Neben der Verbindung zwischen STEP 7 und der CPU kann es mehrere Client/Server-Verbindungen geben. Gemischte Client- und Server-Verbindungen werden bis zur maximalen Anzahl der von der CPU zugelassenen Verbindungen (Seite 595) unterstützt.

Jede MB_SERVER-Verbindung muss eine eindeutige Nummer für Instanz-DB und IP-Port verwenden. Je IP-Port wird nur eine Verbindung unterstützt. Für jede Verbindung muss MB_SERVER (mit eindeutigem Instanz-DB und IP-Port) einzeln ausgeführt werden.

Ein Modbus TCP-Client (Master) muss die Client/Server-Verbindung über den Parameter DISCONNECT steuern. Die grundlegenden Aktionen eines Modbus-Clients werden im Folgenden gezeigt.

1. Verbindung zu einem Server (Slave) mit bestimmter IP-Adresse und IP-Portnummer aufbauen
2. Client-Übertragung einer Modbus-Meldung auslösen und Antworten vom Server empfangen
3. Gegebenenfalls die Verbindungsunterbrechung zwischen Client und Server auslösen, um die Verbindung mit einem anderen Server zu ermöglichen

Modbus RTU-Anweisungen in Ihrem Programm

- **Modbus_Comm_Load:** Eine Ausführung von `Modbus_Comm_Load` ist erforderlich, um PtP-Portparameter wie Baudrate, Parität und Flusskontrolle einzurichten. Nachdem der CPU-Port für das Modbus RTU-Protokoll konfiguriert ist, kann er nur von der Anweisung `Modbus_Master` oder der Anweisung `Modbus_Slave` verwendet werden.
- **Modbus_Master:** Mit der Anweisung `Modbus_Master` kann die CPU als Modbus-RTU-Mastergerät für die Kommunikation mit einem oder mehreren Modbus-Slavegeräten eingesetzt werden.
- **Modbus_Slave:** Mit der Anweisung `Modbus_Slave` kann die CPU als Modbus-RTU-Slavegerät für die Kommunikation mit einem Modbus-Mastergerät eingesetzt werden.

Modbus TCP-Anweisungen in Ihrem Programm

- **MB_CLIENT:** Client/Server-TCP-Verbindung herstellen, Befehlsmeldung senden, Antwort empfangen und Trennen der Verbindung vom Server steuern.
- **MB_SERVER:** Bei Anforderung Verbindung zu einem Modbus-TCP-Client aufbauen, Modbus-Meldung empfangen und Antwort senden.

Siehe auch

Siemens-Website Service & Support (http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

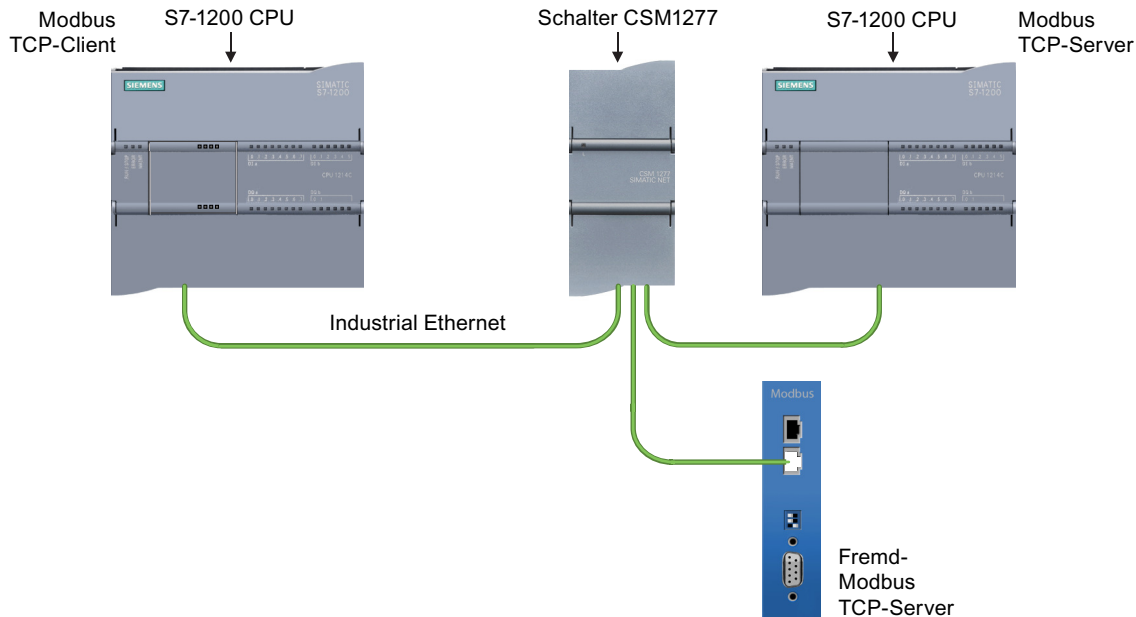
13.5.2 Modbus TCP

13.5.2.1 Übersicht

Ab Version V4.1 der S7-1200 CPU in Verbindung mit STEP 7 V13 SP1 wird die Fähigkeit von Modbus TCP dahingehend erweitert, dass erweiterte T-Bausteinanweisungen verwendet werden können.

Die S7-1200 unterstützt daher zwei Sätze PtP-Anweisungen:

- Ältere Modbus TCP-Anweisungen (Seite 1148): Diese Modbus TCP-Anweisungen waren vor der Version V4.0 der S7-1200 vorhanden.
- Modbus TCP-Anweisungen (Seite 1022): Diese Modbus TCP-Anweisungen stellen alle Funktionen der alten Anweisungen bereit.



13.5.2.2 Version der Modbus TCP-Anweisungen auswählen

Die folgenden Versionen der Modbus TCP-Anweisungen sind in STEP 7 verfügbar:

- Alte Version 2.1: Kompatibel mit allen CPU- und CM-Varianten
- Alte Version 3.1: Kompatibel mit allen CPU- und CM-Varianten
- Version 4.2: Kompatibel mit CPUs ab V4.0 und CMs ab V2.1
- Version 5.1: Kompatibel mit CPUs ab V4.2 und CMs ab V2.1
- Version 6.0: Kompatibel mit CPUs ab V4.2 und CMs ab V2.1

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Rufen Sie in der Taskcard "Anweisung" die MODBUS TCP-Anweisungen in der Gruppe Kommunikationsprozessor unter "Andere" auf.

Um die Version einer Modbus TCP-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Others		
MODBUS TCP		V3.1
MB_CLIENT	Communicate via PROFINET...	V3.1
MB_SERVER	Communicate via PROFINET...	V3.0
Communication processor		V3.1

13.5 Modbus-Kommunikation

Wenn Sie eine Modbus TCP-Anweisung über das Anweisungsverzeichnis in Ihr Programm einfügen, wird in der Projektnavigation eine neue FB-Instanz angelegt. Die neue FB-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer Modbus TCP-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmmeditor angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines Modbus TCP-FBs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der Modbus TCP-Anweisung anzuzeigen.

13.5.2.3 Modbus TCP-Anweisungen

Anweisung MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren)

Tabelle 13-56 Anweisung MB_CLIENT

KOP/FUP	SCL	Beschreibung																																								
<p>*"MB_CLIENT_DB"</p> <div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">MB_CLIENT</p> <table style="width: 100%; border-collapse: collapse;"> <tr><td>EN</td><td>ENO</td></tr> <tr><td>REQ</td><td>DONE</td></tr> <tr><td>DISCONNECT</td><td>BUSY</td></tr> <tr><td>MB_MODE</td><td>ERROR</td></tr> <tr><td>MB_DATA_ADDR</td><td>STATUS</td></tr> <tr><td>MB_DATA_LEN</td><td></td></tr> <tr><td>RD_MB_</td><td></td></tr> <tr><td>DATA_ADDR</td><td></td></tr> <tr><td>RD_MB_</td><td></td></tr> <tr><td>DATA_LEN</td><td></td></tr> <tr><td>WR_MB_</td><td></td></tr> <tr><td>DATA_ADDR</td><td></td></tr> <tr><td>WR_MB_</td><td></td></tr> <tr><td>DATA_LEN</td><td></td></tr> <tr><td>MB_DATA_PTR</td><td></td></tr> <tr><td>CONNECT</td><td></td></tr> <tr><td>RD_MB_</td><td></td></tr> <tr><td>DATA_PTR</td><td></td></tr> <tr><td>WR_MB_</td><td></td></tr> <tr><td>DATA_PTR</td><td></td></tr> </table> </div>	EN	ENO	REQ	DONE	DISCONNECT	BUSY	MB_MODE	ERROR	MB_DATA_ADDR	STATUS	MB_DATA_LEN		RD_MB_		DATA_ADDR		RD_MB_		DATA_LEN		WR_MB_		DATA_ADDR		WR_MB_		DATA_LEN		MB_DATA_PTR		CONNECT		RD_MB_		DATA_PTR		WR_MB_		DATA_PTR		<pre>"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_);</pre>	<p>MB_CLIENT kommuniziert als Modbus TCP-Client über den PROFINET-Port an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich. MB_CLIENT kann eine Client-Server-Verbindung herstellen, eine Modbus-Funktionsanforderung senden, eine Antwort empfangen und das Trennen der Verbindung von einem Modbus TCP-Server steuern.</p>
EN	ENO																																									
REQ	DONE																																									
DISCONNECT	BUSY																																									
MB_MODE	ERROR																																									
MB_DATA_ADDR	STATUS																																									
MB_DATA_LEN																																										
RD_MB_																																										
DATA_ADDR																																										
RD_MB_																																										
DATA_LEN																																										
WR_MB_																																										
DATA_ADDR																																										
WR_MB_																																										
DATA_LEN																																										
MB_DATA_PTR																																										
CONNECT																																										
RD_MB_																																										
DATA_PTR																																										
WR_MB_																																										
DATA_PTR																																										

Tabelle 13-57 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	In	Bool FALSCH = Keine Modbus-Kommunikationsanforderung WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server
DISCONNECT	IN	Bool Mit dem Parameter DISCONNECT kann Ihr Programm den Verbindungsaufbau und -abbau zu einem Modbus-Servergerät steuern. Ist DISCONNECT = 0 und keine Verbindung vorhanden, versucht MB_CLIENT eine Verbindung zur zugewiesenen IP-Adresse und Portnummer aufzubauen. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird versucht, die Verbindung zu trennen. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
MB_MODE	IN	USInt Modusauswahl: Weist die Art der Anforderung (Lesen, Schreiben oder Diagnose) zu. Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_ADDR	IN	UDInt Modbus-Anfangsadresse: Weist die Anfangsadresse der Daten zu, auf die MB_CLIENT zugreifen soll. In der folgenden Tabelle der Modbus-Funktionen finden Sie gültige Adressen.
MB_DATA_LEN	IN	UInt Modbus-Datenlänge: Weist die Anzahl der Bits oder Wörter zu, auf die diese Anforderung zugreifen soll. In der folgenden Tabelle der Modbus-Funktionen finden Sie gültige Längen.
MB_DATA_PTR	IN_OUT	Variant Pointer auf das Modbus-Datenregister: Das Register puffert Daten, die an einen Modbus-Server gesendet oder von dort empfangen werden. Der Pointer muss einen nicht optimierten globalen DB oder eine Adresse im Speicherbereich der Merker zuweisen.
CONNECT	IN_OUT	Variant Verweis auf eine Datenbaueinrichtung, die Verbindungsparameter im Systemdatentyp "TCON_IP_v4" enthält. Die folgenden Datentypen werden ebenfalls unterstützt: TCON_IP_V4_SEC, TCON_QDN und TCON_QDN_SEC. Siehe "Parameter für die PROFINET-Verbindung (Seite 624)".
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> • 0 - Keine Anweisung MB_CLIENT in Bearbeitung • 1 - Anweisung MB_CLIENT in Bearbeitung
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_CLIENT mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung

Modbus-Funktion 23

Beschreibung

Mit der Modbus-Funktion 23 wird in einem Auftrag Folgendes durchgeführt:

1. Daten werden von der CPU zum Modbus-Server übertragen und in ein oder mehrere Haltereister geschrieben.
2. Daten werden aus einem oder mehreren Haltereistern des Modbus-Servers gelesen und in die CPU übertragen.

13.5 Modbus-Kommunikation

Die Anweisung "MB_CLIENT" unterstützt die Modbus-Funktion 23 als Anweisung der Version V6.0.

Parameter

Wenn die Modbus-Funktion 23 verwendet wird, muss der Parameter MB_MODE den Wert 123 haben.

Die Parameter MB_DATA_ADDR, MB_DATA_LEN und MB_DATA_PTR werden nicht verwendet und müssen ihre Standardeinstellungen als Werte haben.

Wenn die Modbus-Funktion 23 verwendet wird, werden sechs neue Parameter verwendet, die in der folgenden Tabelle beschrieben werden. Jeder dieser Parameter beginnt mit "RD_" oder "WR_", um kenntlich zu machen, ob er zur Leseaufgabe (RD_) oder zur Schreibaufgabe (WR_) gehört. Standardmäßig sind diese Parameter ausgeblendet. Bei Verwendung der Modbus-Funktion 23 müssen diese sechs Parameter alle verwendet werden. Wenn Sie eine andere Modbus-Funktion verwenden, müssen diese sechs Parameter den Wert 0 haben oder leer sein. Andernfalls wird der STATUS-Wert 16#818D zurückgegeben.

Tabelle 13-58 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
RD_MB_DATA_ADDR	IN	UInt	Anfangsadresse auf dem Remote-Gerät, ab der Daten gelesen werden sollen. Zulässige Werte: 0 bis 65535
RD_MB_DATA_LEN	IN	UInt	Anzahl der Register, die vom Remote-Gerät gelesen werden sollen. Zulässige Werte: 1 bis 125
WR_MB_DATA_ADDR	IN	UInt	Anfangsadresse auf dem Remote-Gerät, ab der Daten geschrieben werden sollen. Zulässige Werte: 0 bis 65535
WR_MB_DATA_LEN	IN	UInt	Anzahl der Register, die ins Remote-Gerät geschrieben werden sollen. Zulässige Werte: 1 bis 121
RD_MB_DATA_PTR	IN_OUT	Variant	Zeiger auf einen Datenpuffer für die vom Modbus-Server zu lesenden Daten. Als Datentypen sind die gleichen Datentypen wie die für MB_DATA_PTR zulässig.
WR_MB_DATA_PTR	IN_OUT	Variant	Zeiger auf einen Datenpuffer für die in den Modbus-Server zu schreibenden Daten. Als Datentypen sind die gleichen Datentypen wie die für MB_DATA_PTR zulässig.

STATUS-Parameter

Die Bedeutung der STATUS-Werte 16#8383, 8189, 818A, 818B ist erweitert. Der STATUS-Wert 16#818D wird hinzugefügt.

Projekt aktualisieren, Anweisung aktualisieren

Wenn Sie ein vorhandenes Projekt (das z.B. mit TIA Portal V16 erstellt wurde) mit MB_CLIENT-Anweisungen (z.B. Anweisungsversion V5.2) aktualisieren, wird nicht automatisch die neue Version der Anweisung in Ihrem Programm verwendet. Um Modbus-Funktion 23 zu verwenden, müssen Sie die Anweisungsversion manuell aktualisieren.

Parameter REQ

FALSCH = Keine Modbus-Kommunikationsanforderung

WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server

Wenn keine Instanz von MB_CLIENT aktiv ist und der Parameter DISCONNECT = 0 ist, wird bei REQ = 1 eine neue Modbus-Anforderung gestartet. Ist die Verbindung noch nicht hergestellt, wird eine neue Verbindung aufgebaut.

Wird dieselbe Instanz von MB_CLIENT erneut mit DISCONNECT = 0 and REQ = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Modbus-Übertragungen durchgeführt. Sobald die aktuelle Anforderung jedoch abgearbeitet ist, kann eine neue Anforderung verarbeitet werden, sofern MB_CLIENT mit REQ = 1 ausgeführt wird.

Wenn die aktuelle MB_CLIENT-Kommunikationsanforderung abgearbeitet ist, ist das Bit DONE einen Zyklus lang WAHR. Das Bit DONE kann als Zeitfenster für die Sequenzierung mehrerer MB_CLIENT-Anforderungen verwendet werden.

Hinweis

Konsistenz der Eingangsdaten während der Verarbeitung von MB_CLIENT

Nachdem ein Modbus-Client eine Modbus-Operation initiiert, werden alle Eingangszustände intern gespeichert und dann bei jedem nachfolgenden Aufruf verglichen. Der Vergleich dient dazu, festzustellen, ob ein bestimmter Aufruf der ursprüngliche Auslöser der aktiven Client-Anforderung war. Mehrere Aufrufe von MB_CLIENT können mit Hilfe eines gemeinsamen Instanz-DBs durchgeführt werden.

Es ist wichtig, dass die Eingänge während des Zeitraums, in dem eine MB_CLIENT -Operation aktiv verarbeitet wird, nicht geändert werden. Wird diese Regel missachtet, kann MB_CLIENT die aktive Instanz nicht ermitteln.

Über die Parameter MB_MODE und MB_DATA_ADDR wählen Sie die Modbus-Kommunikationsfunktion aus.

Anweisung MB_CLIENT nutzt einen MB_MODE-Eingang statt eines Funktionscodeeingangs. MB_DATA_ADDR weist die Modbus-Anfangsadresse der dezentralen Daten zu.

Die Kombination aus MB_MODE und MB_DATA_ADDR legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MB_MODE, MB_DATA_ADDR und der Modbus-Funktion:

Tabelle 13-59 Modbus-Funktionen

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
0	01	1 bis 2000	Ausgangsbits lesen: 1 bis 2000 Bit pro Anforderung	1 bis 9999
101	01	1 bis 2000	Ausgangsbits lesen: 1 bis 2000 Bit pro Anforderung	00000 bis 65535
0	02	1 bis 2000	Eingangsbits lesen: 1 bis 2000 Bit pro Anforderung	10001 bis 19999
102	02	1 bis 2000	Eingangsbits lesen: 1 bis 2000 Bit pro Anforderung	00000 bis 65535

13.5 Modbus-Kommunikation

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
0	03	1 bis 125	Halteregister lesen: 1 bis 125 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
103	03	1 bis 125	Halteregister lesen: 1 bis 125 Wörter pro Anforderung	00000 bis 65535
0	04	1 bis 125	Eingangswörter lesen: 1 bis 125 Wörter pro Anforderung	30001 bis 39999
104	04	1 bis 125	Eingangswörter lesen: 1 bis 125 Wörter pro Anforderung	00000 bis 65535
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
105	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	00000 bis 65535
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
106	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	00000 bis 65535
1	15	2 bis 1968	Mehrere Ausgangsbits schreiben: 2 bis 1968 Bits pro Anforderung	1 bis 9999
1	16	2 bis 123	Mehrere Halteregister schreiben: 2 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
2	15	1 bis 1968	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bit pro Anforderung	1 bis 9999
115	15	1 bis 1968	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bit pro Anforderung	00000 bis 65535
2	16	1 bis 123	Ein oder mehrere Halteregister schreiben: 1 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
116	16	1 bis 123	Ein oder mehrere Halteregister schreiben: 1 bis 123 Wörter pro Anforderung	00000 bis 65535
11	11	0	Statuswort und Ereigniszähler der Serverkommunikation lesen. Das Statuswort zeigt beschäftigt an (0 = nicht beschäftigt, 0xFFFF = beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert. Sowohl Parameter MB_DATA_ADDR als auch Parameter MB_DATA_LEN von MB_CLIENT wird bei dieser Funktion ignoriert.	
80	08	1	Serverzustand mit Diagnosecode 0x0000 prüfen (Prüf Schleifentest, Server gibt ein Echo der Anforderung zurück) 1 Wort pro Anforderung	
81	08	1	Server-Ereigniszähler mit Diagnosecode 0x000A zurücksetzen 1 Wort pro Anforderung	

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
123	23	1 bis 121 (Schreiben) 1 bis 125 (Lesen)	Schreiben von Halteregeistern im Remote-Gerät und Lesen von Halteregeistern im Remote-Gerät in einem Auftrag. Hinweis: Diese Modbus-Funktion wird von "MB_CLIENT" ab Anweisungsversion V6.0 unterstützt. Die Parameter RD_MB_DATA_ADDR, RD_MB_DATA_LEN, WR_MB_DATA_ADDR, WR_MB_DATA_LEN, RD_MB_DATA_PTR, WR_MB_DATA_PTR werden zu diesem Zweck verwendet.	
3 bis 10, 12 bis 79, 82 bis 100, 107 bis 114, 117 bis 255			Reserviert	

Hinweis**MB_DATA_PTR weist einen Puffer zu, um aus einem Modbus TCP-Server gelesene bzw. in den Server geschriebene Daten zu speichern**

Der Datenpuffer kann sich in einem nicht optimierten globalen DB oder in einer Adresse im Speicherbereich der Merker befinden.

Verwenden Sie für einen Puffer im Speicherbereich der Merker das Format des Any-Pointer. Hierbei handelt es sich um das Format P#"Bitadresse" "Datentyp" "Länge". Ein Beispiel: P#M1000.0 WORD 500.

Zuweisung eines Kommunikationspuffers durch den Parameter MB_DATA_PTR

- Kommunikationsfunktionen von MB_CLIENT:
 - 1-Bit-Daten aus Modbus-Serveradressen lesen und schreiben (00001 bis 09999)
 - 1-Bit-Daten aus Modbus-Serveradressen lesen (10001 bis 19999)
 - 16-Bit-Wortdaten aus Modbus-Serveradressen lesen (30001 bis 39999) und (40001 bis 49999)
 - 16-Bit-Wortdaten in Modbus-Serveradressen schreiben (40001 bis 49999)
- Daten in Wort- oder Bitgröße werden in den oder aus dem von MB_DATA_PTR zugewiesenen Puffer im DB oder Merkerbereich übertragen.
- Wenn von MB_DATA_PTR ein DB als Puffer zugewiesen ist, müssen Sie allen DB-Datenelementen Datentypen zuweisen.
 - Der 1-Bit-Datentyp Bool stellt eine Modbus-Bitadresse dar.
 - Datentypen mit einzelnen 16-Bit-Wörtern wie WORD, UInt und Int stellen eine Modbus-Wortadresse dar.
 - Datentypen mit 32-Bit-Doppelwörtern wie DWORD, DInt und Real stellen zwei Modbus-Wortadressen dar.

13.5 Modbus-Kommunikation

- Komplexe DB-Elemente können von MB_DATA_PTR zugewiesen werden, z.B.
 - Arrays
 - Benannte Strukturen, in denen jedes Element eindeutig ist.
 - Benannte komplexe Strukturen, in denen jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Es ist nicht erforderlich, dass die Datenbereiche für MB_DATA_PTR in demselben globalen Datenbaustein (oder Speicherbereich der Merker) liegen. Sie können einen Datenbaustein für Modbus-Lesevorgänge, einen anderen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jeden MB_CLIENT anlegen.

Parameter CONNECT weist Daten zum Herstellen einer PROFINET-Verbindung zu

Sie müssen einen globalen Datenbaustein verwenden und die erforderlichen Verbindungsdaten speichern, bevor Sie diesen DB am Parameter CONNECT referenzieren können.

1. Um die CONNECT-Daten zu speichern, erstellen Sie einen neuen globalen DB oder verwenden einen vorhandenen globalen DB. In einem DB können Sie mehrere TCON_IP_v4-Datenstrukturen speichern. Jede Modbus TCP-Client- bzw. -Server-Verbindung verwendet eine TCON_IP_v4-Datenstruktur. Am Parameter CONNECT verweisen Sie auf die Verbindungsdaten.
2. Geben Sie dem DB und einer statischen Variable aussagekräftige Namen. Nennen Sie den Datenbaustein beispielsweise "Modbus-Verbindungen" und die statische Variable "TCPaktiv_1" (für die Modbus TCP-Clientverbindung 1).
3. Im DB-Editor weisen Sie der statischen Variable des Beispiels "TCPaktiv_1" den Systemdatentyp "TCON_IP_v4" in der Spalte "Datentyp" zu.
4. Erweitern Sie die Struktur TCON_IP_v4, damit Sie die Verbindungsparameter ändern können, wie in der folgenden Abbildung gezeigt.
5. Ändern Sie die Daten in der Struktur TCON_IP_v4 für eine MB_CLIENT-Verbindung.
6. Geben Sie den Verweis auf die DB-Struktur für den Parameter CONNECT von MB_CLIENT ein. In diesem Beispiel lautet der Verweis "Modbus-Verbindungen".TCPaktiv_1.

Modbus connections				
	Name	Datentyp	Startwert	Kommentar
1	Static			
2	TCPactive_1	TCON_IP_v4		
3	Interfaceld	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connection: 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

Ändern der TCP_IP_V4-DB-Daten für jede MB_CLIENT-Verbindung

- **InterfaceID:** Klicken Sie im Fenster der Gerätekonfiguration auf das Bild des PROFINET-Anschlusses der CPU. Klicken Sie dann auf das Register "Allgemein" und verwenden Sie die dort angezeigte Hardwarekennung.
- **ID:** Geben Sie eine Verbindungs-ID zwischen 1 und 4095 ein. Die Modbus TCP-Kommunikation erfolgt über die zugrunde liegenden Anweisungen TCON, TDISCON, TSEND und TRCV für die offene Benutzerkommunikation.
- **ConnectionType:** Verwenden Sie bei TCP/IP den Standard 16#0B (Dezimalzahl = 11).
- **ActiveEstablished:** Dieser Wert muss 1 oder WAHR sein. Die Verbindung ist deshalb aktiv, weil MB_CLIENT die Modbus-Kommunikation initiiert.
- **RemoteAddress:** Geben Sie die IP-Adresse des Modbus TCP-Zielservers in die vier ADDR-Array-Elemente ein. Geben Sie beispielsweise 192.168.2.241 wie im Bild oben gezeigt ein.
- **RemotePort:** Der Standardwert ist 502. Diese Nummer ist die IP-Portnummer des Modbus-Servers, mit dem MB_CLIENT eine Verbindung aufzubauen und zu kommunizieren versucht. Bei den Modbus-Servern einiger Dritthersteller ist es erforderlich, dass Sie eine andere Portnummer verwenden.
- **LocalPort:** Dieser Wert muss bei einer MB_CLIENT-Verbindung 0 sein.

Mehrere Clientverbindungen

Ein Modbus TCP-Client unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communication-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und -Server, darf die maximale Anzahl der unterstützten Open User Communication-Verbindungen nicht überschreiten.

Einzelne gleichzeitige Clientverbindungen müssen die folgenden Regeln einhalten:

- Jede MB_CLIENT-Verbindung muss einen eindeutigen Instanz-DB nutzen.
- Jede MB_CLIENT-Verbindung muss eine eindeutige IP-Adresse eines Servers zuweisen.
- Jede MB_CLIENT-Verbindung muss eine eindeutige Verbindungs-ID zuweisen.
- Eindeutige IP-Portnummern sind möglicherweise je nach Serverkonfiguration erforderlich.

13.5 Modbus-Kommunikation

Für jeden Instanz-DB muss eine andere Verbindungs-ID verwendet werden. Zusammengefasst heißt dies, dass Instanz-DB und Verbindungs-ID gepaart sind und für jede Verbindung eindeutig sein müssen.

Tabelle 13-60 Instanz-Datenbaustein von MB_CLIENT: für den Benutzer zugängliche statische Variablen

Variable	Datentyp	Voreinstellung	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), die auf eine blockierte Modbus-Client-Instanz gewartet werden soll, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Clientanforderung ausgegeben wurde und die Anwendung dann aufhört, die Clientfunktion auszuführen, bevor die Anforderung vollständig abgearbeitet wurde. Maximaler Grenzwert bei der S7-1200 sind 55 Sekunden.
MB_Unit_ID	Wort	255	Modbus-Geräteerkennung: Ein Modbus TCP-Server wird über seine IP-Adresse angesprochen. Deshalb wird der Parameter MB_UNIT_ID bei der Modbus TCP-Adressierung nicht verwendet. Der Parameter MB_UNIT_ID entspricht dem Feld der Slaveadresse beim Modbus RTU-Protokoll. Wenn ein Modbus TCP-Server als Gateway zu einem Modbus RTU-Protokoll verwendet wird, kann das Slavegerät im seriellen Netzwerk über MB_UNIT_ID identifiziert werden. Der Parameter MB_UNIT_ID würde in diesem Fall die Anforderung an die richtige Modbus RTU-Slaveadresse weiterleiten. Einige Modbus TCP-Geräte benötigen den Parameter MB_UNIT_ID möglicherweise, um innerhalb eines eingeschränkten Wertebereichs zu liegen.
RCV_TIMEOUT	Real	2,0	Zeit in Sekunden, die MB_CLIENT auf die Antwort eines Servers auf eine Anforderung wartet.
Verbunden	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Server verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden

Tabelle 13-61 MB_CLIENT-Protokollfehler

STATUS* (W#16#)	Lokale und/oder entfernte Fehler	Fehlercode in der Antwort von MB_SERVER (B#16#)	Beschreibung
80C8	Lokal	-	Keine Antwort des Servers im festgelegten Zeitraum. Prüfen Sie die Verbindung zum Modbus-Server. Dieser Fehler wird erst bei Abschluss der konfigurierten Versuchswiederholungen gemeldet. Wenn die Anweisung "MB_CLIENT" im festgelegten Zeitraum keine Antwort mit der ursprünglich übertragenen Transaktions-ID erhält (siehe statische Variable MB_TRANSACTION_ID), wird dieser Fehlercode ausgegeben.
8380	Lokal	-	Empfangenes Modbus-Telegramm hat falsches Format oder es wurden zu wenige Bytes empfangen.
8381	Entfernt	01	Funktionscode wird nicht unterstützt.

STATUS* (W#16#)	Lokale und/oder entfernte Fehler	Fehlercode in der Antwort von MB_SERVER (B#16#)	Beschreibung
8382	Lokal	-	<ul style="list-style-type: none"> Die Länge des Modbus-Telegramms im Telegramm-Header entspricht nicht der Anzahl der empfangenen Bytes. Die Anzahl der Bytes entspricht nicht der Anzahl der tatsächlich übertragenen Bytes (nur Funktionen 1-4). Beispiel: Das ist der Fall, wenn "MB_CLIENT" eine ungerade Anzahl von Wörtern anfordert, doch "MB_SERVER" immer eine gerade Anzahl von Wörtern sendet. Die Anfangsadresse im empfangenen Telegramm entspricht nicht der gespeicherten Anfangsadresse (Funktionen 5, 6, 15, 16). Die Anzahl der Wörter entspricht nicht der Anzahl der tatsächlich übertragenen Wörter (Funktionen 15 und 16).
	Entfernt	03	Ungültige Längenangabe im empfangenen Modbus-Telegramm. Serverseite prüfen.
8383	Lokal	-	<ul style="list-style-type: none"> Anweisungsversion < V6.0: Fehler beim Lesen oder Schreiben von Daten oder Zugriff außerhalb des Adressbereichs von MB_DATA_PTR. Anweisungsversion ≥ V6.0: Fehler beim Lesen oder Schreiben von Daten oder Zugriff außerhalb des Adressbereichs von MB_DATA_PTR, RD_MB_DATA_PTR oder WR_MB_DATA_PTR.
	Entfernt	02	Fehler beim Lesen oder Schreiben von Daten oder Zugriff außerhalb des Adressbereichs des Servers.
8384	Lokal	-	<ul style="list-style-type: none"> Ungültiger Ausnahmecode empfangen. Es wurde ein anderer Datenwert empfangen als ursprünglich vom Client gesendet wurde (Funktionen 5, 6 und 8). Ungültiger Statuswert empfangen (Funktion 11)
	Entfernt	03	Fehler im Datenwert für Funktion 5
8385	Lokal	-	<ul style="list-style-type: none"> Diagnosecode nicht unterstützt. Es wurde ein anderer Teilfunktionscode empfangen als ursprünglich vom Client gesendet wurde (Funktion 8).
	Entfernt	03	Diagnosecode nicht unterstützt.
8386	Lokal	-	Empfangener Funktionscode entspricht nicht dem ursprünglich gesendeten.
8387	Lokal	-	Die Protokoll-ID des vom Server empfangenen Modbus TCP-Telegramms ist nicht "0".
8388	Lokal	-	Der Modbus-Server hat eine andere Datenlänge gesendet als angefordert wurde. Dieser Fehler tritt nur bei Verwendung der Modbus-Funktionen 5, 6, 15 oder 16 auf.
* Die Statuscodes können als Ganzzahl- oder Hexadezimalwerte im Programmeditor angezeigt werden. Für Informationen zum Wechseln der Anzeigeformate, siehe "Siehe auch".			

Tabelle 13-62 Bedingungs-codes der Ausführung von MB_CLIENT ¹

STATUS (W#16#)	MB_CLIENT-Parameterfehler
7001	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Dieser Code wird nur für die erste Ausführung eines Verbindungsaufbaus oder -abbaus ausgegeben.
7002	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Diese wird für alle nachfolgenden Ausführungen ausgegeben, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.
7003	Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
80C8	Der Server hat nicht während der festgelegten Zeit reagiert. MB_CLIENT muss innerhalb der zugewiesenen Zeit über die ursprünglich gesendete Transaktions-ID eine Antwort erhalten oder es wird dieser Fehler ausgegeben. Prüfen Sie die Verbindung zum Modbus-Servergerät. Dieser Fehler wird erst ausgegeben, nachdem Wiederholungen (sofern zutreffend) durchgeführt wurden.
8188	Parameter MB_MODE hat einen ungültigen Wert.
8189	<ul style="list-style-type: none"> Anweisungsversion < V6.0: Ungültige Adressierung von Daten am Parameter MB_DATA_ADDR. Anweisungsversion ≥ V6.0: Ungültige Adressierung von Daten am Parameter MB_DATA_ADDR, RD_MB_DATA_ADDR oder WR_MB_DATA_ADDR.
818A	<ul style="list-style-type: none"> Anweisungsversion < V6.0: Ungültige Datenlänge am Parameter MB_DATA_LEN. Anweisungsversion ≥ V6.0: Ungültige Datenlänge am Parameter MB_DATA_LEN, RD_MB_DATA_LEN oder WR_MB_DATA_LEN.
818B	Ungültiger Pointer auf den DATA_PTR-Bereich. Hierbei kann es sich um die Kombination aus MB_DATA_ADDRESS + MB_DATA_LEN handeln.
818C	Pointer DATA_PTR zeigt auf einen nicht optimierten DB-Bereich (hierbei muss es sich um einen nicht optimierten DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln)
818D	Ein oder mehrere Parameter haben nicht ihren Standardwert, werden jedoch mit der angegebenen Modbus-Funktion nicht verwendet. Beispiel: Wenn MB_MODE den Wert 123 hat, müssen MB_DATA_ADDR und MB_DATA_LEN den Wert 0 haben und MB_DATA_PTR muss leer sein. Wenn MB_MODE einen anderen Wert als 123 hat, müssen alle Parameter, die mit "RD_" oder "WR_" beginnen, den Wert 0 haben oder leer sein.
8200	Der Port ist durch die Verarbeitung einer vorhandenen Modbus-Anforderung belegt.
8380	Der empfangene Modbus-Rahmen ist inkorrekt oder es wurden zu wenige Bytes empfangen.
8387	Der zugewiesene Parameter der Verbindungs-ID unterscheidet sich von der für vorherige Anforderungen verwendeten ID. Es darf in jedem Instanz-DB von MB_CLIENT nur eine einzige Verbindungs-ID verwendet werden. Dieser Code wird auch als interner Fehler ausgegeben, wenn die von einem Server empfangene Modbus TCP-Protokoll-ID nicht 0 ist.
8388	Ein Modbus-Server hat eine Datenmenge zurückgegeben, die sich von der angeforderten Menge unterscheidet. Dieser Code gilt nur für die Modbus-Funktionen 15 und 16.

¹ Neben den oben aufgeführten MB_CLIENT-Fehlern können auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV) Fehler gemeldet werden.

Siehe auch

Asynchrone Kommunikationsverbindungen (Seite 595)

Anweisung MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren)

Tabelle 13-63 Anweisung MB_SERVER

KOP/FUP	SCL	Beschreibung										
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">"MB_SERVER_DB"</p> <div style="border: 1px solid black; padding: 2px; margin: 5px 0;"> <p style="text-align: center;">MB_SERVER</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid black;">EN</td> <td style="width: 50%;">ENO</td> </tr> <tr> <td style="border-right: 1px solid black;">DISCONNECT</td> <td>NDR</td> </tr> <tr> <td style="border-right: 1px solid black;">CONNECT</td> <td>DR</td> </tr> <tr> <td style="border-right: 1px solid black;">MB_HOLD_REG</td> <td>ERROR</td> </tr> <tr> <td></td> <td>STATUS</td> </tr> </table> </div> </div>	EN	ENO	DISCONNECT	NDR	CONNECT	DR	MB_HOLD_REG	ERROR		STATUS	<pre>"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_);</pre>	<p>MB_SERVER kommuniziert als Modbus TCP-Server über den PROFINET-Port an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.</p> <p>MB_SERVER kann eine Anforderung für den Verbindungsaufbau mit einem Modbus TCP-Client annehmen, eine Modbus-Funktionsanforderung empfangen und eine Antwortmeldung senden.</p>
EN	ENO											
DISCONNECT	NDR											
CONNECT	DR											
MB_HOLD_REG	ERROR											
	STATUS											

Tabelle 13-64 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
DISCONNECT	IN	Bool	MB_SERVER versucht, eine "passive" Verbindung mit einem Partnergerät aufzubauen. Das bedeutet, dass der Server passiv überwacht, ob es TCP-Verbindungsanforderungen von IP-Adressen gibt, die Anforderungen senden. Ist DISCONNECT = 0 und keine Verbindung vorhanden, kann eine passive Verbindung aufgebaut werden. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird versucht, die Verbindung zu trennen. Über diesen Parameter kann Ihr Programm steuern, wann eine Verbindung angenommen wird. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
CONNECT	IN	Variant	Verweis auf eine Datenbaueinstuktur, die Verbindungsparameter im Systemdatentyp "TCON_IP_v4" enthält. Die folgenden Datentypen werden ebenfalls unterstützt: TCON_IP_V4, TCON_QDN und TCON_QDN_SEC. Siehe "Parameter für die PROFINET-Verbindung (Seite 624)".
MB_HOLD_REG	IN_OUT	Variant	Pointer auf das Modbus-Halteregister von MB_SERVER: Bei dem Halteregister muss es sich um einen nicht optimierten globalen DB oder um eine Adresse im Speicherbereich der Merker handeln. Dieser Speicherbereich dient dazu, die Daten zu speichern, auf die ein Modbus-Client mit den Modbus-Registerfunktionen 3 (Lesen), 6 (Schreiben), 16 (Schreiben) und 23 (Lesen/Schreiben) zugreifen kann.
NDR	OUT	Bool	Neue Daten bereit: 0 = Keine neuen Daten, 1 = Gibt an, dass von einem Modbus-Client neue Daten geschrieben wurden
DR	OUT	Bool	Daten lesen: 0 = Daten nicht gelesen, 1 = Gibt an, dass die Daten von einem Modbus-Client gelesen wurden
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_SERVER mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

Hinweis

Vorausgesetzte Firmwareversion der CPU

Für die in diesem Abschnitt des Handbuchs beschriebenen Modbus TCP-Anweisungen ist die Firmwareversion V4.1 oder höher erforderlich.

Parameter CONNECT weist Daten zum Herstellen einer PROFINET-Verbindung zu

Sie müssen einen globalen Datenbaustein verwenden und die erforderlichen Verbindungsdaten speichern, bevor Sie diesen DB am Parameter CONNECT referenzieren können.

1. Um die CONNECT-Daten zu speichern, erstellen Sie einen neuen globalen DB oder verwenden einen vorhandenen globalen DB. In einem DB können Sie mehrere TCON_IP_v4-Datenstrukturen speichern. Jede Modbus TCP-Client- bzw. -Server-Verbindung verwendet eine TCON_IP_v4-Datenstruktur. Am Parameter CONNECT verweisen Sie auf die Verbindungsdaten.
2. Geben Sie dem DB und einer statischen Variable aussagekräftige Namen. Nennen Sie den Datenbaustein beispielsweise "Modbus-Verbindungen" und die statische Variable "TCPpassiv_1" (für die Modbus TCP-Serververbindung 1).
3. Im DB-Editor weisen Sie der statischen Variable des Beispiels "TCPaktiv_1" den Systemdatentyp "TCON_IP_v4" in der Spalte "Datentyp" zu.
4. Erweitern Sie die Struktur TCON_IP_v4, damit Sie die Verbindungsparameter ändern können, wie in der folgenden Abbildung gezeigt.
5. Ändern Sie die Daten in der Struktur TCON_IP_v4 für eine MB_SERVER-Verbindung.
6. Geben Sie den Verweis auf die DB-Struktur für den Parameter CONNECT von MB_SERVER ein. In diesem Beispiel lautet der Verweis "Modbus-Verbindungen".TCPpassiv_1.

	Name	Datentyp	Startwert	Kommentar
1	Static			
2	TCPpassive_1	TCON_IP_v4		
3	Interfaceld	HW_ANY	64	HW-identifier of IE-interface submodule
4	ID	CONN_OUC	1	connection reference / identifier
5	ConnectionType	Byte	16#0B	type of connction: 11=TCPIP, 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

Ändern der TCP_IP_V4-DB-Daten für jede MB_SERVER-Verbindung

- **InterfaceID:** Klicken Sie im Fenster der Gerätekonfiguration auf das Bild des PROFINET-Anschlusses der CPU. Klicken Sie dann auf das Register "Allgemein" und verwenden Sie die dort angezeigte Hardwarekennung.
- **ID:** Geben Sie eine Zahl zwischen 1 und 4095 ein, die für diese Verbindung eindeutig ist. Die Modbus TCP-Kommunikation erfolgt über die zugrunde liegenden Anweisungen TCON, TDISCON, TSEND und TRCV für die offene Benutzerkommunikation. Bis zu acht Verbindungen der offenen Benutzerkommunikation sind gleichzeitig zulässig.
- **ConnectionType:** Verwenden Sie bei TCP/IP den Standard 16#0B (Dezimalwert = 11).
- **ActiveEstablished:** Dieser Wert muss 0 oder FALSCH sein. Die Verbindung ist deshalb passiv, weil MB_SERVER auf eine Kommunikationsanforderung von einem Modbus-Client wartet.
- **RemoteAddress:** Es gibt zwei Optionen.
 - Verwenden Sie 0.0.0.0, und MB_CLIENT reagiert auf eine Modbus-Anforderung von jedem TCP-Client.
 - Geben Sie die IP-Adresse eines Ziel-Modbus-TCP-Clients ein, und MB_CLIENT reagiert nur auf eine Anforderung von der IP-Adresse dieses Clients. Geben Sie beispielsweise 192.168.2.241 wie im Bild oben gezeigt ein.
- **RemotePort:** Dieser Wert muss bei einer MB_SERVER-Verbindung 0 sein.
- **LocalPort:** Der Standardwert ist 502. Diese Nummer ist die IP-Portnummer des Modbus-Clients, mit dem MB_SERVER eine Verbindung aufzubauen und zu kommunizieren versucht. Die Modbus-Clients einiger Dritthersteller benötigen eine andere Portnummer.

Modbus- und Prozessabbildadressen

Bei MB_SERVER können eingehende Modbus-Funktionscodes (1, 2, 4, 5 und 15) Bits und Wörter direkt im Prozessabbild der Eingänge und Ausgänge lesen und schreiben. Bei den Funktionscodes für die Datenübertragung (3, 6 und 16) muss der Parameter MB_HOLD_REG als Datentyp größer als ein Byte definiert sein. Die folgende Tabelle zeigt die Zuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 13-65 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	An	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	An	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Funktionscodes eingehender Modbus-Meldungen (3, 6 und 16) lesen oder schreiben Wörter in einem Modbus-Halteregister, bei dem es sich um einen Merker oder um einen Datenbaustein handeln kann. Der Typ des Halteregisters wird vom Parameter MB_HOLD_REG angegeben.

Hinweis

Zuweisung des Parameters MB_HOLD_REG

Modbus-Halteregister, die als Arrays aus Wörtern, Ganzzahlen (16 Bit), breiten Zeichen, vorzeichenlosen Ganzzahlen (16 Bit), Bytes, kurzen Ganzzahlen, vorzeichenlosen kurzen Ganzzahlen, Zeichen, Doppelwörtern, Ganzzahlen (32 Bit), vorzeichenlosen Ganzzahlen (32 Bit) oder Realzahlen definiert sind, können in jedem Speicherbereich abgelegt werden.

Modbus-Halteregister, die Sie als Strukturen definiert haben, müssen in nicht-optimierten DBs abgelegt werden.

Für ein Modbus-Halteregister im Speicherbereich der Merker verwenden Sie das Format des Any-Pointers. Dies ist das Format P#"Bitadresse" "Datentyp" "Länge". Beispiel: P#M1000.0 WORD 500.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 13-66 Beispiele für die Zuordnung von Modbus-Adressen zu Adressen im Speicher der CPU

Modbus-Adresse	Beispiele für Parameter von MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	"Rezept".Inhaltsstoff[5]

Header des Modbus-Anwendungsprotokolls

Der Header des Modbus-Anwendungsprotokolls umfasst die ersten sieben Bytes jeder Modbus TCP-Nachricht. Dieser Header enthält die Transaktionskennung, die Protokollkennung, die Länge und die Geräteerkennung. Die Antwortnachricht der MB_SERVER-Anweisung enthält die gleichen Werte für die Transaktionskennung, die Protokollkennung und die Geräteerkennung, die in der Modbus-Anforderungsnachricht empfangen wurden. Das Längenfeld wird von der Anweisung MB_SERVER berechnet.

Mehrere Serververbindungen

Es können mehrere Serververbindungen hergestellt werden. Ein einziges PLC-Gerät kann gleichzeitig Verbindungen mit mehreren Modbus TCP-Clients herstellen.

Ein Modbus TCP-Server unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communication-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und Modbus TCP-Server,

darf die maximale Anzahl der unterstützten Open User Communication-Verbindungen nicht überschreiten (Seite 595). Die Modbus TCP-Verbindungen können von Client- und Serververbindungen gemeinsam genutzt werden.

Einzelne gleichzeitige Serververbindungen müssen die folgenden Regeln einhalten:

- Jede MB_SERVER-Verbindung muss einen eindeutigen Instanz-DB nutzen.
- Jede MB_SERVER-Verbindung muss eine eindeutige IP-Portnummer zuweisen. Je Port wird nur eine Verbindung unterstützt.
- Jede MB_SERVER-Verbindung muss eine eindeutige Verbindungs-ID zuweisen.
- MB_SERVER muss für jede Verbindung einzeln aufgerufen werden (mit dem entsprechenden Instanz-DB).

Die Verbindungs-ID muss für jede einzelne Verbindung eindeutig sein. Für jeden einzelnen Instanz-DB muss eine einzige Verbindungs-ID verwendet werden. Der Instanz-DB und die Verbindungs-ID sind gepaart und müssen für jede Verbindung eindeutig sein.

Tabelle 13-67 Funktionscodes der Modbus-Diagnose

Modbus-Diagnosefunktionen von MB_SERVER		
Codes	Teilfunktion	Beschreibung
08	0x0000	Echotest Abfragedaten ausgeben: Die Anweisung MB_SERVER gibt einem Modbus-Client das Echo eines empfangenen Datenworts zurück.
08	0x000A	Kommunikationseigniszähler löschen: Die Anweisung MB_SERVER löscht den Kommunikationseigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationseigniszähler abrufen: Die Anweisung MB_SERVER nutzt einen internen Kommunikationseigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und Modbus-Schreibanforderungen, die an den Modbus-Server gesendet werden, zu erfassen. Der Zähler wird bei Funktion 8, Funktion 11 und allen Anforderungen, die zu einem Kommunikationsfehler führen, nicht hochgezählt. Die Broadcast-Funktion ist bei Modbus TCP nicht verfügbar, weil nur eine Client-Server-Verbindung zur Zeit vorhanden ist.

Variablen im Datenbaustein (DB) der Anweisung MB_SERVER

Diese Tabelle zeigt die öffentlichen statischen Variablen im Instanz-Datenbaustein von MB_SERVER, die in Ihrem Programm verwendet werden können.

Tabelle 13-68 Öffentliche statische Variablen von MB_SERVER

Variable	Datentyp	Voreinstellung	Beschreibung
HR_Start_Offset	Word	0	Weist die Anfangsadresse des Modbus-Halteregisters zu
Request_Count	Word	0	Die Anzahl aller von diesem Server empfangenen Anforderungen
Server_Message_Count	Word	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen
Xmt_Rcv_Count	Word	0	Die Anzahl der Sendungen oder Datenempfänge, bei denen ein Fehler aufgetreten ist. Wird auch inkrementiert, wenn eine Meldung empfangen wird, bei der es sich um eine ungültige Modbus-Meldung handelt.
Exception_Count	Word	0	Modbus-spezifische Fehler, die eine zurückgegebene Ausnahme erfordern

13.5 Modbus-Kommunikation

Variable	Datentyp	Voreinstellung	Beschreibung
Success_Count	Word	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen ohne Protokollfehler.
Connected	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Client verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden
QB_Start	UInt	0	Startadresse der Ausgangsbytes, in die die CPU schreiben kann (AB0 bis AB65535)
QB_Count	UInt	65535	Anzahl Bytes, in die ein entferntes Gerät schreiben kann. Ist QB_Count = 0, kann ein entferntes Gerät nicht in die Ausgänge schreiben. Beispiel: Wenn nur AB10 bis AB17 beschreibbar sein sollen, muss QB_Start = 10 und QB_Count = 8 sein.
QB_Read_Start	UInt	0	Startadresse der Ausgangsbytes, aus denen die CPU lesen kann (AB0 bis AB65535)
QB_Read_Count	UInt	65535	Anzahl der Ausgangsbytes, aus denen ein entferntes Gerät lesen kann. Ist QB_Count = 0, kann ein entferntes Gerät nicht aus den Ausgängen lesen. Beispiel: Wenn nur AB10 bis AB17 lesbar sein sollen, muss QB_Start = 10 und QB_Count = 8 sein.
IB_Read_Start	UInt	0	Startadresse der Eingangsbytes, aus denen die CPU lesen kann (EB0 bis EB65535)
IB_Read_Count	UInt	65535	Anzahl der Eingangsbytes, aus denen ein entferntes Gerät lesen kann. Ist IB_Count = 0, kann ein entferntes Gerät nicht aus den Eingängen lesen. Beispiel: Wenn nur EB10 bis EB17 lesbar sein sollen, muss IB_Start = 10 und IB_Count = 8 sein.
NDR_immediate	Bool	FALSCH	Gleiche Bedeutung wie Parameter NDR (neue Daten bereit). Der MB_SERVER aktualisiert "NDR_immediate" mit dem gleichen Aufruf, der eine Modbus TCP Schreibenanforderung verarbeitet.
DR_immediate	Bool	FALSCH	Gleiche Bedeutung wie Parameter DR (Daten lesen). Der MB_SERVER aktualisiert "DR_immediate" mit dem gleichen Aufruf, der eine Modbus TCP Schreibenanforderung verarbeitet.

Ihr Programm kann Daten in die steuernden Modbus-Server-Operationen und die folgenden Variablen schreiben:

- HR_Start_Offset
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- IB_Read_Start
- IB_Read_Count

Es gelten folgende Versionsanforderungen für die Verfügbarkeit der Variablen im Datenbaustein (DB) der Anweisung MB_SERVER

Tabelle 13-69 Versionsanforderungen für die Verfügbarkeit der Variablen im Datenbaustein (DB) der Anweisung MB_SERVER: Anweisung, TIA Portal und S7-1200 CPU

Version der MB_SERVER Anweisung	Version von TIA Portal	Version der S7-1200 CPU Firmware (FW)	Datenbausteinvariablen
4.2	V14 SP1	CPU FW V4.0 oder höher	QB_Start
			QB_Count
5.0 oder höher	V15 oder höher	CPU FW V4.2 oder höher	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count
			NDR_immediate
DR_immediate			

HR_Start_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR_Start_Offset verwenden, um eine andere Anfangsadresse als 40001 für das Modbus-Halteregister zu konfigurieren.

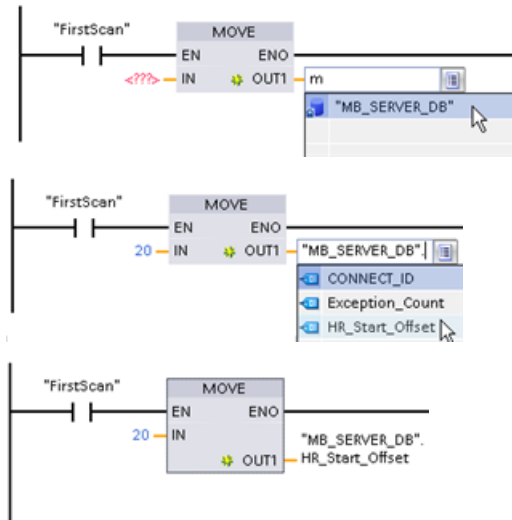
Sie können z. B. ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 40119 führt zu einem Adressierungsfehler.

Tabelle 13-70 Beispiel für die Adressierung des Modbus-Halteregisters

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR_Start_Offset sind Wortdaten im Instanz-Datenbaustein von MB_SERVER, die die Anfangsadresse des Modbus-Halteregisters zuweisen. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie MB_SERVER in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise MB_SERVER in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und HR_Start_Offset zuweisen. Die Anfangsadresse muss vor der Ausführung von MB_SERVER zugewiesen werden.



Eingabe einer Variable für den Modbus-Server über den Namen des Standard-DB:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste der DB-Namen "MB_SERVER_DB" aus.
3. Wählen Sie in der Klappliste der DB-Variablen "MB_SERVER_DB.HR_Start_Offset" aus.

Zugriff auf Datenbereiche in Datenbausteinen (DB) statt des direkten Zugriffs auf Modbus-Adressen

Ab der Version V5.0 der MB_SERVER -Anweisung und Firmware-Version V4.2 der S7-1200 CPU kann auf Datenbereiche in DBs zugegriffen werden, statt direkt auf Prozessabbilder und Haltereister zuzugreifen. Hierfür müssen auf der Eigenschaftenseite "Attribute" der globalen DB die Kontrollkästchen "Nur in Ladespeicher ablegen" und "Optimierter Bausteinzugriff" ausgewählt werden.

Wenn eine Modbus-Anforderung eintrifft und kein Datenbereich für den Modbus-Datentyp des entsprechenden Funktionscodes festgelegt wurde, behandelt die Anweisung MB_SERVER die Anforderung wie in früheren Anweisungsversionen: Sie können direkt auf Prozessabbilder und Haltereister zugreifen.

Wenn ein Datenbereich für den Modbus-Datentyp des Funktionscodes festgelegt wurde, wird von der MB_SERVER -Anweisung aus diesem Datenbereich gelesen bzw. in diesen Datenbereich geschrieben. Ob gelesen oder geschrieben wird, hängt von der Art des Auftrags ab.

Hinweis

Wenn ein Datenbereich konfiguriert ist, ignoriert die MB_SERVER -Anweisung die von den statischen Variablen in dem Instanz-Datenbaustein, der dem Datentyp des Datenbereichs entspricht, konfigurierten Offsets oder Bereiche. Diese Offsets und Bereiche gelten nur für das Prozessabbild oder den Speicher, auf das/den MB_HOLD_REG verweist. Die Start- und Längenparameter des Datenbereichs bieten eine eigene Art der Definition von Offsets und Bereichen.

Bei einer einzelnen Modbus-Anforderung kann nur in einen Datenbereich geschrieben bzw. daraus gelesen werden. Sollen beispielsweise Haltereister gelesen werden, die sich über mehrere Datenbereiche erstrecken, sind mehrere Modbus-Anforderungen notwendig.

Für die Festlegung der Datenbereiche gelten folgende Regeln:

- Sie können bis zu acht Datenbereiche in verschiedenen DBs festlegen, wobei jeder DB nur einen Datenbereich enthalten darf. Eine einzelne MODBUS-Anforderung kann nur aus genau einem Datenbereich lesen oder in genau einen Datenbereich schreiben. Jeder Datenbereich entspricht einem MODBUS-Adressbereich. Die Datenbereiche legen Sie in der statischen Variablen "Data_Area_Array" des Instanz-DB fest.
- Sollen weniger als acht Datenbereiche verwendet werden, müssen die benötigten Datenbereiche lückenlos nacheinander angeordnet werden. Beim ersten leeren Eintrag in den Datenbereichen wird die Datenbereichssuche bei der Verarbeitung abgebrochen. Werden zum Beispiel die Feldelemente 1, 2, 4 und 5 als Datenbereiche festgelegt, erkennt "Data_Area_Array" nur die Feldelemente 1 und 2, da Feldelement 3 leer ist.

13.5 Modbus-Kommunikation

- Feld Data_Area_Array besteht aus acht Elementen: Data_Area_Array[1] bis Data_Area_Array[8]
- Jedes Feldelement Data_Area_Array[x], $1 \leq x \leq 8$ ist ein UDT des Typs MB_DataArea und wie folgt strukturiert:

Parameter	Datentyp	Bedeutung
data_type	UInt	<p>Kennung für den MODBUS-Datentyp, der auf diesen Datenbereich abgebildet ist:</p> <ul style="list-style-type: none"> • 0: Kennung für ein leeres Feldelement oder einen unbenutzten Datenbereich. Die Werte für db, start und length sind dann ohne Bedeutung. • 1: Prozessabblausgang (in Verbindung mit Funktionscodes 1, 5 und 15) • 2: Prozessabbildeingang (in Verbindung mit Funktionscode 2) • 3: Halteregeister (in Verbindung mit Funktionscodes 3, 6 und 16) • 4: Eingangsregister (in Verbindung mit Funktionscode 4) <p>Hinweis: Wenn Sie einen Datenbereich für einen MODBUS-Datentyp festgelegt haben, kann die Anweisung MB_SERVER nicht mehr direkt auf diesen MODBUS-Datentyp zugreifen. Wenn die Adresse einer MODBUS-Anforderung für einen solchen Datentyp nicht einem festgelegten Datenbereich entspricht, wird in STATUS der Wert W#16#8383 zurückgegeben.</p>
db	UInt	<p>Nummer des Datenbausteins, auf dem das MODBUS-Register oder die anschließend festgelegten Bits abgebildet sind</p> <p>Die DB-Nummer muss in den Datenbereichen einmalig vorhanden sein. Die DB-Nummer darf nicht in mehreren Datenbereichen vorhanden sein.</p> <p>Auf der Eigenschaftenseite "Attribute" der globalen DB müssen die Kontrollkästchen "Nur in Ladespeicher ablegen" und "Optimierter Bausteinzugriff" abgewählt werden.</p> <p>Die Datenbereiche beginnen auch mit der Byte-Adresse 0 des DB.</p> <p>Zulässige Werte: 1 bis 60999</p>
start	UInt	<p>Erste MODBUS-Adresse, die auf den Datenbaustein abgebildet ist, beginnend mit Adresse 0.0</p> <p>Zulässige Werte: 0 bis 65535</p>
length	UInt	<p>Anzahl Bits (für die Werte 1 und 2 von data_type) oder Anzahl Register (für die Werte 3 und 4 von data_type)</p> <p>Die MODBUS-Adressbereiche eines MODBUS-Datentyps dürfen sich nicht überlappen.</p> <p>Zulässige Werte: 1 bis 65535</p>

Beispiele für die Festlegung von Datenbereichen:

- Erstes Beispiel: data_type = 3, db = 1, start = 10, length = 6
Die CPU bildet die Haltereister (data_type = 3) in Datenbaustein 1 (db = 1) ab und legt die Modbus-Adresse 10 (start = 10) auf Datenwort 0 und die letzte gültige Modbus-Adresse 15 (length = 6) auf Datenwort 5.
- Zweites Beispiel: data_type = 2, db = 15, start = 1700, length = 112
Die CPU bildet die Eingänge (data_type = 2) in Datenbaustein 15 (db = 15) ab und legt die Modbus-Adresse 1700 (start = 1700) auf Datenwort 0 und die letzte gültige Modbus-Adresse 1811 (length = 112) auf Datenwort 111.

Bedingungscode

Tabelle 13-71 Bedingungscode der Ausführung von MB_SERVER ¹

STATUS (W#16#)	Antwortcode an Modbus-Server(B#16#)	Modbus-Protokollfehler
7001		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird bei der ersten Ausführung eines Verbindungsaufbaus oder -abbaus ausgegeben.
7002		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird für alle nachfolgenden Ausführungen ausgegeben, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.
7003		Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
8187		MB_HOLD_REG ist nicht gültig, zeigt möglicherweise in einen optimierten DB oder auf einen Bereich, der kleiner als 2 Byte ist.
818C		Pointer MB_HOLD_REG zeigt auf einen nicht optimierten DB-Bereich (hierbei muss es sich um einen nicht optimierten globalen DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln) oder Timeout für gesperrten Prozess überschreitet den Grenzwert von 55 Sekunden. (spezifisch für die S7-1200)
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in Datenlänge: <ul style="list-style-type: none"> • Ungültige Längenangabe im empfangenen Modbus-Telegramm. • Die Telegrammlänge im Header des Modbus-Telegramms entspricht nicht der Anzahl der tatsächlich empfangenen Bytes. • Die Anzahl der Bytes im Header des Modbus-Telegramms entspricht nicht der Anzahl der tatsächlich empfangenen Bytes (Funktionen 15 und 16).
8383	02	Datenadressfehler oder Zugriff außerhalb der Grenzen des Adressbereichs von MB_HOLD_REG
8384	03	Fehler im Datenwert

13.5 Modbus-Kommunikation

STATUS (W#16#)	Antwortcode an Modbus-Server(B#16#)	Modbus-Protokollfehler
8385	03	Datendiagnosecode wird nicht unterstützt (Funktionscode 08)
8389		Ungültige Datenbereichsdefinition: <ul style="list-style-type: none"> • Ungültiger Wert für den Datentyp • DB-Nummer ist ungültig oder nicht vorhanden: <ul style="list-style-type: none"> – Ungültiger DB-Wert – DB-Nummer ist nicht vorhanden – DB-Nummer wird bereits von einem anderen Datenbereich verwendet – DB mit optimiertem Zugriff – DB befindet sich nicht im Arbeitsspeicher • Ungültiger Längenwert • Überschneidung von MODBUS-Adressbereichen, die zu demselben MODBUS-Datentyp gehören

¹ Neben den oben aufgeführten MB_SERVER-Fehlern können Fehler auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV) gemeldet werden.

MB_RED_CLIENT (redundante Kommunikation über PROFINET als Modbus TCP-Client)

Sie können diese Anweisung verwenden, um eine Verbindung zwischen einer S7-1200 CPU und einem Gerät, das das Modbus TCP-Protokoll unterstützt, herzustellen.

Tabelle 13-72 Anweisung MB_RED_CLIENT

KOP/FUP	SCL	Beschreibung																																							
<div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center; color: cyan;">%DB4</p> <p style="text-align: center;">"MB_RED_CLIENT_DB_1"</p> <p style="text-align: center; background-color: #e0e0e0; padding: 2px;">MB_RED_CLIENT </p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— EN</td> <td style="padding: 2px;">ENO</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— REG_KEY</td> <td style="padding: 2px;">LICENSED</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— USE_ALL_CONN</td> <td style="padding: 2px;">IDENT_CODE</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— REQ</td> <td style="padding: 2px;">DONE</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— DISCONNECT</td> <td style="padding: 2px;">BUSY</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— MB_MODE</td> <td style="padding: 2px;">ERROR</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— MB_DATA_ADDR</td> <td style="padding: 2px;">STATUS_0A</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— MB_DATA_LEN</td> <td style="padding: 2px;">STATUS_1A</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;">— MB_DATA_PTR</td> <td style="padding: 2px;">STATUS_0B</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;"></td> <td style="padding: 2px;">STATUS_1B</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;"></td> <td style="padding: 2px;">RED_ERR_S7</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;"></td> <td style="padding: 2px;">RED_ERR_DEV</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> <tr> <td style="border-right: 1px solid gray; padding: 2px;"></td> <td style="padding: 2px;">TOT_COM_ERR</td> <td style="border-right: 1px solid gray; padding: 2px;">—</td> </tr> </table> </div>	— EN	ENO	—	— REG_KEY	LICENSED	—	— USE_ALL_CONN	IDENT_CODE	—	— REQ	DONE	—	— DISCONNECT	BUSY	—	— MB_MODE	ERROR	—	— MB_DATA_ADDR	STATUS_0A	—	— MB_DATA_LEN	STATUS_1A	—	— MB_DATA_PTR	STATUS_0B	—		STATUS_1B	—		RED_ERR_S7	—		RED_ERR_DEV	—		TOT_COM_ERR	—	<pre> "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_); </pre>	<p>Die Anweisung MB_RED_CLIENT kommuniziert als Modbus TCP-Client über die PROFINET-Verbindung.</p> <p>Mit der Anweisung MB_RED_CLIENT können Sie eine redundante Verbindung zwischen dem Client und dem Server herstellen, Modbus-Anforderungen senden, Antworten empfangen und das Beenden der Verbindung durch den Modbus TCP-Client steuern.</p>
— EN	ENO	—																																							
— REG_KEY	LICENSED	—																																							
— USE_ALL_CONN	IDENT_CODE	—																																							
— REQ	DONE	—																																							
— DISCONNECT	BUSY	—																																							
— MB_MODE	ERROR	—																																							
— MB_DATA_ADDR	STATUS_0A	—																																							
— MB_DATA_LEN	STATUS_1A	—																																							
— MB_DATA_PTR	STATUS_0B	—																																							
	STATUS_1B	—																																							
	RED_ERR_S7	—																																							
	RED_ERR_DEV	—																																							
	TOT_COM_ERR	—																																							

Tabelle 13-73 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REG_KEY ¹	IN	STRING[17] Registrierungscode für die Lizenzierung Die Anweisung MB_RED_CLIENT muss auf jeder CPU einzeln lizenziert werden.
USE_ALL_CONN	IN	Bool Geben Sie die Anzahl der konfigurierten Verbindungen an, über die das Telegramm gesendet werden soll: <ul style="list-style-type: none"> • 0: Telegramm über eine einzige Verbindung senden, nur im Fehlerfall zur nächsten Verbindung schalten • 1: Telegramm über alle konfigurierten Verbindungen senden
REQ	IN	Bool Modbus-Abfrage an den Modbus TCP-Server Der Parameter REQ ist stufengesteuert. Das bedeutet, dass die Anweisung, solange der Eingang gesetzt ist (REQ = WAHR), Kommunikationsanforderungen sendet. Wenn die Verbindung noch nicht aufgebaut ist, wird sie jetzt aufgebaut und das Modbus-Telegramm sofort danach gesendet. Änderungen an den Eingangsparametern werden erst wirksam, wenn der Server reagiert hat oder eine Fehlermeldung ausgegeben wurde. Wenn der Parameter REQ während einer laufenden Modbus-Anforderung erneut gesetzt wird, findet anschließend keine weitere Übertragung statt.
DISCONNECT	IN	Bool Mit diesem Parameter steuern Sie das Herstellen und Beenden der Verbindung zum Modbus-Server: <ul style="list-style-type: none"> • 0: Herstellen der Kommunikationsverbindung zu dem Verbindungspartner, der im Parameter CONNECT konfiguriert ist (siehe Parameter CONNECT). • 1: Trennen der Kommunikationsverbindung. Beim Beenden der Verbindung wird keine andere Funktion ausgeführt. Nach dem erfolgreichen Beenden der Verbindung wird am Parameter STATUS_x der Wert 0003 ausgegeben.
MB_MODE ²	IN	USInt Wählt den Modus der Modbus-Anforderung (Lesen, Schreiben oder Diagnose) oder direkte Auswahl einer Modbus-Funktion
MB_DATA_ADDR ²	IN	UDInt Modbus-Adresse abhängig von MB_MODE
MB_DATA_LEN	IN	UInt Datenlänge: Anzahl von Bits oder Registern für den Datenzugriff
MB_DATA_PTR ²	IN_OUT	Variant Pointer auf einen Datenpuffer für die vom Modbus-Server zu empfangenden Daten oder die an den Modbus-Server zu sendenden Daten.
LICENSED ¹	OUT	Bool <ul style="list-style-type: none"> • 0: Anweisung ist nicht lizenziert • 1: Anweisung ist lizenziert
IDENT_CODE ¹	OUT	STRING[18] Identifikation für die Lizenzierung. Verwenden Sie diese Zeichenfolge, um den Registrierungscode REG_KEY anzufordern.
DONE	OUT	Bool Das Bit am Ausgangsparameter DONE wird auf "1" gesetzt, sobald der aktivierte Modbus-Auftrag an mindestens einer Verbindung fehlerfrei abgeschlossen wurde.
BUSY	OUT	Bool <ul style="list-style-type: none"> • 0: Keine Modbus-Anforderung in Bearbeitung • 1: Modbus-Anforderung wird verarbeitet Der Ausgangsparameter BUSY ist beim Verbindungsaufbau und beim Beenden der Verbindung nicht gesetzt.

Parameter und Datentyp		Datentyp	Beschreibung
ERROR	OUT	Bool	<ul style="list-style-type: none"> 0: Kein Fehler 1: Der aktivierte Modbus-Auftrag konnte an keiner der konfigurierten Verbindungen erfolgreich übertragen werden. Die Fehlerursache wird vom Parameter STATUS_x angegeben.
STATUS_0A ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 0A.
STATUS_1A ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 1A.
STATUS_0B ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 0B.
STATUS_1B ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 1B.
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> 0: Kein Redundanzfehler in SIMATIC 1: Redundanzfehler in SIMATIC
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> 0: Kein Redundanzfehler auf der Seite des Linkpartners 1: Redundanzfehler auf der Seite des Linkpartners
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> 0: Mindestens 1 konfigurierte Verbindung ist hergestellt 1: Vollständiger Kommunikationsverlust, alle konfigurierten Verbindungen sind beendet

¹ Weitere Informationen finden Sie im nachfolgenden Abschnitt "Lizenzierung".

² Weitere Informationen finden Sie im nachfolgenden Abschnitt "Eingangsparameter: MB_MODE, MB_DATA_ADDR, MB_DATA_LEN und MB_DATA_PTR".

³ Weitere Informationen finden Sie im nachfolgenden Abschnitt "Ausgangsparameter: STATUS_x, RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR".

Hinweis

Konsistente Eingangsdaten während eines Aufrufs von MB_RED_CLIENT

Wenn eine Modbus-Client-Anweisung aufgerufen wird, werden die Werte der Eingangsparameter intern gespeichert. Während das Telegramm verarbeitet wird, dürfen die Werte nicht geändert werden.

Hinweis

Vorausgesetzte Firmwareversion der CPU

Für die in diesem Abschnitt des Handbuchs beschriebenen Modbus TCP-Anweisungen ist die Firmwareversion V4.2 oder höher erforderlich.

Für die Verwendung der Anweisung wird kein zusätzliches Hardwaremodul benötigt.

Mehrere Clientverbindungen

Die CPUs können mehrere Modbus TCP-Client-Verbindungen verarbeiten. Die maximale Anzahl von Verbindungen hängt von der verwendeten CPU ab und ist in den technischen Daten der CPU zu finden. Die Gesamtzahl von Verbindungen einer CPU, einschließlich der von Modbus TCP-Clients und -Servern darf die maximale Anzahl der unterstützten Verbindungen nicht überschreiten.

Bei einzelnen Client-Verbindungen beachten Sie die folgenden Regeln:

- Jede MB_RED_CLIENT -Verbindung muss einen eindeutigen Instanz-DB verwenden.
- Bei jeder MB_RED_CLIENT -Verbindung muss eine eindeutige Server-IP-Adresse angegeben werden.
- Jede MB_RED_CLIENT -Verbindung benötigt eine eindeutige Verbindungs-ID. Die Verbindung muss CPU-weit eindeutig sein.

Betrieb und Redundanz

Die Kommunikationsteilnehmer können als unabhängige oder redundante Teilnehmer ausgeführt werden. Wenn einer der Partner unabhängig ausgeführt ist, wird dies als einseitige Redundanz bezeichnet. Wenn beide Partner unabhängig ausgeführt sind, wird dies zweiseitige Redundanz genannt:

- Einseitige Redundanz:
 - Beschreibung: Für jede Verbindung zwischen den Kommunikationspartnern muss jeweils eine Verbindung konfiguriert werden. Die Verbindungspunkte der **SIMATIC S7** werden als **0** und **1** bezeichnet. Die Verbindungspunkte der **Kommunikationspartner** werden als **A** und **B** bezeichnet. Bei der R-CPU oder H-CPU bezieht sich 1 auf den Verbindungspunkt 0, und R-CPU oder H-CPU 2 bezieht sich auf den Verbindungspunkt 1.
 - Konfiguration: Wenn die S7 redundant ausgeführt ist, wird eine Verbindung vom S7-Verbindungspunkt 0 zu Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **A** => Verbindung **0A**), und es wird eine Verbindung vom S7-Verbindungspunkt 1 zum Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **A** => Verbindung **1A**). Die Abbildung zeigt die Verbindungsbezeichnungen:

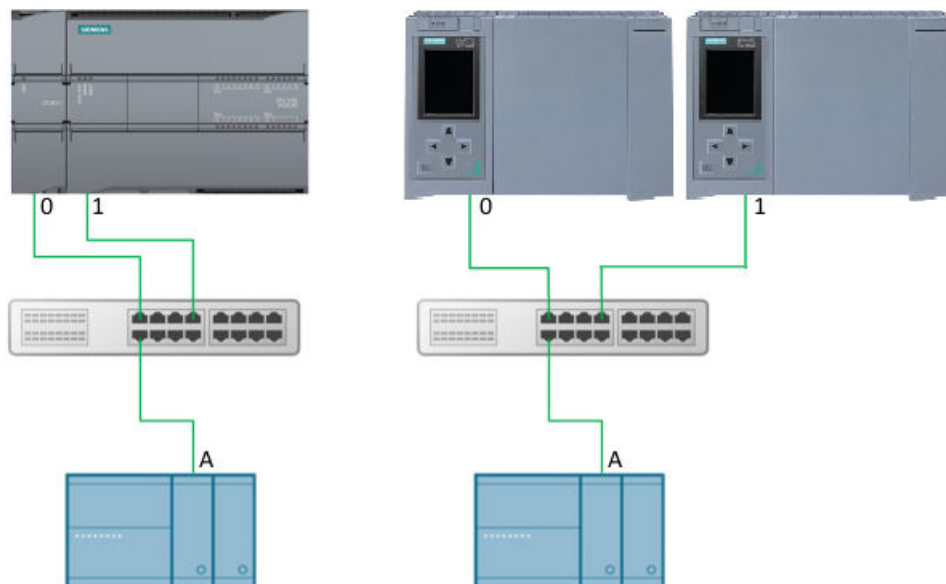


Bild 13-1 Einseitige Redundanz S7

- Wenn die S7 unabhängig und der Linkpartner redundant ausgeführt ist, wird eine Verbindung vom S7-Verbindungspunkt 0 zu Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **A** => Verbindung **0A**), und es wird eine Verbindung vom S7-Verbindungspunkt 0 zum Knotenpunkt B des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **B** => Verbindung **0B**). Die Abbildung zeigt die Verbindungsbezeichnungen:

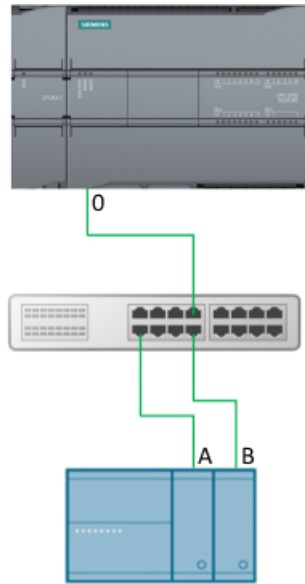


Bild 13-2 Einseitige Redundanz Partner

- Zweiseitige Redundanz:
 - Beschreibung: Für jede Verbindung zwischen den Kommunikationspartnern muss jeweils eine Verbindung konfiguriert werden. Die Verbindungspunkte der **SIMATIC S7** werden als 0 und 1 bezeichnet. Die Verbindungspunkte der **Kommunikationspartner** werden als **A** und **B** bezeichnet.
Bei der R-CPU oder H-CPU bezieht sich 1 auf den Verbindungspunkt 0, und R-CPU oder H-CPU 2 bezieht sich auf den Verbindungspunkt 1.

13.5 Modbus-Kommunikation

- Konfiguration: Bei zweiseitiger Redundanz werden vom Verbindungspunkt 0 zwei Verbindungen erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **A** => Verbindung **0A** und Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **B** => Verbindung **0B**), und es werden zwei Verbindungen vom Verbindungspunkt 1 der S7 zu den Knotenpunkten A und B des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **A** => Verbindung **1A** und Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **B** => Verbindung **1B**). Die Abbildung zeigt die Verbindungsbezeichnungen:

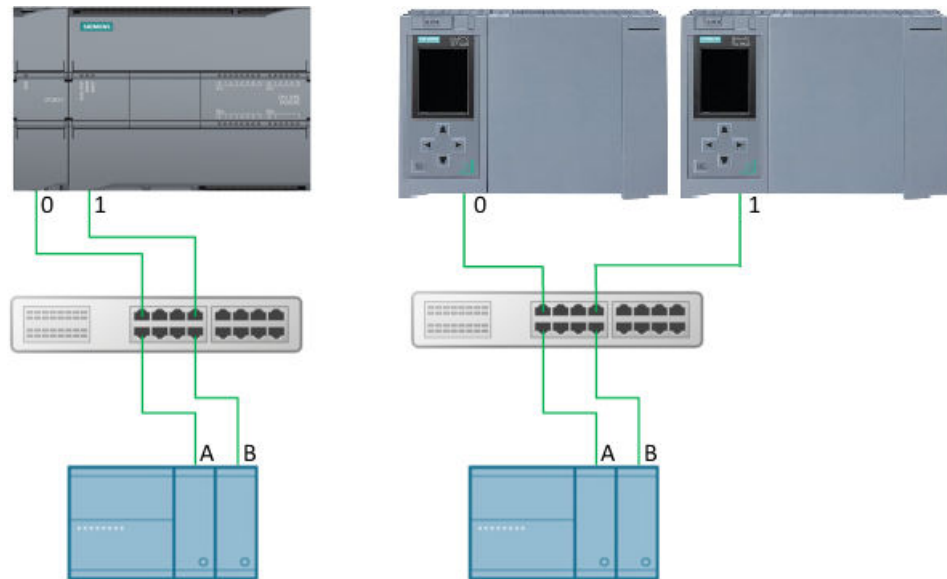


Bild 13-3 Zweiseitige Redundanz:

- Telegrammverarbeitung: Die Telegramme können über eine oder über alle konfigurierten Verbindungen gesendet werden:
 - Telegramme über eine Verbindung senden: Das MODBUS-Telegramm wird mit der Einstellung `USE_ALL_CONN = FALSE` über eine - die derzeit aktive - Verbindung gesendet. Bei einer Zeitüberschreitung (keine Antwort vom Server) oder bei einem Verbindungsfehler wird versucht, das Telegramm über die anderen (maximal 4) konfigurierten Verbindungen zu senden. Die Reihenfolge lautet dann 0A, 1A, 0B und 1B. Wenn ein Telegramm erfolgreich über eine der Verbindungen übertragen wurde, wird diese Verbindung als "aktiv" gekennzeichnet und der weitere Telegrammverkehr wird über diese Verbindung ausgeführt. Bei einem Verbindungsfehler der aktiven Verbindung wird erneut auf allen konfigurierten Verbindungen versucht, das Telegramm zu senden. Wenn alle Sendeversuche fehlschlagen, werden `ERROR` und `STATUS_x` entsprechend gesetzt.
Wenn ein Antworttelegramm empfangen wird, wird eine Plausibilitätsprüfung durchgeführt. Ist diese Prüfung erfolgreich, werden die erforderlichen Aktionen durchgeführt und der Auftrag fehlerfrei ausgeführt. Der Ausgang `DONE` wird gesetzt. Werden während der Prüfung Fehler erkannt, wird der Auftrag mit Fehlern beendet, das Bit `ERROR` wird gesetzt und an `STATUS_x` wird eine Fehlernummer angezeigt. In diesem Fall wird kein neuer Versuch unternommen, das Telegramm auf der nächsten konfigurierten Verbindung zu senden. Ein Umschalten zu den anderen konfigurierten Verbindungen findet nur statt, wenn ein Verbindungsfehler erkannt oder keine Antwort erhalten wurde.
 - Telegramme über alle Verbindungen senden: Das MODBUS-Telegramm wird mit der Einstellung `USE_ALL_CONN = TRUE` über alle konfigurierten, hergestellten Verbindungen gesendet. Nach dem Empfang des Antworttelegramms an einer der Verbindungen wird eine Gültigkeitsprüfung durchgeführt. Ist diese Prüfung erfolgreich, werden die erforderlichen Aktionen durchgeführt. Wurde an mindestens einer Verbindung ein gültiges Antworttelegramm empfangen, wird der Ausgang `DONE` gesetzt.
- Redundanzausgänge `RED_ERR_S7`, `RED_ERR_DEV` und `TOT_COM_ERR`:

13.5 Modbus-Kommunikation

- Die Redundanzbits RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR werden abhängig von den Zuständen der Statusausgänge gesetzt:

Anzahl der fehlerhaften Verbindungen	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
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

Bild 13-4 Anzeige der Alarmbits für Redundanzeinrichtung auf beiden Seiten

Anzahl der fehlerhaften Verbindungen	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

Bild 13-5 Anzeige der Alarmbits für Redundanzeinrichtung auf einer Seite

Hinweis

Portnummern für Client und Server

Der Modbus-Client verwendet eine Portnummer mit Beginn an 2000. Der Modbus-Server wird üblicherweise über die Portnummer 502 angesprochen.

Parametrierung

Für die S7-1200 können Sie die Anweisung **MB_RED_CLIENT V1.0** und **V1.1** verwenden. Die CPU implementiert die Verbindungen über die lokale Schnittstelle der CPU oder des CM/CP. Über die Struktur **TCON_IP_V4** konfiguriert die CPU die Verbindungen und baut sie auf.

Konfiguration von **MB_RED_CLIENT**: Sie nehmen die folgenden Einstellungen im Konfigurationsdialog der Anweisung **MB_RED_CLIENT** vor:

- Verbindungsparameter für die Verbindungen 0A, 1A, 0B und 1B (weitere Informationen zur Redundanzkonfiguration finden Sie unter "Betrieb und Redundanz" oben)
- Interne Parameter (optional)

Sie können den Konfigurationsdialog mit der Anweisung **MB_RED_CLIENT** oder über die Technologieobjekte öffnen:

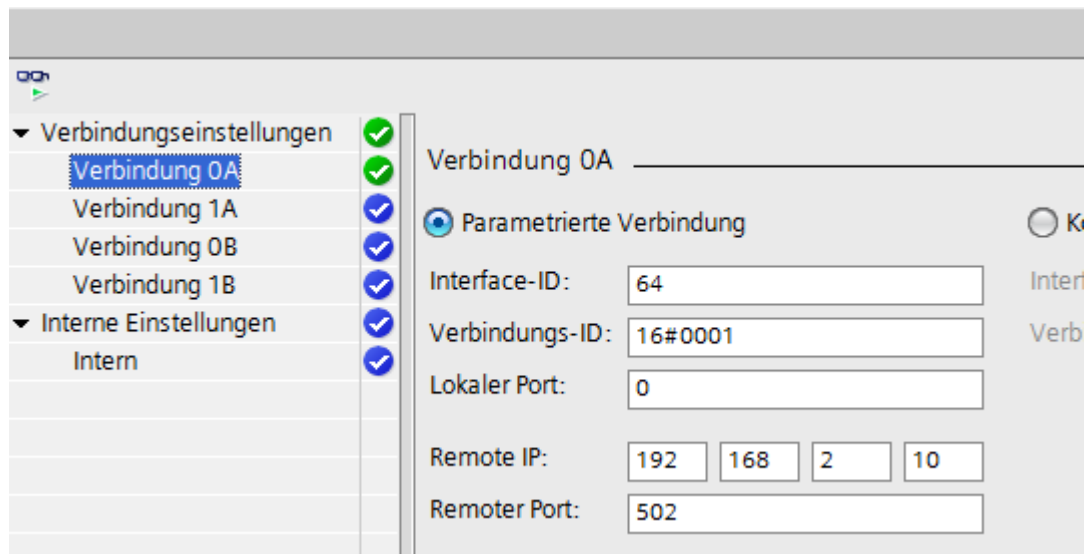
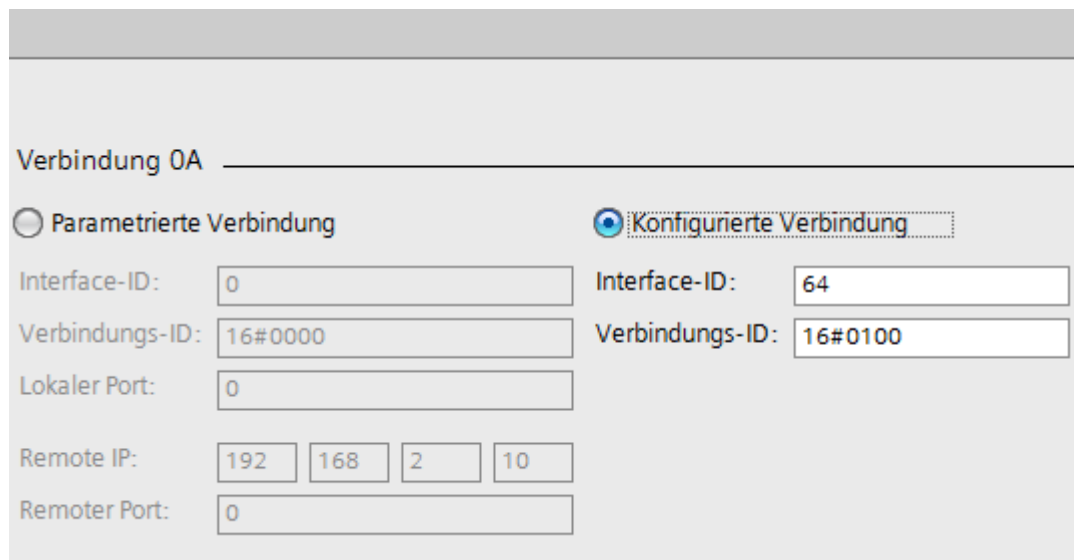


Bild 13-6 Parametrierte Client-Verbindung



13.5 Modbus-Kommunikation

Bild 13-7 Konfigurierte Client-Verbindung

Variable	Startwert	Beschreibung
Konfigurierte Verbindungen		
Interface ID	64	HW-Kennung der verwendeten PN-Schnittstelle
Connection ID	16#0000	Verbindungs-IDs für die verwendeten Verbindungen Diese Verbindungs-IDs müssen CPU-weit eindeutig sein.
Local port	0	Lokale Portnummer des Clients. Standardmäßig ist für den Client keine Portnummer eingegeben.
Remote IP	0.0.0.0	Entfernte IP-Adresse des Servers
Remote port	502	Entfernte Portnummer des Servers Der Standardport für den Modbus/TCP-Server ist 502.
Konfigurierte Verbindungen		
Interface ID	64	HW-Kennung der verwendeten PN-Schnittstelle
Connection ID	16#0000	Verbindungs-IDs für die verwendeten Verbindungen Diese Verbindungen werden in der Netzsicht konfiguriert.

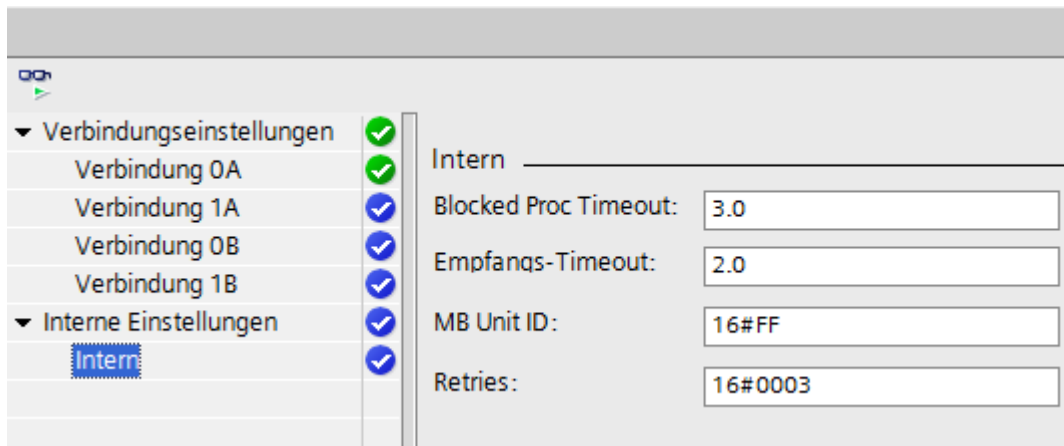


Bild 13-8 Interne Parameter (optional)

Variable	Datentyp	Startwert	Beschreibung
Blocked Proc Time	REAL	3.0	Wartezeit in Sekunden, bevor die statische Variable ACTIVE zurückgesetzt wird, wenn eine Modbus-Instanz blockiert ist. Dies kann beispielsweise auftreten, wenn eine Client-Anforderung ausgegeben wird und die Ausführung der Client-Funktion abbricht, bevor die Anforderung vollständig ausgeführt wurde. Die Wartezeit muss zwischen 0,5 s und 55 s betragen.
Receive timeout	REAL	2.0	Zeitintervall in Sekunden, in dem die Anweisung "MB_RED_CLIENT" auf eine Antwort vom Server wartet. Das Intervall muss zwischen 0,5 s und 55 s betragen.

Variable	Datentyp	Startwert	Beschreibung
MB_Unit_ID	BYTE	255	<p>Modbus-Geräteerkennung: Ein Modbus TCP-Server wird über seine IP-Adresse angesprochen. Aus diesem Grund wird der Parameter MB_UNIT_ID bei der Modbus TCP-Adressierung nicht verwendet.</p> <p>Der Parameter MB_UNIT_ID entspricht dem Feld der Slave-Adresse beim Modbus RTU-Protokoll. Wenn ein Modbus/TCP-Server als Gateway für ein Modbus RTU-Protokoll verwendet wird, kann das Slave-Gerät im seriellen Netzwerk über MB_UNIT_ID identifiziert werden. Der Parameter MB_UNIT_ID leitet in diesem Fall die Anforderung an die richtige Modbus RTU-Slave-Adresse weiter.</p> <p>Beachten Sie bitte, dass einige Modbus/TCP-Geräte für die Initialisierung innerhalb eines begrenzten Wertebereichs möglicherweise den Parameter MB_UNIT_ID benötigen.</p>
Retries	WORD	3	Anzahl der Sendeversuche der Anweisung MB_RED_CLIENT, bevor sie den Fehler W#16#80C8 zurückgibt.

Hinweis**Variable MB_Transaction_ID**

Wenn die Transaktions-ID in der Antwort des Modbus TCP-Servers nicht der Transaktions-ID des Auftrags von MB_RED_CLIENT entspricht, wartet die Anweisung MB_RED_CLIENT während des Zeitraums RCV_TIMEOUT * RETRIES auf die Antwort des Modbus TCP-Servers mit der richtigen Transaktions-ID. Sobald diese Zeit abgelaufen ist, wird der Fehler W#16#80C8 zurückgegeben.

Lizenzierung

Die Anweisung MB_RED_CLIENT unterliegt einer Gebühr und Sie müssen die Anweisung für jede CPU einzeln lizenzieren. Die Lizenzierung läuft in zwei Schritten ab:

- Anzeigen des IDENT_CODE der Lizenz
- Eingeben des Registrierungsschlüssels REG_KEY: Sie müssen den Registrierungsschlüssel REG_KEY bei jeder Anweisung MB_RED_CLIENT eingeben. Speichern Sie den REG_KEY in einem globalen Datenbaustein, aus dem alle MB_RED_CLIENT-Anweisungen den erforderlichen Registrierungsschlüssel erhalten.

Vorgehensweise zum Anzeigen des IDENT_CODE der Lizenz:

1. Parametrieren Sie die Anweisung MB_RED_CLIENT entsprechend Ihren Anforderungen in einem zyklischen OB. Laden Sie das Programm in die CPU und versetzen Sie die CPU in RUN.
2. Öffnen Sie den Instanz-DB der Modbus-Anweisung und klicken Sie auf die Schaltfläche "Alle beobachten".

3. Der Instanz-DB zeigt am Ausgang IDENT_CODE eine 18-stellige Zeichenfolge an.

	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

Bild 13-9 Lizenz

- Kopieren Sie diese Zeichenfolge mit der Funktion Kopieren/Einfügen aus dem Datenbaustein und fügen Sie sie in das Formular ein (das Sie nach der Bestellung des Produkts per E-Mail erhalten haben oder sich auch auf der CD befindet).
- Senden Sie dieses Formular über eine Serviceanfrage an Kundensupport (<https://support.industry.siemens.com/my/ww/de/requests/#createRequest>). Daraufhin erhalten Sie den Registrierungsschlüssel für Ihre CPU.

Vorgehensweise zum Eingeben des Registrierungsschlüssels REG_KEY:

- Fügen Sie über "Neuen Baustein hinzufügen..." einen neuen globalen Datenbaustein mit einem eindeutigen symbolischen Namen, z.B. "Lizenz-DB", ein.
- Erstellen Sie in diesem Baustein einen Parameter REG_KEY mit dem Datentyp STRING[17].

Lizenz_DB				
	Name	Datentyp	Offset	Startwert
1	Static			
2	REG_KEY	String[17]	0.0	"

Bild 13-10 REG KEY

- Kopieren Sie den übertragenen 17-stelligen Registrierungsschlüssel mit der Funktion Kopieren/Einfügen in die Spalte "Startwert".
- Geben Sie im zyklischen OB am Parameter REG_KEY der Anweisung MB_RED_CLIENT den Namen des Lizenz-DB und den Namen der Zeichenfolge ein (z.B. Lizenz_DB.REG_KEY).
- Geänderte Bausteine in die CPU laden Sie können den Registrierungsschlüssel während der Laufzeit eingeben. Es ist nicht nötig, die CPU von STOP in RUN zu versetzen.
- Die Modbus/TCP-Kommunikation über die Anweisung MB_RED_CLIENT ist jetzt für diese CPU lizenziert, das Ausgangsbit LICENSED ist TRUE.

Vorgehensweise zum Korrigieren von fehlender oder fehlerhafter Lizenzierung:

- Wenn Sie einen fehlerhaften Registrierungsschlüssel oder gar keinen Registrierungsschlüssel eingeben, blinkt die ERROR LED der CPU. Außerdem nimmt die CPU bei der S7-1200 auch noch einen zyklischen Eintrag in den Diagnosepuffer bezüglich der fehlenden Lizenz vor.

Diagnosepuffer

Ereignisse

CPU-Zeitstempel berücksichtigt lokale PG/PC-Zeit

Nr.	Datum und Uhrzeit	Ereignis
1	20.12.2018 12:30:33.459	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
2	20.12.2018 12:30:33.458	Kein gültiger License Key für das Funktionspaket
3	20.12.2018 12:30:27.453	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
4	20.12.2018 12:30:27.452	Kein gültiger License Key für das Funktionspaket
5	20.12.2018 12:30:21.445	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
6	20.12.2018 12:30:21.444	Kein gültiger License Key für das Funktionspaket
7	20.12.2018 12:30:15.439	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
8	20.12.2018 12:30:15.438	Kein gültiger License Key für das Funktionspaket
9	20.12.2018 12:30:09.431	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB

Anzeige einfrieren

Bild 13-11 Diagnosepuffer

- Bei einem fehlenden oder fehlerhaften Registrierungsschlüssel verarbeitet die CPU die Modbus TCP-Kommunikation. Die CPU zeigt am Ausgang STATUS_x jedoch immer "W#16#0A90" (Kein gültiger License Key für Funktionspaket) an. Das Ausgangsbit LICENSED ist FALSE.

Eingangsparameter: MB_MODE, MB_DATA_ADDR, MB_DATA_LEN und MB_DATA_PTR

Die Kombination aus MB_MODE, MB_DATA_ADDR und MB_DATA_LEN legt den Funktionscode fest, der in der aktuellen Modbus-Meldung verwendet wird:

- **MB_MODE** enthält die Information, ob gelesen oder geschrieben werden soll:
Lesen: MB_MODE = 0, 101, 102, 103 und 104
Schreiben: MB_MODE = 1, 2, 105, 106, 115 und 116 (Hinweis: Wenn MB_MODE = 2, gibt es keine Unterscheidung zwischen den Modbus-Funktionen 15 und 05 bzw. zwischen den Modbus-Funktionen 16 und 06.)
- **MB_DATA_ADDR** enthält die Information, was gelesen oder geschrieben werden soll, sowie die Adressinformation, anhand der die Anweisung MB_RED_CLIENT die entfernte Adresse berechnet.
- **MB_DATA_LEN** enthält die Anzahl der zu lesenden/schreibenden Werte.

13.5 Modbus-Kommunikation

Die folgende Tabelle zeigt die Beziehung zwischen den Eingangsparametern MB_MODE, MB_DATA_ADDR, MB_DATA_LEN der Anweisung MB_READ_CLIENT und der Modbus-Funktion:

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus-Funktion	Funktion und Datentyp
0	1 bis 9.999	1 bis 2.000	01	1 bis 2.000 Ausgangsbits an der entfernten Adresse 0 bis 9.998 lesen
0	10.001 bis 19.999	1 bis 2.000	02	1 bis 2.000 Eingangsbits an der entfernten Adresse 0 bis 9.998 lesen
0	<ul style="list-style-type: none"> • 40.001 bis 49.999 • 400.001 bis 465.535 	1 bis 125	03	<ul style="list-style-type: none"> • 1 bis 125 Haltereister an der entfernten Adresse 0 bis 9.998 lesen • 1 bis 125 Haltereister an der entfernten Adresse 0 bis 65.534 lesen
0	30.001 bis 39.999	1 bis 125	04	1 bis 125 Eingangswörter an der entfernten Adresse 0 bis 9.998 lesen
1	1 bis 9.999	1	05	1 Ausgangsbit an der entfernten Adresse 0 bis 9.998 schreiben
1	<ul style="list-style-type: none"> • 40.001 bis 49.999 • 400.001 bis 465.535 	1	06	<ul style="list-style-type: none"> • 1 Haltereister an der entfernten Adresse 0 bis 9.998 schreiben • 1 Haltereister an der entfernten Adresse 0 bis 65.534 schreiben
1	1 bis 9.999	2 bis 1.968	15	2 bis 1.968 Ausgangsbits an der entfernten Adresse 0 bis 9.998 schreiben
1	<ul style="list-style-type: none"> • 40.001 bis 49.999 • 400.001 bis 465.535 	2 bis 123	16	<ul style="list-style-type: none"> • 2 bis 123 Haltereister an der entfernten Adresse 0 bis 9.998 schreiben • 2 bis 123 Haltereister an der entfernten Adresse 0 bis 65.534 schreiben
2	1 bis 9.999	1 bis 1.968	15	1 bis 1.968 Ausgangsbits an der entfernten Adresse 0 bis 9.998 schreiben
2	<ul style="list-style-type: none"> • 40.001 bis 49.999 • 400.001 bis 465.535 	1 bis 123	16	<ul style="list-style-type: none"> • 1 bis 123 Haltereister an der entfernten Adresse 0 bis 9.998 schreiben • 1 bis 123 Haltereister an der entfernten Adresse 0 bis 65.534 schreiben
11	Die Anweisung wertet die Parameter MB_DATA_ADDR und MB_DATA_LEN nicht aus, wenn diese Funktion ausgeführt wird.		11	Statuswort und Ereigniszähler des Servers lesen: <ul style="list-style-type: none"> • Das Statuswort stellt den Verarbeitungsstatus dar (0 - keine Verarbeitung, 0xFFFF - Verarbeitung). • Der Ereigniszähler wird erhöht, wenn die Modbus-Anforderung erfolgreich ausgeführt wurde. Wenn während der Ausführung einer Modbus-Funktion ein Fehler aufgetreten ist, wird vom Server eine Meldung gesendet, aber der Ereigniszähler wird nicht hochgezählt.

MB_MODE	MB_DATA_ADDR	MB_DATA_LEN	Modbus-Funktion	Funktion und Datentyp
80	-	1	08	Serverstatus mit dem Diagnosecode 0x0000 prüfen (Schleifenrückgabeprüfung - der Server sendet die Anforderung zurück): 1 WORD pro Aufruf
81	-	1	08	Ereigniszähler des Servers mit dem Diagnosecode 0x000A zurücksetzen: 1 WORD pro Aufruf
101	0 bis 65.535	1 bis 2.000	01	1 bis 2.000 Ausgangsbits an der entfernten Adresse 0 bis 65.535 lesen
102	0 bis 65.535	1 bis 2.000	02	1 bis 2.000 Eingangsbits an der entfernten Adresse 0 bis 65.535 lesen
103	0 bis 65.535	1 bis 125	03	1 bis 125 Halteregister an der entfernten Adresse 0 bis 65.535 lesen
104	0 bis 65.535	1 bis 125	04	1 bis 125 Eingangswörter an der entfernten Adresse 0 bis 65.535 lesen
105	0 bis 65.535	1	05	1 Ausgangsbit an der entfernten Adresse 0 bis 65.535 schreiben
106	0 bis 65.535	1	06	1 Halteregister an der entfernten Adresse 0 bis 65.535 schreiben
115	0 bis 65.535	1 bis 1.968	15	1 bis 1.968 Ausgangsbits an der entfernten Adresse 0 bis 65.535 schreiben
116	0 bis 65.535	1 bis 123	16	1 bis 123 Halteregister an der entfernten Adresse 0 bis 65.535 schreiben
3 bis 10, 12 bis 79, 82 bis 100, 107 bis 114, 117 bis 255				Reserviert

Beispiel:

Variable	Bedeutung
MB_MODE = 1 MB_DATA_ADDR = 1 MB_DATA_LEN = 1	Schreibt 1 Ausgangsbit mit Funktionscode 5, beginnend an der entfernten Adresse 0.
MB_MODE = 15 MB_DATA_ADDR = 1 MB_DATA_LEN = 2	Schreibt 2 Ausgangsbits mit Funktionscode 15, beginnend an der entfernten Adresse 0.
MB_MODE = 104 MB_DATA_ADDR = 17834 MB_DATA_LEN = 125	Schreibt 125 Eingangswörter mit Funktionscode 4, beginnend an der entfernten Adresse 17.834.

MB_DATA_PTR:

Der Parameter MB_DATA_PTR ist ein Pointer auf einen Datenpuffer für die vom Modbus-Server zu empfangenden Daten oder die an den Modbus-Server zu sendenden Daten. Sie können einen globalen Datenbaustein oder einen Merker (M) als Datenpuffer verwenden.

Verwenden Sie für einen Puffer im Merkerbereich (M) einen Pointer im Format ANY wie folgt: "P#Bitadresse" "Datentyp" "Länge" (Beispiel: P#M1000.0 WORD 500)

Abhängig von dem Speicherbereich, in dem sich der Datenpuffer befindet, kann MB_DATA_PTR auf verschiedene Datenstrukturen verweisen:

- Wenn Sie einen globalen DB mit optimiertem Zugriff verwenden, kann MB_DATA_PTR auf eine Variable mit elementarem Datentyp oder auf ein Array aus elementaren Datentypen verweisen. Die folgenden Datentypen werden unterstützt:

Datentyp	Länge in Bits
Bool	1
Byte, SInt, USInt, Char	8
Word, Int, WChar, UInt	16
DWord, DInt, UDInt, Real	32

Sie können alle unterstützten Datentypen für alle Modbus-Funktionen verwenden. Beispiel: MB_RED_CLIENT kann auch ein empfangenes Bit in einer Variable vom Type Byte in eine angegebene Adresse schreiben, ohne andere Bits in diesem Byte zu ändern. Deshalb ist es nicht nötig, ein Array aus Bits zu haben, um bitorientierte Funktionen auszuführen.

- Wenn Sie einen Merkerbereich oder einen globalen DB mit Standardzugriff als Speicherbereich verwenden, gibt es für MB_DATA_PTR keine Einschränkung bei den elementaren Datentypen mehr; MB_DATA_PTR kann dann auch auf komplexe Datenstrukturen wie PLC-Datentypen (UDT) und Systemdatentypen (SDT) verweisen.

Hinweis

Verwendung eines Merkerbereichs als Datenpuffer

Wenn Sie für MB_DATA_PTR einen Merkerbereich als Datenpuffer verwenden, müssen Sie diese Variable beachten. Bei den S7-1200 CPUs sind es 8 KB.

Ausgangsparameter: STATUS_x, RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR

Die CPU zeigt Fehlermeldungen an den Statusausgängen der Anweisung MB_RED_CLIENT an:

Hinweis

Fehlerstatuscodes können im Programmierer als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden:

1. Öffnen Sie den gewünschten Baustein im Programmierer.
 2. Schalten Sie den Programmierstatus ein, indem Sie auf "Beobachten ein/aus" klicken. (Wenn Sie noch keine Online-Verbindung hergestellt haben, wird der Dialog "Online gehen" geöffnet. In diesem Dialog können Sie eine Online-Verbindung herstellen.)
 3. Wählen Sie die Variable aus, die Sie beobachten möchten, und wählen Sie im Kontextmenü unter "Anzeigeformat" das gewünschte Anzeigeformat aus.
-

- Parameter STATUS_x (allgemeine Statusinformationen):

STATUS (W#16#)	Beschreibung
0000	Anweisung fehlerfrei ausgeführt.
0001	Verbindung hergestellt.
0003	Verbindung beendet.
0A90	Die Anweisung MB_RED_CLIENT ist nicht lizenziert. Weitere Informationen finden Sie im obigen Abschnitt "Lizenzierung".
0AFF	Die Verbindung ist nicht konfiguriert und wird nicht verwendet. Die Verbindung "0A" muss konfiguriert werden.
7000	Kein Auftrag aktiv und keine Verbindung hergestellt (REQ = 0, DISCONNECT = 1).
7001	Verbindungsaufbau angestoßen.
7002	Zwischenzeitlicher Aufruf. Verbindung wird aufgebaut.
7003	Verbindung wird beendet.
7004	Verbindung hergestellt und beobachtet. Keine laufende Auftragsbearbeitung.
7005	Daten werden gesendet.
7006	Daten werden empfangen.

- Parameter STATUS_x (Protokollfehler)

STATUS (W#16#)	Beschreibung
80C8	Keine Antwort des Servers im festgelegten Zeitraum. Prüfen Sie die Verbindung zum Modbus-Server. Dieser Fehler wird erst bei Abschluss der konfigurierten Versuchswiederholungen gemeldet. Wenn die Anweisung MB_RED_CLIENT im festgelegten Zeitraum keine Antwort mit der ursprünglich übertragenen Transaktions-ID erhält (siehe statische Variable MB_TRANSACTION_ID), wird dieser Fehlercode ausgegeben.
8380	Empfangenes Modbus-Telegramm hat falsches Format oder es wurden zu wenige Bytes empfangen.
8382	<ul style="list-style-type: none"> Die Länge des Modbus-Telegramms im Telegramm-Header entspricht nicht der Anzahl der empfangenen Bytes. Die Anzahl der Bytes entspricht nicht der Anzahl der tatsächlich übertragenen Bytes (nur Funktionen 1-4). Die Anfangsadresse im empfangenen Telegramm entspricht nicht der gespeicherten Anfangsadresse (Funktionen 5, 6, 15 und 16). Die Anzahl der Wörter entspricht nicht der Anzahl der tatsächlich übertragenen Wörter (Funktionen 15 und 16).
8383	Fehler beim Lesen oder Schreiben von Daten oder Zugriff außerhalb des Adressbereichs von MB_DATA_PTR. Weitere Informationen finden Sie im obigen Abschnitt "MB_DATA_PTR".
8384	<ul style="list-style-type: none"> Ungültiger Ausnahmecode empfangen. Es wurde ein anderer Datenwert empfangen als ursprünglich vom Client gesendet wurde (Funktionen 5, 6 und 8) Ungültiger Statuswert empfangen (Funktion 11)

STATUS (W#16#)	Beschreibung
8385	<ul style="list-style-type: none"> • Diagnosecode nicht unterstützt. • Es wurde ein anderer Teilfunktionscode empfangen als ursprünglich vom Client gesendet wurde (Funktion 8).
8386	Empfangener Funktionscode entspricht nicht dem ursprünglich gesendeten.
8387	Die Protokoll-ID des vom Server empfangenen Modbus TCP-Telegramms ist nicht "0".
8388	Der Modbus-Server hat eine anderen Datenlänge gesendet als verarbeitet wurde. Dieser Fehler tritt nur bei Verwendung der Modbus-Funktionen 5, 6, 15 oder 16 auf.

- Parameter STATUS_x (Parameterfehler)

STATUS (W#16#)	Beschreibung
80B6	Ungültiger Verbindungstyp; es werden nur TCP-Verbindungen unterstützt.
80BB	Parameter ActiveEstablished hat einen ungültigen Wert. Nur aktiver Verbindungsaufbau für Client zulässig (ActiveEstablished = TRUE).
8188	Parameter MB_MODE hat einen ungültigen Wert.
8189	Ungültige Adressierung von Daten am Parameter MB_DATA_ADDR
818A	Ungültige Datenlänge am Parameter MB_DATA_LEN
818B	Parameter MB_DATA_PTR hat einen ungültigen Pointer. Sie sollten auch die Werte der Parameter MB_DATA_ADDR und MB_DATA_LEN prüfen. (Weitere Informationen zu "MB_DATA_ADDR" finden Sie im obigen Abschnitt "MB_DATA_ADDR".)
818C	Zeitüberschreitung am Parameter BLOCKED_PROC_TIMEOUT oder RCV_TIMEOUT (siehe statische Variablen der Anweisung). BLOCKED_PROC_TIMEOUT und RCV_TIMEOUT müssen zwischen 0,5 s und 55,0 s sein.
8200	<ul style="list-style-type: none"> • Die CPU verarbeitet derzeit an dem Port eine andere Modbus-Anforderung. • Eine andere Instanz von MB_RED_CLIENT mit den gleichen Verbindungsparametern verarbeitet eine bestehende Modbus-Anforderung.

Hinweis

Fehlercodes von intern verwendeten Kommunikationsanweisungen

Bei der Anweisung MB_RED_CLIENT können, neben den in den Tabellen aufgeführten Fehlern, von den Kommunikationsanweisungen (TCON, TDISCON, TSEND, TRCV, T_DIAG und TRESET) verursachte Fehler, die von der Anweisung verwendet werden, auftreten.

Die CPU weist die Fehlercodes über den Instanz-Datenbaustein der Anweisung MB_RED_CLIENT zu. Die CPU zeigt die Fehlercodes für die jeweilige Anweisung im Abschnitt "Static" unter STATUS an.

Die Bedeutung der Fehlercodes ist in der Dokumentation der entsprechenden Kommunikationsanweisung verfügbar.

Hinweis

Kommunikationsfehler beim Senden oder Empfangen von Daten

Wenn beim Senden oder Empfangen von Daten ein Kommunikationsfehler auftritt (80C4 (Temporärer Kommunikationsfehler. Die angegebene Verbindung wird temporär beendet.), 80C5 (Der entfernte Partner hat die Verbindung aktiv beendet.) oder 80A1 (Die angegebene Verbindung wurde beendet oder ist noch nicht aufgebaut.)), beendet die CPU die bestehende Verbindung.

Das bedeutet, dass Sie alle zurückgegebenen STATUS-Werte sehen können, wenn die Verbindung beendet wird, und dass der STATUS-Code, der den Abbruch der Verbindung verursacht hat, nur ausgegeben wird, wenn die Verbindung beendet wird.

Beispiel: Wenn beim Empfangen von Daten ein temporärer Kommunikationsfehler auftritt, wird zunächst der STATUS 7003 (ERROR = False) und dann 80C4 (ERROR = True) ausgegeben.

MB_RED_SERVER (Kommunikation über PROFINET als Modbus TCP-Server)

Sie können diese Anweisung verwenden, um eine Verbindung zwischen einer S7-1200 CPU und einem Gerät, das das Modbus TCP-Protokoll unterstützt, herzustellen.

Tabelle 13-74 Anweisung MB_RED_SERVER

KOP/FUP	SCL	Beschreibung																																				
<p style="text-align: center;">%DB5 "MB_RED_SERVER_DB"</p> <div style="border: 1px solid gray; padding: 5px;"> <p style="text-align: center;">MB_RED_SERVER</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-right: 1px solid gray;">EN</td> <td style="width: 50%;">ENO</td> </tr> <tr> <td style="border-right: 1px solid gray;">REG_KEY</td> <td>LICENSED</td> </tr> <tr> <td style="border-right: 1px solid gray;">DISCONNECT</td> <td>IDENT_CODE</td> </tr> <tr> <td style="border-right: 1px solid gray;">MB_HOLD_REG</td> <td>DR_NDR_0A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>ERROR_0A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>STATUS_0A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>DR_NDR_1A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>ERROR_1A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>STATUS_1A</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>DR_NDR_0B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>ERROR_0B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>STATUS_0B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>DR_NDR_1B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>ERROR_1B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>STATUS_1B</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>RED_ERR_S7</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>RED_ERR_DEV</td> </tr> <tr> <td style="border-right: 1px solid gray;"></td> <td>TOT_COM_ERR</td> </tr> </table> </div>	EN	ENO	REG_KEY	LICENSED	DISCONNECT	IDENT_CODE	MB_HOLD_REG	DR_NDR_0A		ERROR_0A		STATUS_0A		DR_NDR_1A		ERROR_1A		STATUS_1A		DR_NDR_0B		ERROR_0B		STATUS_0B		DR_NDR_1B		ERROR_1B		STATUS_1B		RED_ERR_S7		RED_ERR_DEV		TOT_COM_ERR	<pre>"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_);</pre>	<p>Die Anweisung MB_RED_SERVER kommuniziert als Modbus TCP-Server über die PROFINET-Verbindung.</p> <p>Die Anweisung MB_RED_SERVER verarbeitet Verbindungsanforderungen eines Modbus TCP-Clients, empfängt und verarbeitet Modbus-Anforderungen und sendet Antworten.</p>
EN	ENO																																					
REG_KEY	LICENSED																																					
DISCONNECT	IDENT_CODE																																					
MB_HOLD_REG	DR_NDR_0A																																					
	ERROR_0A																																					
	STATUS_0A																																					
	DR_NDR_1A																																					
	ERROR_1A																																					
	STATUS_1A																																					
	DR_NDR_0B																																					
	ERROR_0B																																					
	STATUS_0B																																					
	DR_NDR_1B																																					
	ERROR_1B																																					
	STATUS_1B																																					
	RED_ERR_S7																																					
	RED_ERR_DEV																																					
	TOT_COM_ERR																																					

Tabelle 13-75 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REG_KEY ¹	IN	STRING[17]	Registrierungscode für die Lizenzierung Die Anweisung MB_RED_SERVER muss auf jeder CPU einzeln lizenziert werden.
DISCONNECT	IN	Bool	Mit der Anweisung MB_RED_SERVER gehen Sie eine passive Verbindung mit einem Partnermodul ein. Der Server reagiert auf eine Verbindungsanforderung von den IP-Adressen, die in den Verbindungsbeschreibungen als spezifiziert oder nicht spezifiziert angegeben sind. Mit diesem Parameter steuern Sie, wann eine Verbindungsanforderung akzeptiert wird: <ul style="list-style-type: none"> 0: Die CPU stellt eine passive Verbindung her, wenn es keine Kommunikationsverbindung gibt. 1: Initialisierung des Verbindungsabbaus Wenn der Eingang gesetzt ist, verarbeitet die CPU keine zusätzlichen Client-Anforderungen, und der Abbau der Verbindung wird eingeleitet. Nach dem erfolgreichen Abbau der Verbindung wird am Parameter STATUS_x der Wert "0003" ausgegeben.
MB_HOLD_REG ²	IN_OUT	Variant	Pointer auf das Modbus-Halteregister der Anweisung MB_RED_SERVER MB_HOLD_REG muss immer auf einen Speicherbereich verweisen, der größer als zwei Bytes ist. Das Halteregister enthält die Werte, auf die ein Modbus-Client über die Modbus-Funktionen 3 (Lesen), 6 (Schreiben), 16 (Mehrmaliges Schreiben) und 23 (Lesen und Schreiben in einem Auftrag) zugreifen kann.
LICENSED ¹	OUT	Bool	<ul style="list-style-type: none"> 0: Anweisung ist nicht lizenziert 1: Anweisung ist lizenziert
IDENT_CODE ¹	OUT	STRING[18]	Identifikation für die Lizenzierung. Verwenden Sie diese Zeichenfolge, um den Registrierungscode REG_KEY anzufordern.
DR_NDR_OA	OUT	Bool	"Data Read" oder "New Data Ready" für Verbindung 0A: <ul style="list-style-type: none"> 0: Keine neuen Daten 1: Vom Modbus-Client gelesene oder geschriebene neue Daten
ERROR_OA	OUT	Bool	Tritt bei einem Aufruf der Anweisung MB_RED_SERVER für die Verbindung 0A ein Fehler auf, wird der Ausgang des Parameters ERROR_OA auf "1" gesetzt. Ausführliche Informationen zur Fehlerursache werden vom Parameter STATUS_OA angegeben.
STATUS_OA ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 0A.
DR_NDR_1A	OUT	Bool	"Data Read" oder "New Data Ready" für Verbindung 1A: <ul style="list-style-type: none"> 0: Keine neuen Daten 1: Vom Modbus-Client gelesene oder geschriebene neue Daten
ERROR_1A	OUT	Bool	Tritt bei einem Aufruf der Anweisung MB_RED_SERVER für die Verbindung 1A ein Fehler auf, wird der Ausgang des Parameters ERROR_1A auf "1" gesetzt. Ausführliche Informationen zur Fehlerursache werden vom Parameter STATUS_1A angegeben.
STATUS_1A ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 1A.
DR_NDR_OB	OUT	Bool	"Data Read" oder "New Data Ready" für Verbindung 0B: <ul style="list-style-type: none"> 0: Keine neuen Daten 1: Vom Modbus-Client gelesene oder geschriebene neue Daten

Parameter und Datentyp		Datentyp	Beschreibung
ERROR_0B	OUT	Bool	Tritt bei einem Aufruf der Anweisung MB_RED_SERVER für die Verbindung 0B ein Fehler auf, wird der Ausgang des Parameters ERROR_0B auf "1" gesetzt. Ausführliche Informationen zur Fehlerursache werden vom Parameter STATUS_0B angegeben.
STATUS_0B ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 0B.
DR_NDR_1B	OUT	Bool	"Data Read" oder "New Data Ready" für Verbindung 1B: <ul style="list-style-type: none"> • 0: Keine neuen Daten • 1: Vom Modbus-Client gelesene oder geschriebene neue Daten
ERROR_1B	OUT	Bool	Tritt bei einem Aufruf der Anweisung MB_RED_SERVER für die Verbindung 1B ein Fehler auf, wird der Ausgang des Parameters ERROR_1B auf "1" gesetzt. Ausführliche Informationen zur Fehlerursache werden vom Parameter STATUS_1B angegeben.
STATUS_1B ³	OUT	Word	Ausführliche Statusinformationen der Anweisung zur Verbindung 1B.
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> • 0: Kein Redundanzfehler in SIMATIC • 1: Redundanzfehler in SIMATIC
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> • 0: Kein Redundanzfehler auf der Seite des Linkpartners • 1: Redundanzfehler auf der Seite des Linkpartners
RED_ERR_S7 ³	OUT	Bool	<ul style="list-style-type: none"> • 0: Mindestens 1 konfigurierte Verbindung ist hergestellt • 1: Vollständiger Kommunikationsverlust, alle konfigurierten Verbindungen sind beendet

¹ Weitere Informationen finden Sie im nachfolgenden Abschnitt "Lizenzierung".

² Weitere Informationen finden Sie im nachfolgenden Abschnitt "Eingangsparameter MB_HOLD_REG".

³ Weitere Informationen finden Sie im nachfolgenden Abschnitt "Ausgangsparameter: ERROR_x, RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR".

Hinweis

Sicherheitshinweise

Jeder Client im Netzwerk hat Lese- und Schreibzugriff auf die Ein- und Ausgänge im Prozessbild und auf den Datenbaustein- oder Bitspeicherbereich, die im Modbus-Halteregister festgelegt sind. Diese Option soll den Zugriff auf eine IP-Adresse einschränken und unbefugtes Lesen und Schreiben verhindern. Zu beachten ist jedoch, dass die geteilte Adresse auch für den unbefugten Zugriff verwendet werden kann.

Hinweis

Vorausgesetzte Firmwareversion der CPU

Für die in diesem Abschnitt des Handbuchs beschriebenen Modbus TCP-Anweisungen ist die Firmwareversion V4.2 oder höher erforderlich.

Für die Verwendung der Anweisung wird kein zusätzliches Hardwaremodul benötigt.

Mehrere Serververbindungen

Die CPUs können:

- Mehrere Serververbindungen verarbeiten
- Mehrere Verbindungen von verschiedenen Clients gleichzeitig an einem Serverport annehmen

Die maximale Anzahl von Verbindungen hängt von der verwendeten CPU ab und ist in den technischen Daten der CPU zu finden. Die Gesamtzahl von Verbindungen einer CPU, einschließlich der von Modbus TCP-Clients und -Servern darf die maximale Anzahl der unterstützten Verbindungen nicht überschreiten.

Beachten Sie bei Serververbindungen die folgenden Regeln:

- Jede MB_RED_SERVER-Verbindung muss einen eindeutigen Instanz-DB verwenden.
- Für jeden einzelnen Client, der eine Verbindung zum Serverport herstellen möchte, ist eine eindeutige Verbindung/Verbindungs-ID erforderlich.
- Die Verbindungs-IDs müssen CPU-weit eindeutig sein.

Zuordnung der Modbus-Adressen zum Prozessabbild

Die Anweisung MB_RED_SERVER gestattet eingehenden Modbus-Funktionen (1, 2, 4, 5 und 15) direkten Lese- und Schreibzugriff auf die Eingänge und Ausgänge des Prozessabbilds der CPU (Verwendung der Datentypen BOOL und WORD).

Bei S7-1200 CPUs beträgt der Adressraum für das Prozessabbild der Eingänge und das Prozessabbild der Ausgänge 1 KB.

Die folgende Tabelle zeigt den Adressraum der oben aufgeführten Modbus-Funktionen:

Modbus-Funktion					
Funktionscode	Funktion	Datenbereich	Adressraum		
01	Lesen: Bits	Output	0	bis	65.535
02	Lesen: Bits	Input	0	bis	65.535
04	Lesen: WORD	Input	0	bis	65.535
05	Schreiben: Bit	Output	0	bis	65.535
15	Schreiben: Bits	Output	0	bis	65.535

Eingehende Modbus-Anforderungen mit den Funktionscodes 3, 6, 16 und 23 schreiben oder lesen die Modbus-Halteregister (Sie geben das Halteregister mit dem Parameter MB_HOLD_REG oder über Data_Area_Array an).

Modbus-Funktionen

In der folgenden Tabelle werden alle Modbus-Funktionen aufgeführt, die von der Anweisung MB_RED_SERVER unterstützt werden:

Funktionscode	Beschreibung
01	Ausgangsbits lesen
02	Eingangsbits lesen

Funktionscode	Beschreibung
03	Halteregister lesen
04	Eingangswörter lesen
05	Ausgangsbit schreiben
06	Halteregister schreiben
08	<p>Diagnosefunktion:</p> <ul style="list-style-type: none"> Echotest (Teilfunktion 0x0000): Die Anweisung MB_RED_SERVER empfängt ein Datenwort und gibt es unverändert an den Modbus-Client zurück. Ereigniszähler zurücksetzen (Teilfunktion 0x000A): Die Anweisung MB_RED_SERVER setzt die folgenden Ereigniszähler zurück: "Success_Count", "Xmt_Rcv_Count", "Exception_Count", "Server_Message_Count" und "Request_Count".
11	<p>Diagnosefunktion: Ereigniszähler der Kommunikation abrufen</p> <p>Die Anweisung MB_RED_SERVER nutzt einen internen Ereigniszähler für die Kommunikation, um die Anzahl erfolgreich ausgeführter Lese- und Schreibenanforderungen, die an den Modbus-Server gesendet wurden, zu erfassen.</p> <p>Der Ereigniszähler wird bei den Funktionen 8 und 11 nicht hochgezählt. Gleiches gilt bei Anforderungen, die einen Kommunikationsfehler verursachen, beispielsweise, wenn ein Protokollfehler aufgetreten ist. Der Funktionscode in der empfangenen Modbus-Anforderung wird nicht unterstützt.</p>
15	Ausgangsbits lesen
16	Halteregister schreiben
23	Mit einer Anforderung ein Halteregister schreiben und ein Halteregister lesen

Betrieb und Redundanz

Die Kommunikationsteilnehmer können als unabhängige oder redundante Teilnehmer ausgeführt werden. Wenn einer der Partner unabhängig ausgeführt ist, wird dies als einseitige Redundanz bezeichnet. Wenn beide Partner unabhängig ausgeführt sind, wird dies zweiseitige Redundanz genannt:

- Einseitige Redundanz:
 - Für jede Verbindung zwischen den Kommunikationspartnern muss jeweils eine Verbindung konfiguriert werden. Die Verbindungspunkte der **SIMATIC S7** werden als **0** und **1** bezeichnet. Die Verbindungspunkte der **Kommunikationspartner** werden als **A** und **B** bezeichnet.
Bei der R-CPU oder H-CPU bezieht sich 1 auf den Verbindungspunkt 0, und R-CPU oder H-CPU 2 bezieht sich auf den Verbindungspunkt 1.
 - Konfiguration: Wenn die S7 redundant ausgeführt ist, wird eine Verbindung vom S7-Verbindungspunkt 0 zu Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **A** => Verbindung **0A**), und es wird eine Verbindung vom S7-Verbindungspunkt 1 zum Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **A** => Verbindung **1A**). Die Abbildung zeigt die Verbindungsbezeichnungen:

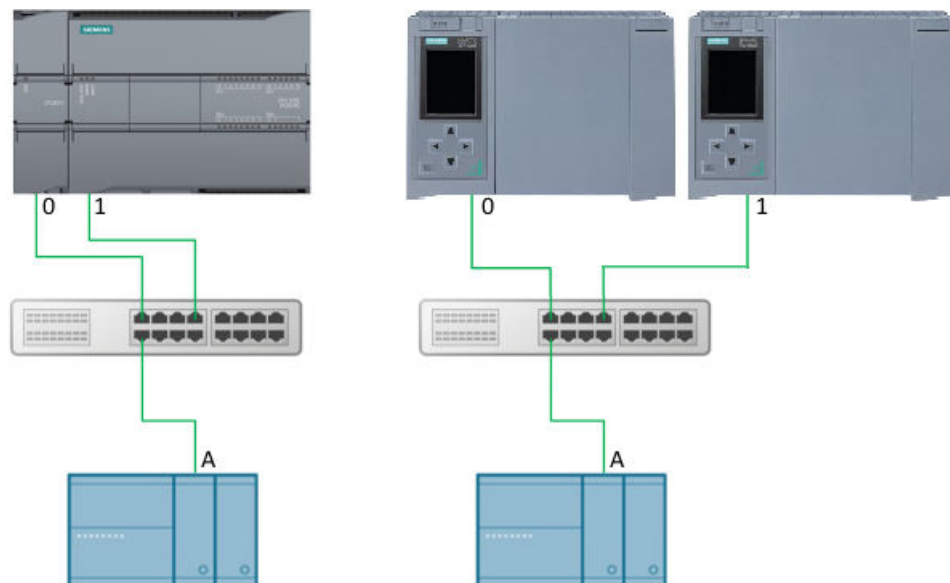


Bild 13-12 Einseitige Redundanz S7

- Wenn die S7 unabhängig und der Linkpartner redundant ausgeführt ist, wird eine Verbindung vom S7-Verbindungspunkt 0 zu Knotenpunkt A des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt 0 zum Partner/Knoten A => Verbindung 0A), und es wird eine Verbindung vom S7-Verbindungspunkt 0 zum Knotenpunkt B des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt 0 zum Partner/Knoten B => Verbindung 0B). Die Abbildung zeigt die Verbindungsbezeichnungen:

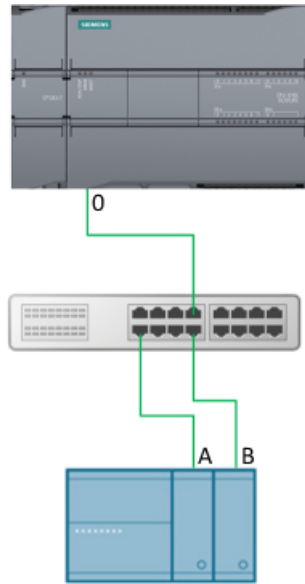


Bild 13-13 Einseitige Redundanz Partner

- Zweiseitige Redundanz:
 - Beschreibung: Für jede Verbindung zwischen den Kommunikationspartnern muss jeweils eine Verbindung konfiguriert werden. Die Verbindungspunkte der **SIMATIC S7** werden als 0 und 1 bezeichnet. Die Verbindungspunkte der **Kommunikationspartner** werden als A und B bezeichnet. Bei der R-CPU oder H-CPU bezieht sich 1 auf den Verbindungspunkt 0, und R-CPU oder H-CPU 2 bezieht sich auf den Verbindungspunkt 1.

13.5 Modbus-Kommunikation

- Konfiguration: Bei zweiseitiger Redundanz werden vom Verbindungspunkt 0 zwei Verbindungen erstellt (Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **A** => Verbindung **0A** und Verbindung vom S7-Verbindungspunkt **0** zum Partner/Knoten **B** => Verbindung **0B**), und es werden zwei Verbindungen vom Verbindungspunkt 1 der S7 zu den Knotenpunkten A und B des Linkpartners erstellt (Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **A** => Verbindung **1A** und Verbindung vom S7-Verbindungspunkt **1** zum Partner/Knoten **B** => Verbindung **1B**). Die Abbildung zeigt die Verbindungsbezeichnungen:

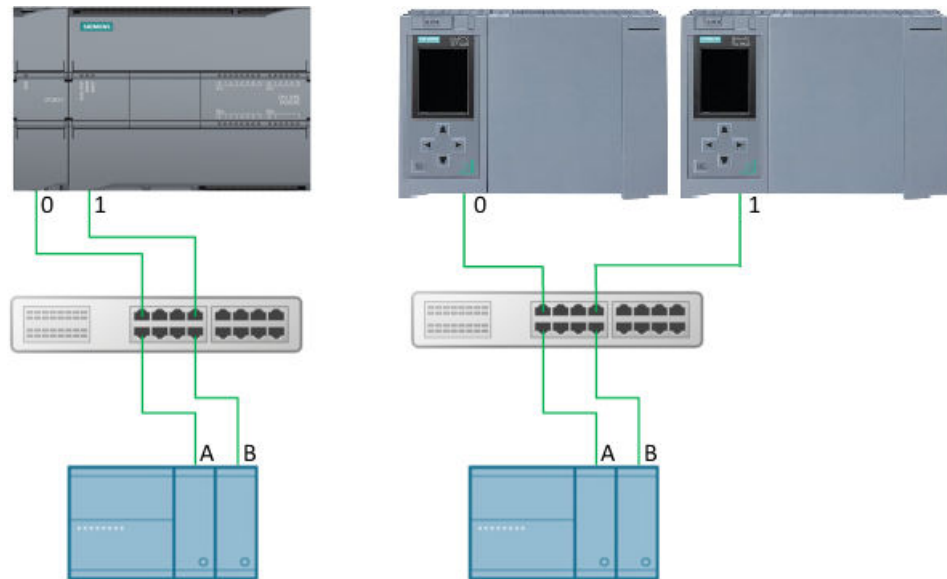


Bild 13-14 Zweiseitige Redundanz:

- Telegrammverarbeitung: Telegramme können über alle konfigurierten Verbindungen empfangen werden. Der Client kann Telegramme entweder über eine Verbindung oder über alle Verbindungen senden. Wenn ein Telegramm an einer Verbindung empfangen wurde, zeigt die CPU den Status am entsprechenden Ausgang DR_NDR_x oder ERROR_x an. Jede Verbindung wird unabhängig ausgeführt und hat keinen Einfluss auf die Anzeige der anderen Verbindungen.
- Redundanzgänge RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR:
 - Die Redundanzbits RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR werden abhängig von den Zuständen der Statusausgänge gesetzt:

Anzahl der fehlerhaften Verbindungen	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
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

Bild 13-15 Anzeige der Alarmbits für Redundanzeinrichtung auf beiden Seiten

Anzahl der fehlerhaften Verbindungen	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

Bild 13-16 Anzeige der Alarmbits für Redundanzeinrichtung auf einer Seite

Hinweis

Portnummern für Client und Server

Der Modbus-Client verwendet eine Portnummer mit Beginn an 2000. Der Modbus-Server wird üblicherweise über die Portnummer 502 angesprochen. Abhängig von der CPU ist es möglich, Port 502 für mehrere Verbindungen zu konfigurieren (Multiport). Wenn der lokale Port 502 für zwei oder mehr Verbindungen konfiguriert wurde, werden die anfordernden Clients bei un spezifizierten Verbindungen zufällig auf die bestehenden Serververbindungen verteilt. Der erste Client, der eine Verbindung zur Anweisung "MB_RED_SERVER" herstellen möchten, wird nicht automatisch Verbindung 0A zugewiesen. Nachdem die Client-Anforderungen den Serververbindungen zugewiesen wurden, bleibt die Zuweisung für die Dauer des Telegrammaustauschs bestehen, bis die Verbindung beendet wird.

Parametrierung

Bei der S7-1200 können Sie die Anweisung MB_RED_SERVER V1.0 und V1.1 verwenden. Die CPU implementiert die Verbindungen über die lokale Schnittstelle der CPU oder des CM/CP. Über die Struktur TCON_IP_V4 konfiguriert die CPU die Verbindungen und baut sie auf.

Konfiguration von MB_RED_SERVER: Sie nehmen die folgenden Einstellungen im Konfigurationsdialog der Anweisung MB_RED_SERVER vor:

- Verbindungsparameter für die Verbindungen 0A, 1A, 0B und 1B (weitere Informationen zur Redundanzkonfiguration finden Sie unter "Betrieb und Redundanz" oben)
- Interne Parameter (optional)

Sie können den Konfigurationsdialog mit der Anweisung MB_RED_SERVER oder über die Technologieobjekte öffnen:

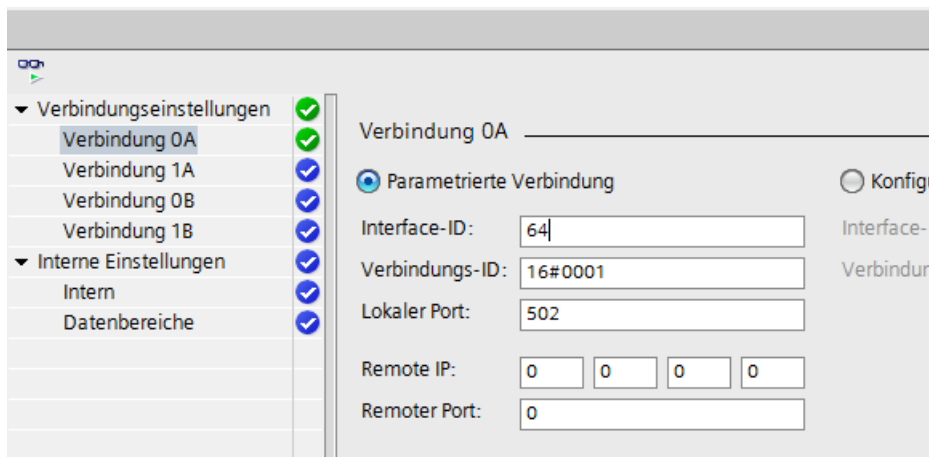


Bild 13-17 Parametrierte Serververbindung

Verbindung OA

Parametrierte Verbindung
 Konfigurierte Verbindung

Interface-ID:
 Interface-ID:

Verbindungs-ID:
 Verbindungs-ID:

Lokaler Port:

Remote IP:

Remoter Port:

Bild 13-18 Konfigurierte Serververbindung

Variable	Startwert	Beschreibung
Konfigurierte Verbindungen		
Interface ID	64	HW-Kennung der verwendeten PN-Schnittstelle
Connection ID	16#0000	Verbindungs-IDs für die verwendeten Verbindungen. Diese Verbindungs-IDs müssen CPU-weit eindeutig sein.
Local port	502	Lokale Portnummer des Serverbausteins. Der Standardport für den Modbus/TCP-Server ist 502.
Remote IP	0.0.0.0	Entfernte IP-Adresse des Clients. Standardmäßig ist für den Client keine IP-Adresse eingegeben.
Remote port	0	Entfernte Portnummer des Clients. Standardmäßig ist für den Client keine Portnummer eingegeben.
Konfigurierte Verbindungen		
Interface ID	64	HW-Kennung der verwendeten PN-Schnittstelle
Connection ID	16#0000	Verbindungs-IDs für die verwendeten Verbindungen. Diese Verbindungen werden in der Netzsicht konfiguriert.

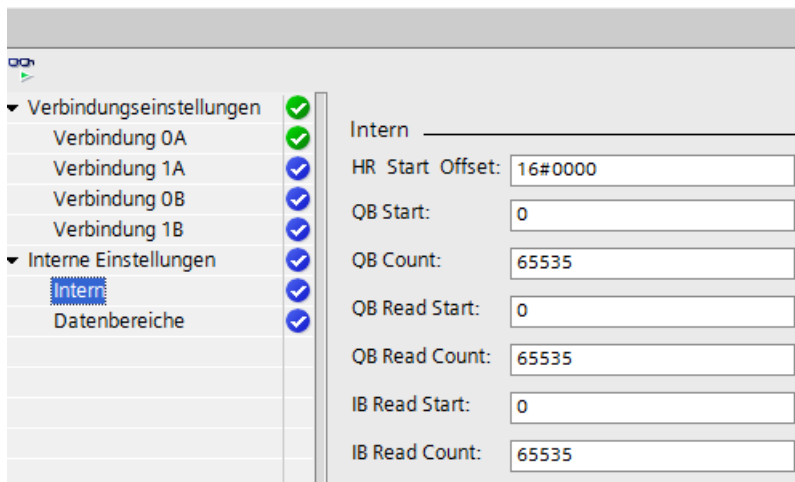


Bild 13-19 Interne Parameter (optional)

Variable	Daten-typ	Start-wert	Beschreibung
HR_Start_Offset	WORD	0	Zuweisung der Anfangsadresse des Modbus-Halteregisters
QB_Start	UINT	0	Anfangsadresse des zulässigen Adressbereichs der Ausgänge, in die der Modbus-Master schreiben kann (Bytes 0 bis 65535)
QB_Count	UINT	0	Anzahl der Ausgangsbytes, in die der Modbus-Master schreiben kann <i>Beispiel:</i> <ul style="list-style-type: none"> • QB_Start = 0 und QB_Count = 10: Der Modbus-Master kann in die Ausgangsbytes 0 bis 9 schreiben. • QB_Count = 0: Der Modbus-Master kann in kein Ausgangsbyte schreiben.
QB_Read_Start	UINT	0	Anfangsadresse des zulässigen Adressbereichs der Ausgänge, die der Modbus-Master lesen kann (Bytes 0 bis 65535)
QB_Read_Count	UINT	0	Anzahl der Ausgangsbytes, die der Modbus-Master lesen kann <i>Beispiel:</i> <ul style="list-style-type: none"> • QB_Read_Start = 0 und QB_Read_Count = 10: Der Modbus-Master kann die Ausgangsbytes 0 bis 9 lesen. • QB_Read_Count = 0: Der Modbus-Master kann kein Ausgangsbyte lesen.
IB_Read_Start	UINT	0	Anfangsadresse des zulässigen Adressbereichs der Eingänge, die der Modbus-Master lesen kann (Bytes 0 bis 65535)
IB_Read_Count	UINT	0	Anzahl der Eingangsbytes, die der Modbus-Master lesen kann <i>Beispiel:</i> <ul style="list-style-type: none"> • IB_Read_Start = 0 und IB_Read_Count = 10: Der Modbus-Master kann die Eingangsbytes 0 bis 9 lesen. • IB_Read_Count = 0: Der Modbus-Master kann kein Eingangsbyte lesen.
Data_Area_Array	ARRAY [1..8]		

Variable	Daten-typ	Start-wert	Beschreibung
data_type	UINT	0	Datentyp: 0 bis 4
db	UINT	0	Datenbausteinnummer
start	UINT	0	Erste Modbus-Adresse im Datenbaustein
length	UINT	0	Anzahl der Modbus-Werte im Datenbaustein

Adressierung über die statische Variable HR_Start_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 0.

Beispiel: Das Haltereister beginnt an MW100 und hat eine Länge von 100 Wörtern.

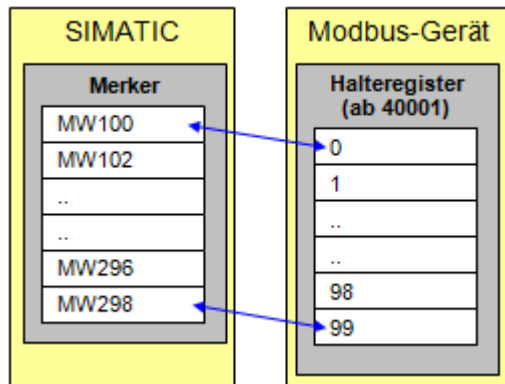


Bild 13-20 HR_Start_Offset_0

Sie können die Variable HR_Start_Offset so definieren, dass das Modbus-Haltereister eine andere Anfangsadresse als 0 hat.

Beispiel: Ein Offset-Wert von 20 im Parameter HR_Start_Offset bedeutet, dass die Anfangsadresse des Haltereisters von 0 nach 20 verschoben wird. Dadurch wird immer dann ein Fehler verursacht, wenn Sie das Haltereister unterhalb der Adresse 20 und oberhalb der Adresse 119 adressieren.

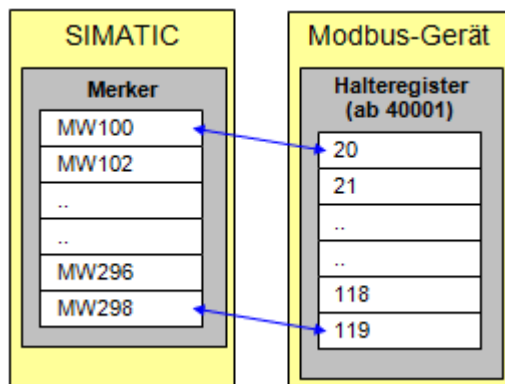


Bild 13-21 HR_Start_Offset_20

Data_Area_Array [1..8]

Für die Zuordnung der MODBUS-Adressen im SIMATIC S7-Speicher stehen acht Datenbereiche zur Verfügung. Wenn der Datenbereich mit dem Datentyp "Haltereister" definiert ist, wird der Parameter MB_HOLD_REG nicht ausgewertet. Stattdessen schreibt oder liest der Modbus-

Master abhängig von der Auftragsart das Modbus-Register und Bits in den Datenbausteinen. Die CPU kann diese Werte in der nachfolgenden Ausführung des Programms weiter verarbeiten.

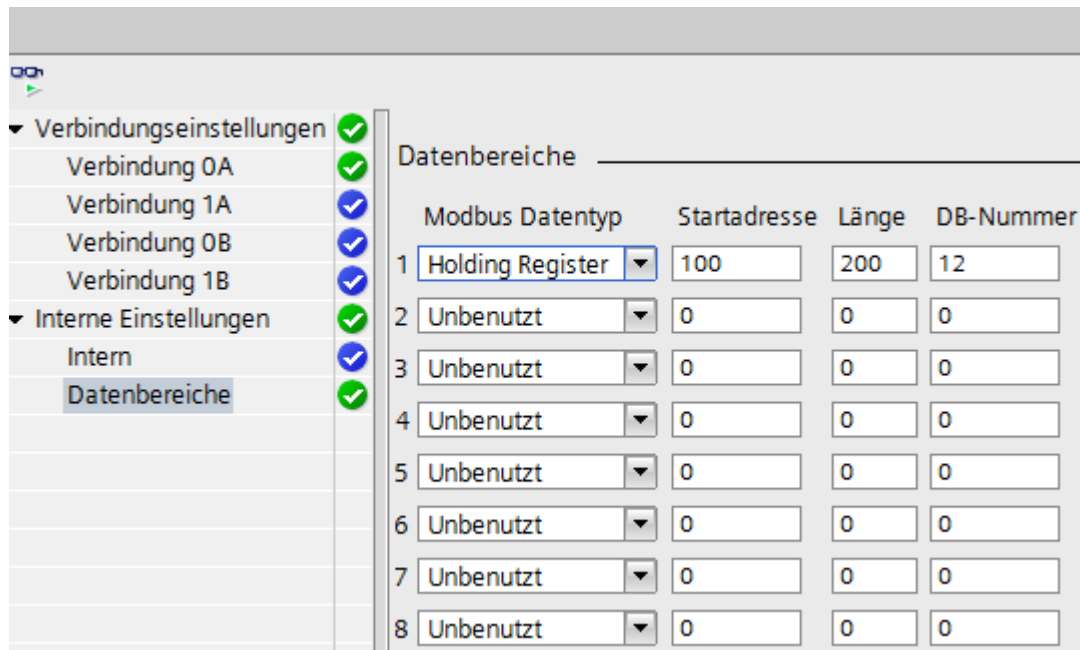


Bild 13-22 Serverdatenbereiche

Sie können mit jedem Auftrag nur aus einem DB lesen oder in einen DB schreiben. Zugriff auf Register oder Bitwerte, die sich in verschiedenen DBs befinden, müssen, selbst wenn die Nummern lückenlos aufeinander folgen, in zwei Aufträge unterteilt werden. Beachten Sie dies bei der Konfiguration. Es ist möglich, mehr Modbus-Bereiche (Register oder Bitwerte) in einem Datenbaustein zuzuordnen als der Modbus-Master in einem Telegramm verarbeiten kann.

data_type

Der Parameter data_type gibt an, welche MODBUS-Datentypen der Modbus-Master in diesem Datenbaustein zuordnet. Wenn in data_type der Wert "0" eingegeben wird, verwendet der Modbus-Master den entsprechenden Datenbereich nicht. Wenn der Modbus-Master mehrere Data_Areas verwenden soll, müssen Sie diese aufeinander folgend definieren. Der Modbus-Master verarbeitet keine Einträge nach einem data_type = 0.

Ken-nung	Datentyp	Beschreibung
0	Bereich nicht verwendet	
1	Ausgangsbits (Spulen)	Bit
2	Eingangsbits (Eingänge)	Bit
3	Halteregister	Wort
4	Eingangswörter (Eingangsregister)	Wort

db

Der Parameter db gibt den Datenbaustein an, der die unten definierten MODBUS-Register oder Bitwerte zuordnet. Die CPU erlaubt die DB-Nummer 0 nicht, weil sie für das System reserviert ist.

start, length

Der Parameter start gibt die erste Modbus-Adresse an, die der Modbus-Master in Datenwort 0 des DB zuordnet. Der Parameter length definiert die Länge, also die Anzahl der MODBUS-Adressen, die der Modbus-Master im Datenbaustein zuordnet. Die definierten Datenbereiche dürfen sich nicht überschneiden. Der Parameter length darf nicht 0 sein.

Beispiel: Adresszuordnung mit Data_Area_Array

Datenbereich 1	data_type	3: Halteregeister
	db	11
	start	0
	length	500
Datenbereich 2	data_type	3: Halteregeister
	db	12
	start	720
	length	181
Datenbereich 3	data_type	4: Eingangswörter
	db	13
	start	720
	length	281
Datenbereich 4	data_type	1: Ausgangsbits
	db	14
	start	640
	length	611
Datenbereich 5	data_type	2: Eingangsbit
	db	15
	start	1700
	length	601
Datenbereich 6	data_type	1: Ausgangsbits
	db	16
	start	1700
	length	601
Datenbereich 7	data_type	Nicht verwendet
	db	0
	start	0
	length	0
Datenbereich 8	data_type	Nicht verwendet
	db	0
	start	0
	length	0

13.5 Modbus-Kommunikation

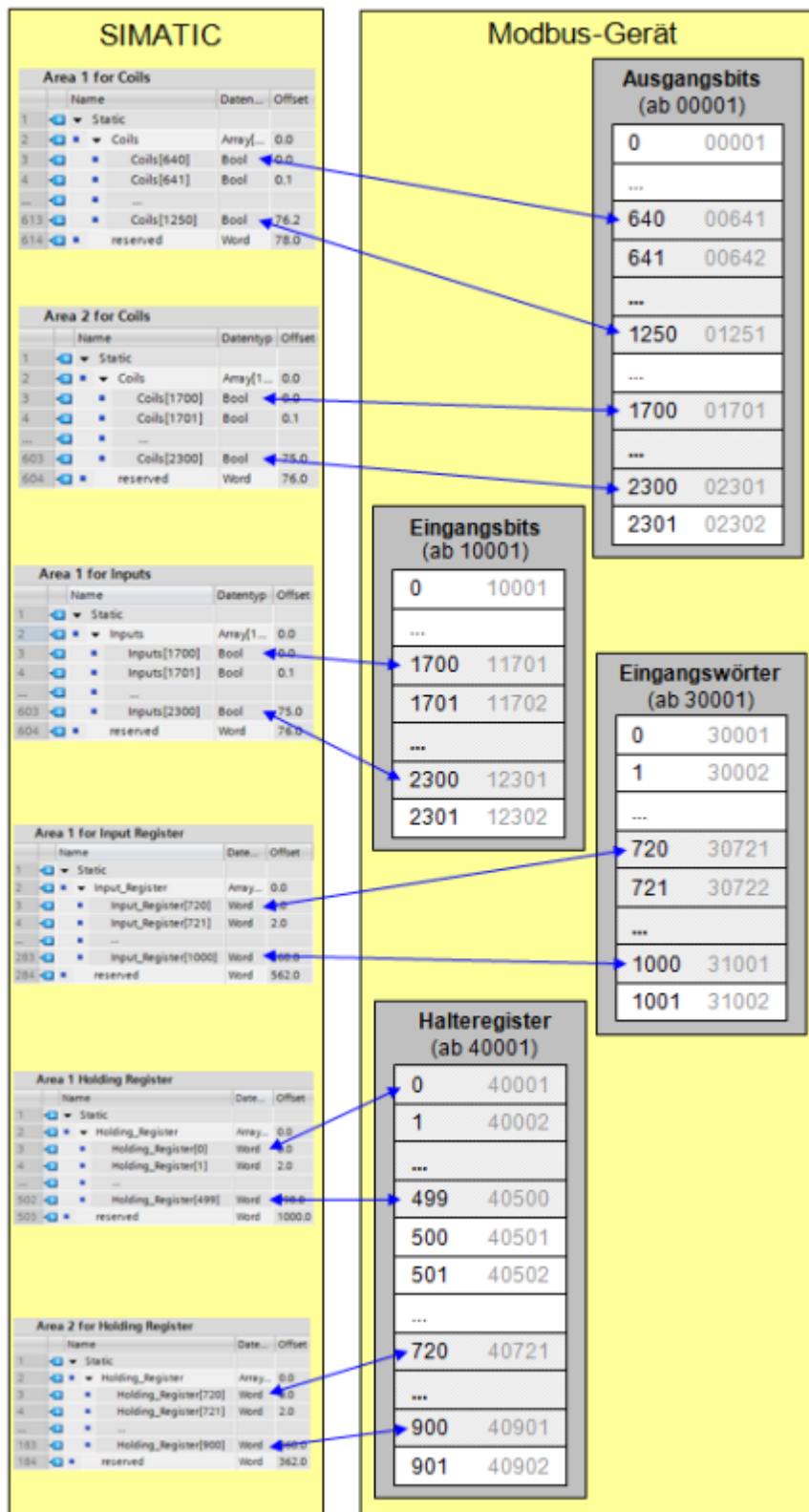


Bild 13-23 Adressaufbau

Lizenzierung

Die Anweisung MB_RED_SERVER unterliegt einer Gebühr und Sie müssen die Anweisung für jede CPU einzeln lizenzieren. Die Lizenzierung läuft in zwei Schritten ab:

- Anzeigen des IDENT_CODE der Lizenz
- Eingeben des Registrierungsschlüssels REG_KEY: Sie müssen den Registrierungsschlüssel REG_KEY bei jeder Anweisung MB_RED_SERVER eingeben. Speichern Sie den REG_KEY in einem globalen Datenbaustein, aus dem alle MB_RED_SERVER-Anweisungen den erforderlichen Registrierungsschlüssel erhalten.

Vorgehensweise zum Anzeigen des IDENT_CODE der Lizenz:

1. Parametrieren Sie die Anweisung MB_RED_SERVER entsprechend Ihren Anforderungen in einem zyklischen OB. Laden Sie das Programm in die CPU und versetzen Sie die CPU in RUN.
2. Öffnen Sie den Instanz-DB der Modbus-Anweisung und klicken Sie auf die Schaltfläche "Alle beobachten".
3. Der Instanz-DB zeigt am Ausgang IDENT_CODE eine 18-stellige Zeichenfolge an.

	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]	"	"RTPCFIGDCDIHJHAH4"
12	DONE	Bool	false	FALSE
13	BUSY	Bool	false	FALSE
14	ERROR	Bool	false	FALSE

Bild 13-24 Lizenz

4. Kopieren Sie diese Zeichenfolge mit der Funktion Kopieren/Einfügen aus dem Datenbaustein und fügen Sie sie in das Formular ein (das Sie nach der Bestellung des Produkts per E-Mail erhalten haben oder sich auch auf der CD befindet).
5. Senden Sie dieses Formular über eine Serviceanfrage an Kundensupport (<https://support.industry.siemens.com/my/ww/de/requests/#createRequest>). Daraufhin erhalten Sie den Registrierungsschlüssel für Ihre CPU.

13.5 Modbus-Kommunikation

Vorgehensweise zum Eingeben des Registrierungsschlüssels REG_KEY:

1. Fügen Sie über "Neuen Baustein hinzufügen..." einen neuen globalen Datenbaustein mit einem eindeutigen symbolischen Namen, z.B. "Lizenz-DB", ein.
2. Erstellen Sie in diesem Baustein einen Parameter REG_KEY mit dem Datentyp STRING[17].

Lizenz_DB				
	Name	Datentyp	Offset	Startwert
1	Static			
2	REG_KEY	String[17]	0.0	

Bild 13-25 REG_KEY

3. Kopieren Sie den übertragenen 17-stelligen Registrierungsschlüssel mit der Funktion Kopieren/Einfügen in die Spalte "Startwert".
4. Geben Sie im zyklischen OB am Parameter REG_KEY der Anweisung MB_RED_SERVER den Namen des Lizenz-DB und den Namen der Zeichenfolge ein (z.B. Lizenz_DB.REG_KEY).
5. Geänderte Bausteine in die CPU laden Sie können den Registrierungsschlüssel während der Laufzeit eingeben. Es ist nicht nötig, die CPU von STOP in RUN zu versetzen.
6. Die Modbus/TCP-Kommunikation über die Anweisung MB_RED_SERVER ist jetzt für diese CPU lizenziert, das Ausgangsbit LICENSED ist TRUE.

Vorgehensweise zum Korrigieren von fehlender oder fehlerhafter Lizenzierung:

- Wenn Sie einen fehlerhaften Registrierungsschlüssel oder gar keinen Registrierungsschlüssel eingeben, blinkt die ERROR LED der CPU. Außerdem nimmt die CPU bei der S7-1200 auch noch einen zyklischen Eintrag in den Diagnosepuffer bezüglich der fehlenden Lizenz vor.

Diagnosepuffer		
Ereignisse		
<input checked="" type="checkbox"/> CPU-Zeitstempel berücksichtigt lokale PG/PC-Zeit		
Nr.	Datum und Uhrzeit	Ereignis
1	20.12.2018 12:30:33.459	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
2	20.12.2018 12:30:33.458	Kein gültiger License Key für das Funktionspaket
3	20.12.2018 12:30:27.453	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
4	20.12.2018 12:30:27.452	Kein gültiger License Key für das Funktionspaket
5	20.12.2018 12:30:21.445	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
6	20.12.2018 12:30:21.444	Kein gültiger License Key für das Funktionspaket
7	20.12.2018 12:30:15.439	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB
8	20.12.2018 12:30:15.438	Kein gültiger License Key für das Funktionspaket
9	20.12.2018 12:30:09.431	Bereichslängenfehler in FB 1086 - Bearbeitung wird fortgesetzt (keine OB

Bild 13-26 Diagnosepuffer

- Bei einem fehlenden oder fehlerhaften Registrierungsschlüssel verarbeitet die CPU die Modbus TCP-Kommunikation. Die CPU zeigt am Ausgang STATUS_x jedoch immer "W#16#0A90" (Kein gültiger License Key für Funktionspaket) an. Das Ausgangsbit LICENSED ist FALSE.

Eingangsparameter MB_HOLD_REG

Der Parameter MB_HOLD_REG ist ein Pointer auf einen Datenpuffer zum Speichern der Daten, auf die ein Modbus-Client Lese- oder Schreibzugriff hat. Sie können einen globalen Datenbaustein (D) oder einen Merker (M) als Speicherbereich verwenden:

- Der obere Grenzwert für die Anzahl von Adressen im Datenbaustein (D) ist von der maximalen Größe eines DB in Ihrer CPU abhängig.
- Der obere Grenzwert für die Anzahl von Merkern (M) ist abhängig vom maximalen Merkerbereich in Ihrer CPU.

Die folgenden Abbildungen zeigen einige Beispiele für die Zuordnung von Modbus-Adressen zum Halteregeister für die Modbus-Funktionen 3 (Mehrere WORD lesen), 6 (Ein WORD schreiben), 16 (Mehrere WORD schreiben) und 23 (Mehrere WORD schreiben und lesen).

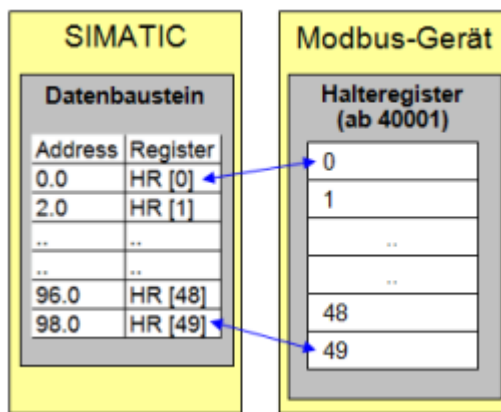


Bild 13-27 MB_HOLD_REG: Datenbaustein mit Offset 0

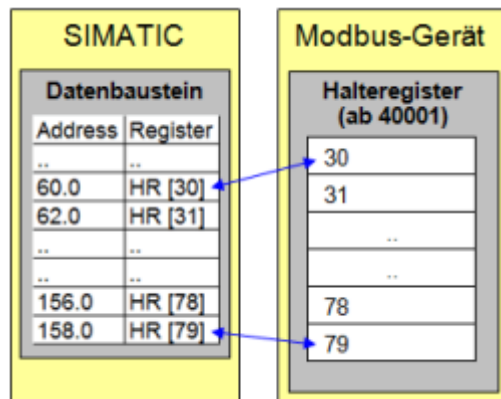


Bild 13-28 MB_HOLD_REG: Datenbaustein mit Offset 60

Data_Area_Array [1..8]: Informationen zur Verwendung der optionalen Parameter Data_Area_Array [1..8], finden Sie im Abschnitt "Parametrierung" oben.

Ausgangsparameter: ERROR_x, STATUS_x, RED_ERR_S7, RED_ERR_DEV und TOT_COM_ERR

Die CPU zeigt Fehlermeldungen an den Statusausgängen der Anweisung MB_RED_SERVER an:

Hinweis

Fehlerstatuscodes können im Programmmeditor als ganzzahlige Werte oder Hexadezimalwerte angezeigt werden:

1. Öffnen Sie den gewünschten Baustein im Programmiereditor.
2. Schalten Sie den Programmierstatus ein, indem Sie auf "Beobachten ein/aus" klicken. (Wenn Sie noch keine Online-Verbindung hergestellt haben, wird der Dialog "Online gehen" geöffnet. In diesem Dialog können Sie eine Online-Verbindung herstellen.)
3. Wählen Sie die Variable aus, die Sie beobachten möchten, und wählen Sie im Kontextmenü unter "Anzeigeformat" das gewünschte Anzeigeformat aus.

Parameter STATUS_x (allgemeine Statusinformationen):

STATUS (W#16#)	Beschreibung
0000	Anweisung fehlerfrei ausgeführt.
0001	Verbindung hergestellt.
0003	Verbindung beendet.
0A90	Die Anweisung MB_RED_SERVER ist nicht lizenziert. Weitere Informationen finden Sie im obigen Abschnitt "Lizenzierung".
0AFF	Die Verbindung ist nicht konfiguriert und wird nicht verwendet. Die Verbindung 0A muss konfiguriert werden.
7000	Kein Aufruf aktiv und keine Verbindung hergestellt (REQ = 0, DISCONNECT = 1).
7001	Erster Aufruf. Verbindungsaufbau angestoßen.
7002	Zwischenzeitlicher Aufruf. Verbindung wird aufgebaut.
7003	Verbindung wird beendet.
7005	Daten werden gesendet.
7006	Daten werden empfangen.

Parameter STATUS_x (Protokollfehler)

STATUS (W#16#)	Fehlercode in der Fehlermeldung von MB_RED_SERVER(B#16#)	Beschreibung
8380	-	Empfangenes Modbus-Telegramm hat falsches Format oder es wurden zu wenige Bytes empfangen.
8381	01	Funktionscode wird nicht unterstützt.
8382	03	Fehler in Datenlänge: <ul style="list-style-type: none"> • Ungültige Längenangabe im empfangenen Modbus-Telegramm • Die Telegrammlänge im Header des Modbus-Telegramms entspricht nicht der Anzahl der tatsächlich empfangenen Bytes. • Die Anzahl der Bytes im Header des Modbus-Telegramms entspricht nicht der Anzahl der tatsächlich empfangenen Bytes (Funktionen 15 und 16).

STATUS (W#16#)	Fehlercode in der Fehlermeldung von MB_RED_SERVER(B#16#)	Beschreibung
8383	02	Fehler in der Datenadresse oder Zugriff außerhalb des Adressbereichs des Haltereisters (Parameter MB_HOLD_REG). Weitere Informationen finden Sie im obigen Abschnitt "MB_HOLD_REG".
8384	03	Fehler im Datenwert (Funktion 05)
8385	03	Diagnosecode wird nicht unterstützt (nur bei Funktion 08).

Parameter STATUS_x (Parameterfehler)

STATUS (W#16#)	Beschreibung
80BB	Parameter ActiveEstablished hat einen ungültigen Wert. Nur passiver Verbindungsaufbau für Server zulässig (active_established = FALSE).
8187	Parameter MB_HOLD_REG hat einen ungültigen Pointer. Datenbereich ist zu klein.
8389	Ungültige Datenbereichsdefinition: <ul style="list-style-type: none"> • Ungültiger data_type-Wert • DB-Nummer ist ungültig oder nicht vorhanden: <ul style="list-style-type: none"> – Ungültiger db-Wert – DB-Nummer ist nicht vorhanden – DB-Nummer wird bereits von einem anderen Datenbereich verwendet – DB mit optimiertem Zugriff – DB befindet sich nicht im Arbeitsspeicher • Ungültiger length-Wert • Überschneidung von MODBUS-Adressbereichen, die zu demselben MODBUS-Datentyp gehören

Hinweis

Fehlercodes von intern verwendeten Kommunikationsanweisungen

Bei der Anweisung MB_RED_SERVER können, neben den in den Tabellen aufgeführten Fehlern, von den Kommunikationsanweisungen ("TCON", "TDISCON", "TSEND", "TRCV", "T_DIAG" und "T_RESET") verursachte Fehler, die von der Anweisung verwendet werden, auftreten.

Die Fehlercodes werden über den Instanz-Datenbaustein der Anweisung MB_RED_SERVER zugewiesen. Die Fehlercodes werden für die jeweilige Anweisung im Abschnitt "Static" der einzelnen Instanzen unter STATUS angezeigt.

Die Bedeutung der Fehlercodes ist in der Dokumentation der entsprechenden Kommunikationsanweisung verfügbar.

Hinweis**Kommunikationsfehler beim Senden oder Empfangen von Daten**

Wenn beim Senden oder Empfangen von Daten ein Kommunikationsfehler auftritt (80C4 (Temporärer Kommunikationsfehler. Die angegebene Verbindung wird temporär beendet.), 80C5 (Der entfernte Partner hat die Verbindung aktiv beendet.), 80A1 (Die angegebene Verbindung wurde beendet oder ist noch nicht aufgebaut.)), beendet die CPU die bestehende Verbindung.

Das bedeutet auch, dass Sie alle zurückgegebenen STATUS-Werte sehen können, wenn die Verbindung beendet wird, und dass der STATUS-Code, der den Abbruch der Verbindung verursacht hat, nur ausgegeben wird, wenn die Verbindung beendet wird.

Beispiel: Wenn beim Empfangen von Daten ein temporärer Kommunikationsfehler auftritt, wird zunächst der STATUS 7003 (ERROR = False) und dann 80C4 (ERROR = True) ausgegeben.

13.5.2.4 Beispiele für Modbus TCP**Beispiel: MB_SERVER für mehrere TCP-Verbindungen**

Sie können mehrere Verbindungen zum Modbus TCP-Server haben. Hierfür muss MB_SERVER für jede Verbindung unabhängig ausgeführt werden. Jede Verbindung muss einen unabhängigen Instanz-DB, eine Verbindungs-ID und einen IP-Port verwenden. Bei der S7-1200 ist nur je eine Verbindung pro IP-Port zulässig.

Um optimales Betriebsverhalten zu erzielen, muss MB_SERVER in jedem Programmzyklus für jede Verbindung ausgeführt werden.

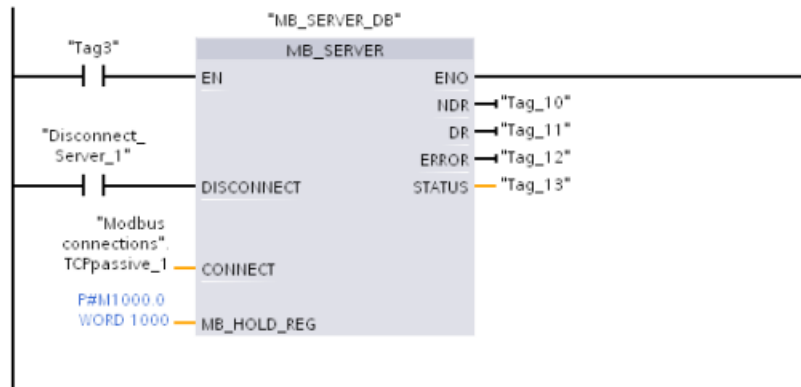
Der Parameter CONNECT verwendet den Systemdatentyp TCON_IP_V4. Im Beispiel befinden sich diese Datenstrukturen im DB "Modbus-Verbindungen". Der DB "Modbus-Verbindungen" enthält zwei TCP_IP_V4-Strukturen: "TCPpassiv_1" (für Verbindung 1) und "TCP_passiv_2" (für Verbindung 2). Bei der in den Netzwerkkommentaren beschriebenen ID der Verbindungseigenschaften und bei LocalPort handelt es sich um in der Datenstruktur CONNECT gespeicherte Datenelemente.

Die TCP_IP_V4 CONNECT-Daten enthalten auch eine IP-Adresse im ADDR-Array RemoteAddress. Die IP-Adresszuweisungen in TCPpassiv_1 und TCP_passiv_2 wirken sich nicht auf den Aufbau der TCP-Serververbindungen aus, bestimmen jedoch, welche Modbus TCP-Clients über die Verbindungen mit jedem MB_SERVER kommunizieren dürfen. MB_SERVER überwacht passiv auf eine Modbus-Client-Meldung und vergleicht die eingehende IP-Adresse der Meldung mit der im entsprechenden ADDR-Array RemoteAddress gespeicherten IP-Adresse.

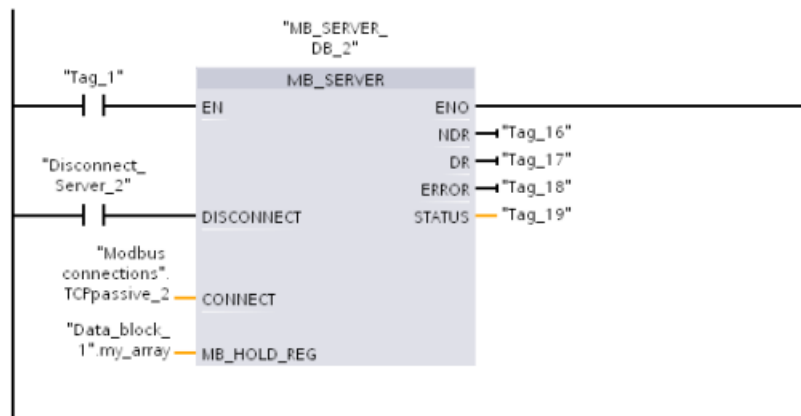
Für die beiden MB_SERVER-Anweisungen sind drei Varianten von IP-Adressen möglich:

- **IP-Adresse = 0.0.0.0**
Jede MB_SERVER-Anweisung reagiert auf alle Modbus TCP-Clients mit beliebiger IP-Adresse.
- **IP-Adresse = Gleiche IP-Adresse in TCPpassiv_1 und TCPpassiv_2**
Beide MB_SERVER-Verbindungen reagieren nur auf Modbus-Clients, die von dieser IP-Adresse stammen.
- **IP-Adresse = Unterschiedliche IP-Adresse in TCPpassiv_1 und TCPpassiv_2**
Jede MB_SERVER-Anweisung reagiert nur auf Modbus-Clients, die von der IP-Adresse stammen, die in ihren jeweiligen TCP_IP_V4-Daten gespeichert ist.

Netzwerk 1: Verbindung 1, Instanz-DB = "MB_SERVER_DB", in "Modbus-Verbindungen.TCPpassiv_1" (ID = 1 und LocalPort = 502)



Netzwerk 2: Verbindung 2, Instanz-DB = "MB_SERVER_DB_1", in "Modbus-Verbindungen.TCPpassiv_2" (ID = 2 und LocalPort = 503)



Beispiel: MB_CLIENT 1: Mehrere Anforderungen mit gemeinsamer TCP-Verbindung

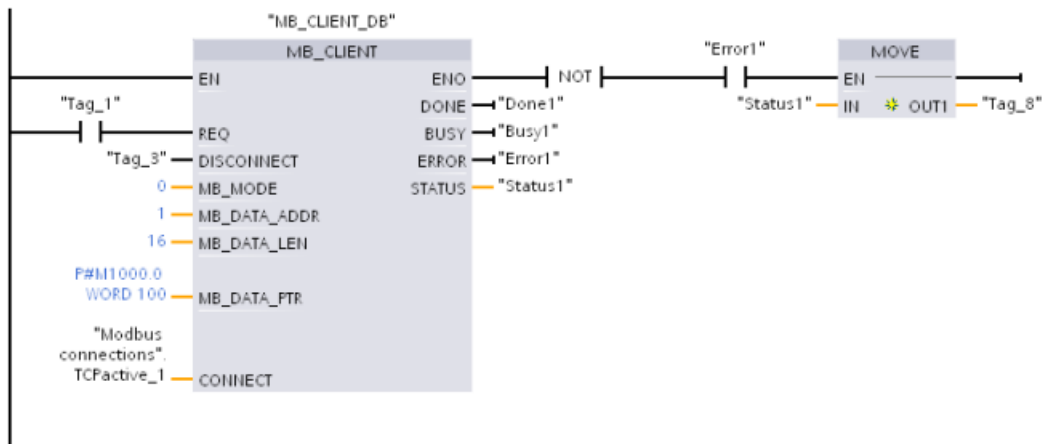
Mehrere Modbus-Clientanforderungen können über die gleiche Verbindung gesendet werden. Hierfür verwenden Sie den gleichen Instanz-DB, die gleiche Verbindungs-ID und Portnummer.

Weil beide MB_CLIENT-Boxen die gleiche TCON_IP_v4-Datenstruktur am Parameter CONNECT ("Modbus_Verbindungen".TCPaktiv_1) verwenden, sind die Verbindungs-ID, Portnummer und IP-Adresse identisch. Die IP-Adressdaten von CONNECT weisen die IP-Adresse des Ziel-Modbus-TCP-Servers zu.

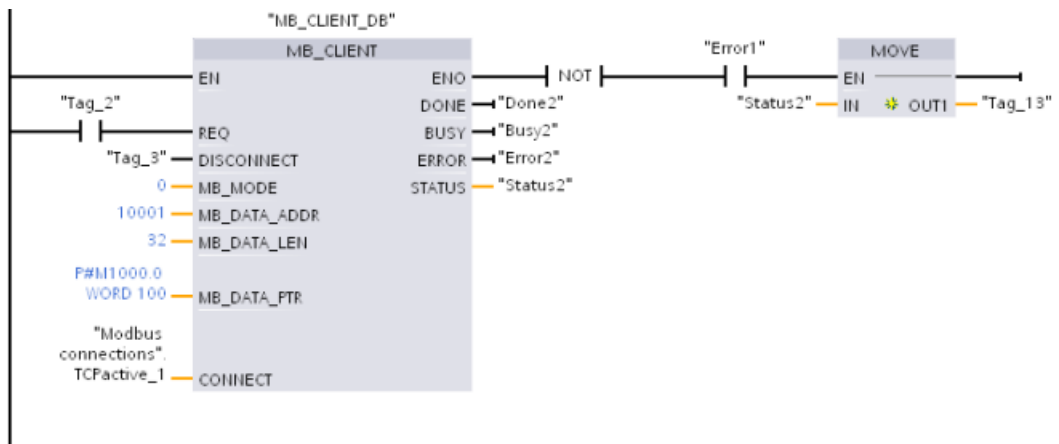
Es kann jeweils nur eine Anweisung MB_CLIENT aktiv sein. Nachdem ein Client seine Ausführung beendet hat, kann der nächste Client mit der Ausführung beginnen. Ihre Programmlogik ist für die Reihenfolge der Ausführung verantwortlich. Das Beispiel zeigt beide Clients, die dezentrale Daten aus einem einzelnen Modbus-Client lesen und die Daten an die CPU des Modbus-Clients übertragen (Speicherbereich der Merker mit Beginn an M1000.0). Ein ausgegebener Fehler wird erfasst (optionale Funktion).

Netzwerk 1: Modbus-Funktion 1 - 16 Ausgangsbits von einem Modbus TCP-Server mit der in "Modbus-Verbindungen".TCPaktiv_1 zugewiesenen IP-Adresse lesen.

13.5 Modbus-Kommunikation



Netzwerk 2: Modbus-Funktion 2 - 32 Eingangsbits von einem Modbus TCP-Server mit der in "Modbus-Verbindungen".TCPaktiv_1 zugewiesenen IP-Adresse lesen.



Beispiel: MB_CLIENT 2: Mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen

Mehrere Modbus-TCP-Client-Anforderungen können über verschiedene Verbindungen gesendet werden. Hierfür müssen unterschiedliche Instanz-DBs und Verbindungs-IDs verwendet werden.

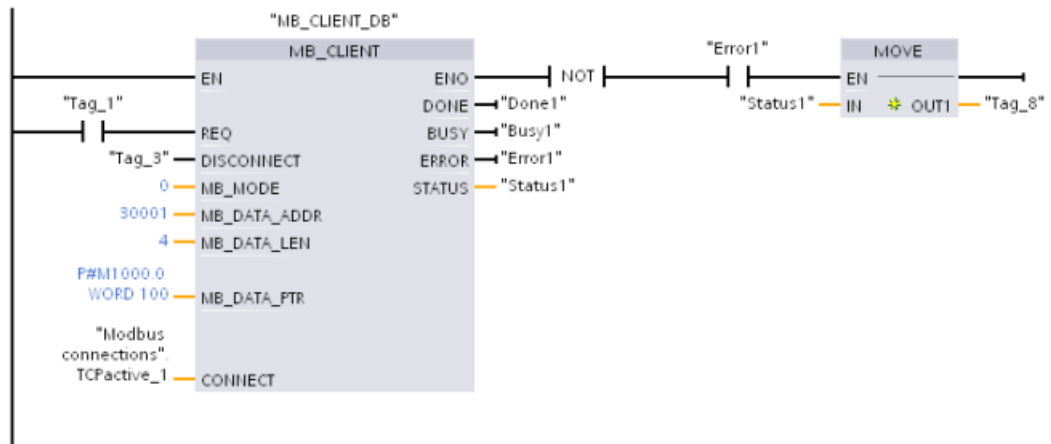
Die RemotePort-Nummer (IP-Port) muss unterschiedlich sein, wenn die Verbindungen zum selben Modbus-Server aufgebaut werden. Bei Verbindungen zu unterschiedlichen Servern gibt es keine Einschränkungen hinsichtlich der IP-Portnummer.

Das Beispiel zeigt zwei Modbus-TCP-Clients, die dezentrale Daten von zwei unterschiedlichen Modbus-TCP-Servern in denselben lokalen CPU-Speicherbereich mit Beginn an Adresse M1000.0 übertragen. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

Netzwerk 1: Modbus-Funktion 4 - Wörter aus dem Prozessabbild der Eingänge von einem Modbus-TCP-Server lesen

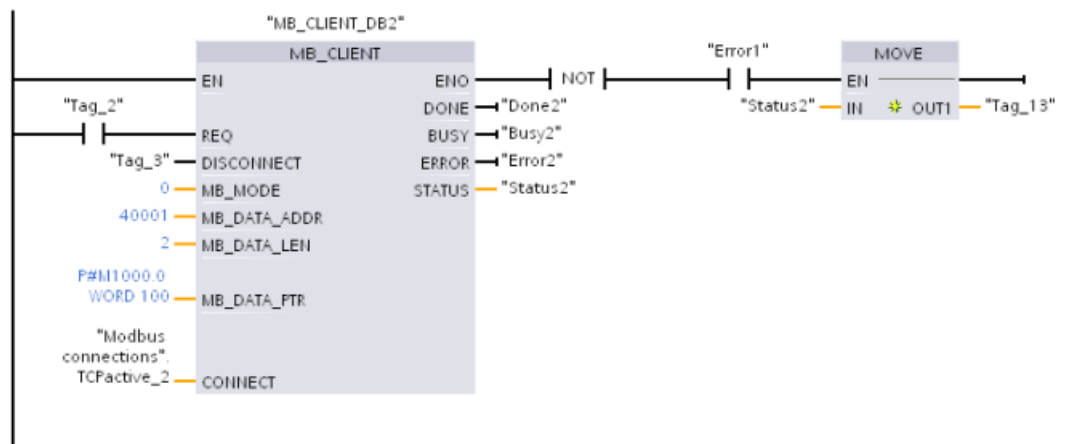
Parameter CONNECT = "Modbus-Verbindungen".TCPaktiv_1: Verbindungs-ID = 1,

RemoteAddress = 192.168.2.241, RemotePort = 502



Netzwerk 2: Modbus-Funktion 3 - Wörter aus dem Haltereister von einem Modbus-TCP-Server lesen

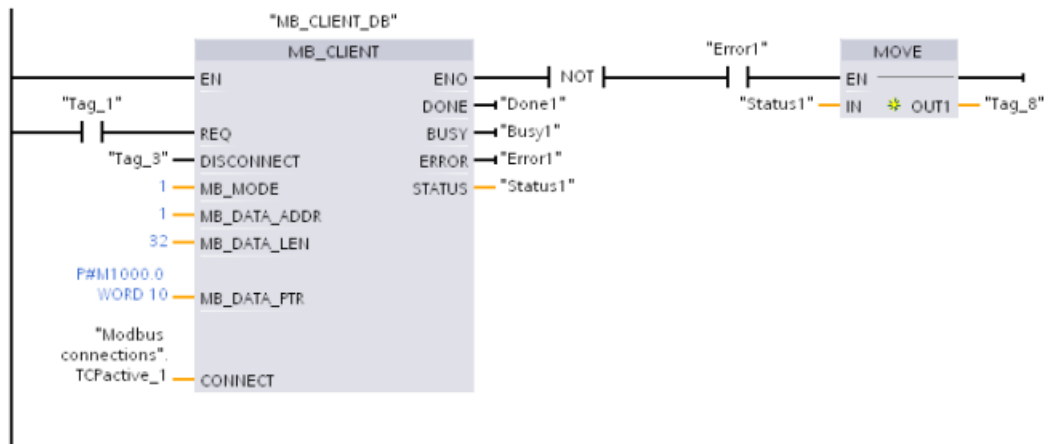
Parameter CONNECT = "Modbus-Verbindungen".TCPaktiv_2: Verbindungs-ID = 2,
RemoteAddress = 192.168.2.242, **RemotePort** = 502



Beispiel: MB_CLIENT 3: Schreibenanforderung für das Prozessabbild der Ausgänge

Dieses Beispiel zeigt eine Modbus-Client-Anforderung, die Bitdaten aus dem lokalen CPU-Speicher (mit Beginn an M1000.0) zu einem dezentralen Modbus-TCP-Server überträgt.

Netzwerk 1: Modbus-Funktion 15 - Ausgangsbits in einen Modbus-Server schreiben

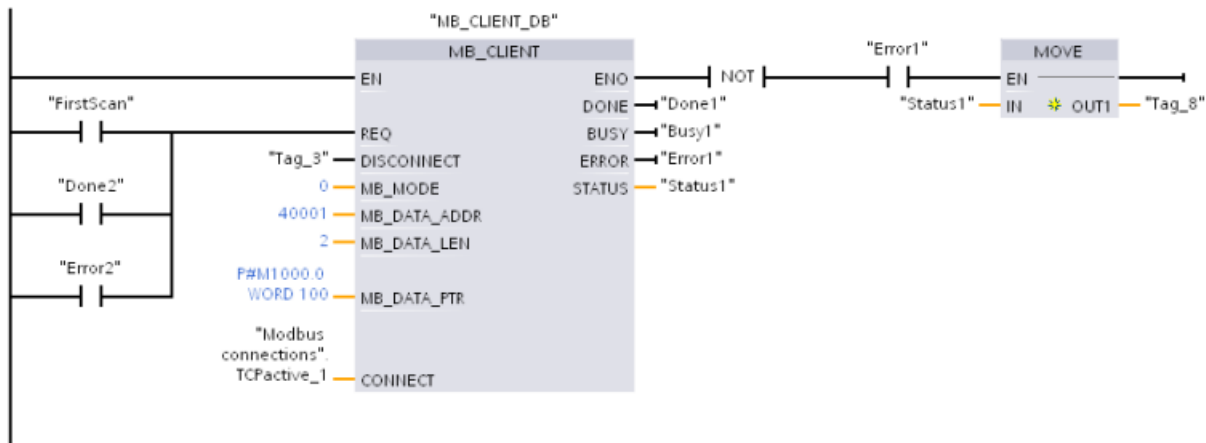


Beispiel: MB_CLIENT 4: Mehrere Anforderungen koordinieren

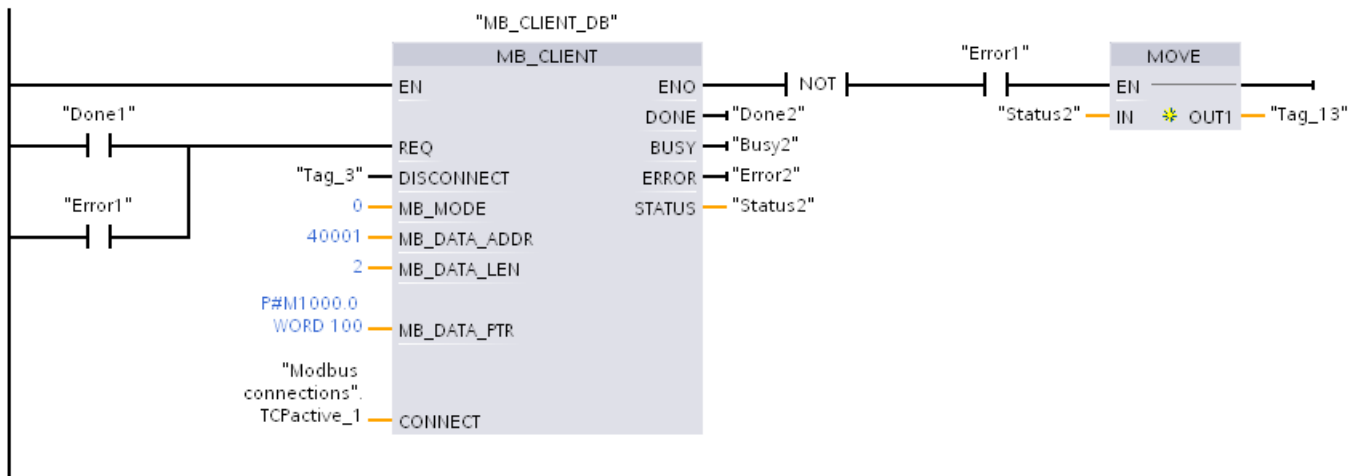
Sie müssen sicherstellen, dass die Ausführung jeder einzelnen Modbus TCP-Anforderung abgeschlossen wird. Die Ausführungsreihenfolge muss von Ihrer Programmlogik gesteuert werden. Das folgende Beispiel zeigt, wie die Ausführungsreihenfolge über die Ausgänge der ersten und der zweiten Clientanforderung gesteuert werden kann.

Das Beispiel zeigt beide Clients, die dieselben CONNECT-Verbindungsdaten verwenden (zu unterschiedlichen Zeiten). Die Clients übertragen Halteregisterdaten von demselben dezentralen Modbus-TCP-Server in dieselbe Adresse im Speicherbereich der Merker der lokalen CPU. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

Netzwerk 1: Modbus-Funktion 3 - Wörter im Halteregister des Modbus-TCP-Servers lesen



Netzwerk 2: Modbus-Funktion 3 - Wörter im Halteregister des Modbus-TCP-Servers lesen



13.5.3 Modbus RTU

13.5.3.1 Übersicht

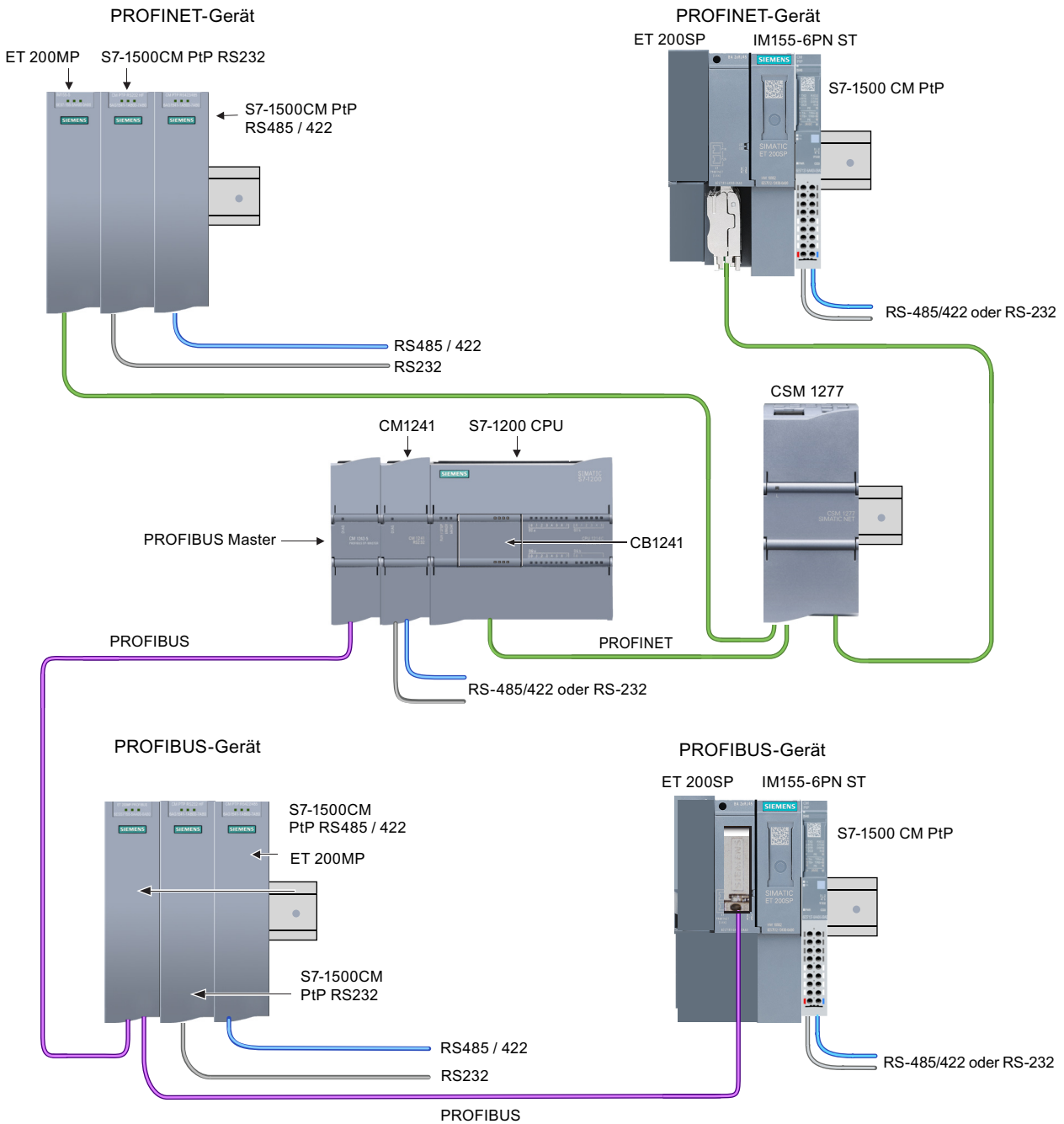
Ab Version V4.1 der S7-1200 CPU in Verbindung mit STEP 7 V13 SP1 wird die Modbus RTU-Fähigkeit der CPU zur Nutzung eines dezentralen PROFINET- oder PROFIBUS-Peripheriebaugruppenträgers für die Kommunikation mit verschiedenen Geräten (RFID-Lesegeräten, GPS und anderen) erweitert:

- PROFINET (Seite 598): Die Ethernet-Schnittstelle der S7-1200 CPU wird mit einem PROFINET Schnittstellenmodul verbunden. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.
- PROFIBUS (Seite 782): Ein PROFIBUS-Kommunikationsmodul wird an der linken Seite des Baugruppenträgers mit der S7-1200 CPU eingesteckt. Das PROFIBUS-Kommunikationsmodul wird mit einem Baugruppenträger verbunden, der ein PROFIBUS-Schnittstellenmodul enthält. Über PtP-Kommunikationsmodule im Baugruppenträger mit dem Schnittstellenmodul ist dann die serielle Kommunikation zu den PtP-Geräten möglich.

Die S7-1200 unterstützt daher zwei Sätze PtP-Anweisungen:

- Ältere Modbus RTU-Anweisungen (Seite 1166): Diese Modbus RTU-Anweisungen wurden vor der Version V4.0 der S7-1200 benutzt und ermöglichen nur die serielle Kommunikation mit Hilfe eines Kommunikationsmoduls CM 1241 oder eines Kommunikationsboards CB 1241.
- Modbus RTU-Anweisungen (Seite 1092): Diese Modbus RTU-Anweisungen stellen alle Funktionen der alten Anweisungen bereit und zusätzlich die Möglichkeit zur Verbindung mit der dezentralen PROFINET- und PROFIBUS-Peripherie. Diese Modbus RTU-Anweisungen ermöglichen das Konfigurieren der Kommunikation zwischen den PtP-Kommunikationsmodulen im Baugruppenträger mit der dezentralen Peripherie und den PtP-Geräten. S7-1200 CM 1241-Module benötigen mindestens die Firmwareversion V2.1, um diese Modbus RTU-Anweisungen verwenden zu können.

13.5 Modbus-Kommunikation



Hinweis

Ab der Version V4.1 der S7-1200 können PtP-Anweisungen für alle Arten der PtP-Kommunikation verwendet werden: seriell, seriell über PROFINET und seriell über PROFIBUS. STEP 7 stellt die alten PtP-Anweisungen nur zur Unterstützung vorhandener Programme bereit. Die alten Anweisungen funktionieren jedoch noch mit V4.1-CPU's sowie mit CPU-Versionen 4.0 und früher. Es ist nicht nötig, ältere Programme auf die neuen Anweisungen zu konvertieren.

13.5.3.2 Version der Modbus RTU-Anweisungen auswählen

Die folgenden Versionen der Modbus RTU-Anweisungen sind in STEP 7 verfügbar:

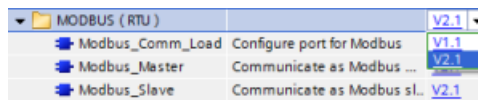
- Version 3.4: Kompatibel mit CPUs ab V4.0 und CMs ab V2.1
- Ab Version 4.3: Kompatibel mit CPUs ab V4.2 und CMs ab V2.1

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen. Zusätzlich unterstützt die S7-1200 CPU weiterhin die alten Versionen der Modbus RTU-Anweisungen (Seite 1166).

Sie können nicht beide Versionen der Anweisungen bei demselben Modul verwenden, Sie können jedoch für zwei verschiedene Module unterschiedliche Versionen der Anwendungen einsetzen.

Rufen Sie in der Taskcard "Anweisung" die MODBUS (RTU)-Anweisungen in der Gruppe Kommunikationsprozessor auf.

Um die Version einer Modbus RTU-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.



Wenn Sie eine Modbus RTU-Anweisung über das Anweisungsverzeichnis in Ihr Programm einfügen, wird in der Projektnavigation eine neue FB-Instanz angelegt. Die neue FB-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer Modbus RTU-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmmeditor angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines Modbus RTU-FBs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der Modbus RTU-Anweisung anzuzeigen.

13.5.3.3 Maximale Anzahl unterstützter Modbus-Slaves

Die Modbus-Adressierung unterstützt maximal 247 Slaves (Slave-Nummern 1 bis 247). Jedes Segment im Modbus-Netzwerk kann, je nach den Lade- und Antriebsfunktionen der RS485-Schnittstelle, maximal 32 Geräte haben. Wenn Sie den Grenzwert von 32 Geräten erreichen, müssen Sie für die Erweiterung zum nächsten Segment einen Repeater verwenden. Sie benötigen sieben Repeater, um die für RS485 an einen Master angeschlossenen 247 Slaves zu unterstützen.

Repeater von Siemens funktionieren nur mit PROFIBUS. Ihre Funktion liegt in der Überwachung der PROFIBUS-Token-Weitergabe. Repeater von Siemens können Sie nicht mit anderen Protokollen einsetzen. Sie benötigen daher für Modbus Repeater von anderen Anbietern.

Für Modbus-Timeouts sind standardmäßig hohe Werte eingestellt. Die Verwendung von Repeatern verursacht deshalb kein Problem hinsichtlich der Zeitverzögerung. Der Modbus-

13.5 Modbus-Kommunikation

Master arbeitet unabhängig davon, ob ein Slave langsam reagiert oder ob mehrere Repeater die Antwort verzögern.

13.5.3.4 Modbus RTU-Anweisungen

Anweisung Modbus_Comm_Load (SIPLUS I/O oder Port am PtP-Modul für Modbus RTU konfigurieren)

Tabelle 13-76 Anweisung Modbus_Comm_Load

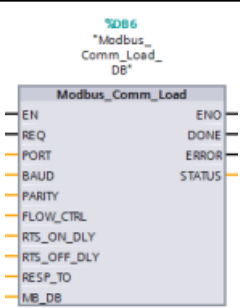
KOP/FUP	SCL	Beschreibung
	<pre>"Modbus_Comm_Load_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_udint_in_, PARITY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung Modbus_Comm_Load konfiguriert SIPLUS I/O oder einen PtP-Port für die Kommunikation über das Modbus-RTU-Protokoll.</p> <p>Hardwareoptionen des Modbus RTU-Ports: Einbau von max. drei CMs (RS485 oder RS232), plus eines CB (R4845).</p> <p>Optionen für Modbus RTU SIPLUS I/O: Installieren Sie die ET 200MP S7-1500CM PtP (RS485 / 422 oder RS232) oder ET 200SP S7-1500 CM PtP (RS485 / 422 oder RS232)</p> <p>Wenn Sie die Anweisung Modbus_Comm_Load in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen.</p>

Tabelle 13-77 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
EN	IN	Bool
REQ	IN	Bool
PORT	IN	Port
BAUD	IN	UDInt
PARITY	IN	UInt

Hinweis: Die Anweisung Modbus_Comm_Load für Modbus RTU nutzt die Anweisungen PDREC und WRREC, um das PTP-Modul zu initialisieren. Die Anweisung RDREC/WRREC arbeitet jedoch asynchron, was bedeutet, dass mehrere Zyklen erforderlich sind, um die Anweisung durchzuführen. Deshalb muss der EN-Parameter von Modbus_Comm_Load bis zum Abschluss der Anweisung RDREC/WRREC wahr bleiben.

Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Nur Version 2.0)

Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen.

Auswahl der Baudrate:
300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Alle anderen Werte sind ungültig.

Auswahl der Parität:

- 0 – Keine
- 1 – Ungerade
- 2 – Gerade

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
FLOW_CTRL ¹	IN	UInt	Auswahl der Flusskontrolle: <ul style="list-style-type: none"> • 0 – (Standard) Keine Flusskontrolle • 1 – Hardware-Flusskontrolle mit RTS immer EIN (gilt nicht bei RS485-Ports) • 2 – Hardware-Flusskontrolle mit RTS geschaltet
RTS_ON_DLY ¹	IN	UInt	Auswahl RTS-Einschaltverzögerung: <ul style="list-style-type: none"> • 0 – (Standard) Keine Verzögerung von "RTS aktiv", bis das erste Zeichen der Meldung gesendet wird. • 1 bis 65535 – Verzögerung in Millisekunden von "RTS aktiv", bis das erste Zeichen der Meldung gesendet wird (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl für FLOW_CTRL angewendet werden.
RTS_OFF_DLY ¹	IN	UInt	Auswahl RTS-Ausschaltverzögerung: <ul style="list-style-type: none"> • 0 – (Standard) Keine Verzögerung nach der Übertragung des letzten Zeichens bis "RTS inaktiv" • 1 bis 65535 – Verzögerung in Millisekunden von der Übertragung des letzten Zeichens bis "RTS inaktiv" (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl für FLOW_CTRL angewendet werden.
RESP_TO ¹	IN	UInt	Antwort-Timeout: Zeit in Millisekunden, die der Modbus_Master auf eine Antwort vom Slave wartet. Wenn der Slave während dieses Zeitraums nicht antwortet, wiederholt der Modbus_Master die Anforderung oder beendet die Anforderung mit einem Fehler, wenn die angegebene Anzahl von Wiederholungen gesendet wurde. 5 ms bis 65535 ms (Standardwert = 1000 ms).
MB_DB	IN	Variant	Ein Verweis auf den Instanz-Datenbaustein der Anweisungen Modbus_Master oder Modbus_Slave. Nachdem Sie Modbus_Master oder Modbus_Slave in Ihr Programm eingefügt haben, ist die DB-Kennung in der Parameter-Klappliste am Anschluss der MB_DB-Box verfügbar.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Nur Version 2.0)
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

¹ Optionale Parameter für Modbus_Comm_Load (V 2.x oder höher). Klicken Sie auf den Pfeil im unteren Bereich der KOP/FUP-Box, um die Box zu erweitern und diese Parameter aufzunehmen.

Modbus_Comm_Load wird ausgeführt, um einen Port für das Modbus-RTU-Protokoll zu konfigurieren. Nachdem ein Port für das Modbus-RTU-Protokoll konfiguriert ist, kann er nur von den Anweisungen Modbus_Master oder Modbus_Slave verwendet werden.

Für die Konfiguration jedes Kommunikationsports, der für die Modbus-Kommunikation eingesetzt wird, muss eine Ausführung von Modbus_Comm_Load genutzt werden. Sie müssen jedem Port, den Sie nutzen, einen eindeutigen Instanz-DB Modbus_Comm_Load zuweisen. Sie können bis zu drei Kommunikationsmodule (RS232 oder RS485) und ein Kommunikationsboard (RS485) in die CPU einbauen. Sie können Modbus_Comm_Load aus einem Anlauf-OB aufrufen und einmal ausführen, oder Sie können den Aufruf für die einmalige Ausführung mit Hilfe des

13.5 Modbus-Kommunikation

Systemmerkers für den ersten Zyklus (Seite 92) initiieren. Führen Sie `Modbus_Comm_Load` nur dann erneut aus, wenn Kommunikationsparameter wie Baudrate oder Parität geändert werden müssen.

Wenn Sie die Modbus-Bibliothek mit einem Modul in einem dezentralen Baugruppenträger einsetzen, muss die Anweisung `Modbus_Comm_Load` in einem zyklischen Interruptprogramm ausgeführt werden (z. B. einmal pro Sekunde oder alle 10 Sekunden). Wenn die Spannungsversorgung des dezentralen Baugruppenträgers unterbrochen oder das Modul gezogen wird, wird bei Wiederkehr des Modulbetriebs nur der festgelegte `HWConfig`-Parameter an das PtP-Modul gesendet. Alle vom `Modbus_Master-Timeout` ausgelösten Anforderungen und der `Modbus_Slave` gehen in den stillen Modus (keine Reaktion auf Meldungen). Durch die zyklische Ausführung der Anweisung `Modbus_Comm_Load` werden diese Probleme behoben.

Wenn Sie `Modbus_Master` oder `Modbus_Slave` in Ihr Programm einfügen, wird der Anweisung ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein wird referenziert, wenn Sie in der Anweisung `Modbus_Comm_Load` den Parameter `MB_DB` angeben.

Variablen des Instanz-Datenbausteins (DB) von `Modbus_Comm_Load`

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von `Modbus_Comm_Load`, die in Ihrem Programm verwendet werden können:

Tabelle 13-78 Statische Variablen im Instanz-DB von `Modbus_Comm_Load`

Variable	Datentyp	Standardwert	Beschreibung
<code>ICHAR_GAP</code>	Word	0	Maximale Zeichenverzögerungszeit zwischen den Zeichen. Dieser Parameter wird in Millisekunden angegeben und erhöht den erwarteten Zeitraum zwischen den empfangenen Zeichen. Die entsprechende Anzahl von Bitzeiten für diesen Parameter wird zum Modbus-Standardwert von 35 Bitzeiten (3,5 Zeichenzeiten) addiert.
<code>RETRIES</code>	Word	2	Anzahl von Wiederholungen, die der Master ausführt, bevor der Fehlercode <code>0x80C8</code> für "Keine Antwort" zurückgegeben wird.
<code>EN_SUPPLY_VOLT</code>	Bool	0	Diagnose für Ausfall Versorgungsspannung L+ aktivieren.
<code>MODE</code>	USInt	0	Betriebsart Folgende Betriebsarten sind gültig: <ul style="list-style-type: none"> • 0 = Vollduplex (RS232) • 1 = Vollduplex (RS422), Vierdrahtbetrieb (Punkt-zu-Punkt) • 2 = Vollduplex (RS422), Vierdrahtbetrieb (Multipoint-Master, CM PtP (ET 200SP)) • 3 = Vollduplex (RS422), Vierdrahtbetrieb (Multipoint-Slave, CM PtP (ET 200SP)) • 4 = Halbduplex (RS485), Zweidrahtbetrieb (siehe Hinweis unten)

Variable	Datentyp	Standardwert	Beschreibung
LINE_PRE	USInt	0	Ausgangszustand Empfangsleitung Folgende Ausgangszustände sind gültig: <ul style="list-style-type: none"> • 0 = "Kein" Ausgangszustand (siehe Hinweis unten) • 1 = Signal R(A) = 5 V DC, Signal R(B) = 0 V DC (Drahtbruchererkennung): Drahtbruchererkennung ist in diesem Ausgangszustand möglich. Kann nur ausgewählt werden bei: "Vollduplex (RS422), Vierdrahtbetrieb (Punkt-zu-Punkt-Verbindung)" und "Vollduplex (RS422), Vierdrahtbetrieb (Multipoint-Slave)". • 2 = Signal R(A) = 0 V DC, Signal R(B) = 5 V DC: Diese Standardeinstellung entspricht dem Ruhezustand (kein aktiver Sendevorgang). In diesem Ausgangszustand ist keine Drahtbruchererkennung möglich.
BRK_DET	USInt	0	Drahtbruchererkennung Die folgenden Einstellungen sind gültig: <ul style="list-style-type: none"> • 0 = Drahtbruchererkennung deaktiviert • 1 = Drahtbruchererkennung aktiviert
EN_DIAG_ALARM	Bool	0	Diagnosealarm aktivieren: <ul style="list-style-type: none"> • 0 = nicht aktiviert • 1 = aktiviert
STOP_BITS	USInt	1	Anzahl Stoppbits: <ul style="list-style-type: none"> • 1 = 1 Stoppbit • 2 = 2 Stoppbits • 0, 3 bis 255 = reserviert

Hinweis

Erforderliche Einstellung für den Einsatz von PROFIBUS-Kabeln mit dem CM 1241 für RS485

Tabelle 13-79 Bedingungs-codes der Ausführung von Modbus_Comm_Load ¹

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
8180	Ungültiger Wert der Port-ID (falscher Port/falsche Hardwareerkennung des Kommunikationsmoduls)
8181	Ungültiger Wert für die Baudrate.
8182	Ungültiger Wert für die Parität.
8183	Ungültiger Wert für die Flusskontrolle.
8184	Ungültiger Wert für die Timeout-Zeit der Antwort (Antwort-Timeout ist kleiner als der Mindestwert von 5 ms)
8185	Der Parameter MB_DB ist kein Instanz-Datenbaustein einer Anweisung Modbus_Master oder Modbus_Slave.

¹ Neben den oben aufgeführten Fehlern der Anweisung Modbus_Comm_Load können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

Anweisung Modbus_Master (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Master kommunizieren)

Tabelle 13-80 Anweisung Modbus_Master

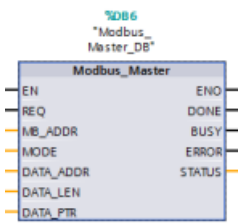
KOP/FUP	SCL	Beschreibung
	<pre>"Modbus_Master_DB" (REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	<p>Die Anweisung Modbus_Master kommuniziert als Modbus-Master über einen Port, der von einer vorherigen Ausführung der Anweisung Modbus_Comm_Load konfiguriert wurde. Wenn Sie die Anweisung Modbus_Master in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein Modbus_Master wird verwendet, wenn Sie in der Anweisung Modbus_Comm_Load den Parameter MB_DB angeben.</p>

Tabelle 13-81 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool 0 = Keine Anforderung 1 = Anforderung, Daten an den Modbus-Slave zu senden
MB_ADDR	IN	V1.0: USInt V2.0: UInt Modbus-RTU-Stationsadresse: Standard-Adressierungsbereich (1 bis 247) Erweiterter Adressierungsbereich (1 bis 65535) Der Wert 0 ist für den Broadcast einer Meldung an alle Modbus-Slaves reserviert. Für den Broadcast werden nur die Modbus-Funktionscodes 05, 06, 15 und 16 unterstützt.
MODE	IN	USInt Auswahl Modus: Gibt die Art der Anforderung an (Lesen, Schreiben oder Diagnose). Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_ADDR	IN	UDInt Anfangsadresse im Slave: Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll. Die gültigen Adressen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_LEN	IN	UInt Datenlänge: Gibt die Anzahl der Bits oder Wörter an, auf die diese Anforderung zugreifen soll. Die gültigen Längen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_PTR	IN_OUT	Variant Daten-Pointer: Zeigt auf die M- oder DB-Adresse (nicht optimierter DB) für die zu schreibenden oder zu lesenden Daten.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool • 0 - Kein Modbus_Master-Vorgang in Bearbeitung • 1 - Modbus_Master-Vorgang in Bearbeitung
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung

Regeln für die Modbus_Master-Kommunikation

- Modbus_Comm_Load muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung Modbus_Master mit diesem Port kommunizieren kann.
- Wenn ein Port zum Initiieren von Modbus-Masteranforderungen verwendet werden soll, darf dieser Port nicht von Modbus_Slave verwendet werden. Eine oder mehrere Instanzen von Modbus_Master können mit diesem Port verwendet werden, jedoch müssen alle Ausführungen von Modbus_Master den gleichen Instanz-DB Modbus_Master für den Port verwenden.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss die Anweisung Modbus_Master auf abgeschlossene Sende- und Empfangsvorgänge abfragen.
- Rufen Sie die Ausführung von Modbus_Master für einen bestimmten Port immer aus einem Programmzyklus-OB auf. Modbus_Master-Anweisungen können nur in jeweils einem Programmzyklus oder in jeweils einer zyklischen/zeitverzögerten Ausführungsschicht ausgeführt werden. Sie dürfen nicht in beiden Prioritätsschichten der Ausführung bearbeitet werden. Die Vorrangunterbrechung einer Modbus_Master-Anweisung durch eine andere Modbus_Master-Anweisung in einer Ausführungsschicht mit höherer Priorität führt zu nicht ordnungsgemäßem Betrieb. Die Modbus_Master-Anweisungen dürfen nicht in Anlauf-, Diagnose- oder Zeitfehlerausführungsschichten bearbeitet werden.
- Nachdem eine Modbus_Master-Anweisung eine Übertragung ausgelöst hat, muss diese Instanz mit aktiviertem Eingang EN fortlaufend ausgeführt werden, bis der Zustand DONE = 1 oder ERROR = 1 ausgegeben wird. Eine bestimmte Modbus_Master-Instanz wird als aktiv betrachtet, bis eines dieser beiden Ereignisse auftritt. Während die ursprüngliche Instanz aktiv ist, führt jeder Aufruf einer weiteren Instanz mit aktiviertem Eingang REQ zu einem Fehler. Wenn die fortlaufende Ausführung der ursprünglichen Instanz stoppt, bleibt der Anforderungszustand für die von der statischen Variable "Blocked_Proc_Timeout" angegebene Zeitdauer aktiv. Nach Ablauf dieser Zeitdauer wird die nächste aufgerufene Modbus_Master-Anweisung mit aktiviertem Eingang REQ zur aktiven Instanz. Dies verhindert, dass eine einzige Modbus_Master-Anweisung ein Monopol innehat oder den Zugriff auf den Port sperrt. Wenn die ursprüngliche aktive Instanz nicht während des von der statischen Variable "Blocked_Proc_Timeout" angegebenen Zeitraums aktiviert wird, löscht die nächste Ausführung von dieser Instanz (ohne gesetzten Eingang REQ) den aktiven Zustand. Falls REQ gesetzt ist, löst diese Ausführung eine neue Modbus_Master-Anforderung aus, so als wenn keine andere Instanz aktiv wäre.

Parameter REQ

0 = Keine Anforderung; 1 = Anforderung, Daten an den Modbus-Slave zu senden

Sie können diesen Eingang über einen pegel- oder flankengesteuerten Kontakt steuern. Immer wenn dieser Eingang aktiviert wird, wird eine Zustandsmaschine gestartet, um sicherzustellen, dass eine andere Anweisung Modbus_Master, die denselben Instanz-DB verwendet, erst dann eine Anforderung abgeben kann, wenn die aktuelle Anforderung abgearbeitet ist. Alle anderen Eingangszustände werden erfasst und intern für die aktuelle Anforderung gespeichert, bis die Antwort empfangen oder ein Fehler erkannt wird.

Wird dieselbe Instanz von Modbus_Master erneut mit Eingang REQ = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Übertragungen durchgeführt. Wenn die Anforderung jedoch abgearbeitet ist, wird zu dem Zeitpunkt, zu dem

Modbus_Master erneut mit REQ-Eingang = 1 ausgeführt wird, eine neue Anforderung abgegeben.

Über die Parameter DATA_ADDR und MODE wählen Sie den Modbus-Funktionstyp aus.

DATA_ADDR (Modbus-Anfangsadresse im Slave): Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll.

Die Anweisung Modbus_Master nutzt statt eines Funktionscodeeingangs einen Eingang MODE. Die Kombination aus MODE und Modbus-Adresse legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MODE, dem Modbus-Funktionscode und dem Modbus-Adressbereich.

Tabelle 13-82 Modbus-Funktionen

MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	Modbus-Adresse
0	01	1 bis 2000 1 bis 1992 ¹	Ausgangsbits lesen: 1 bis (1992 oder 2000) Bit pro Anforderung	1 bis 9999
0	02	1 bis 2000 1 bis 1992 ¹	Eingangsbits lesen: 1 bis (1992 oder 2000) Bit pro Anforderung	10001 bis 19999
0	03	1 bis 125 1 bis 124 ¹	Halteregister lesen: 1 bis (124 oder 125) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
0	04	1 bis 125 1 bis 124 ¹	Eingangswörter lesen: 1 bis (124 oder 125) Bits pro Anforderung	30001 bis 39999
104	04	1 bis 125 1 bis 124 ¹	Eingangswörter lesen: 1 bis (124 oder 125) Bits pro Anforderung	00000 bis 65535
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
1	15	2 bis 1968 2 bis 1960 ¹	Mehrere Ausgangsbits schreiben: 2 bis (1960 oder 1968) Bit pro Anforderung	1 bis 9999
1	16	2 bis 123 2 bis 122 ¹	Mehrere Halteregister schreiben: 2 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
2	15	1 bis 1968 2 bis 1960 ¹	Ein oder mehrere Ausgangsbits schreiben: 1 bis (1960 oder 1968) Bit pro Anforderung	1 bis 9999
2	16	1 bis 123 1 bis 122 ¹	Ein oder mehrere Halteregister schreiben: 1 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
11	11	0	Statuswort und Ereigniszähler der Slave-Kommunikation lesen. Das Statuswort zeigt beschäftigt an (0 – nicht beschäftigt, 0xFFFF - beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert. Beide Operanden DATA_ADDR und DATA_LEN von Modbus_Master werden bei dieser Funktion ignoriert.	
80	08	1	Slave-Zustand über Daten-Diagnosecode 0x0000 prüfen (Prüfschleifentest – Slave gibt ein Echo der Anforderung zurück) 1 Wort pro Anforderung	

MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	Modbus-Adresse
81	08	1	Slave-Ereigniszähler über Daten-Diagnosecode 0x000A zurücksetzen 1 Wort pro Anforderung	
3 bis 10, 12 bis 79, 82 bis 255			Reserviert	

¹ Bei der erweiterten Adressierung ist die maximale Datenlänge je nach Datentyp der Funktion um 1 Byte oder 1 Wort geringer.

Parameter DATA_PTR

Der Parameter DATA_PTR zeigt auf die DB- oder M-Adresse, in die geschrieben oder aus der gelesen wird. Wenn Sie einen Datenbaustein verwenden, müssen Sie einen globalen Datenbaustein anlegen, der den Datenspeicher für Lese- und Schreibvorgänge auf Modbus-Slaves bereitstellt.

Hinweis

Der Datenbaustein DATA_PTR muss die direkte Adressierung zulassen

Der Datenbaustein muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

Ab Version 4.0 der Modbus_Master-Anweisung können Sie das Datenbaustein-Attribut "Optimierter Bausteinzugriff" aktivieren. Sie können nur ein einzelnes Element oder ein Element-Array im optimierten Speicher mit folgenden Datentypen verwenden: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDIInt, Sint oder WChar.

Datenbausteinstrukturen für den Parameter DATA_PTR

- Diese Datentypen gelten für das **Lesen von Wörtern** der Modbus-Adressen 30001 bis 39999, 40001 bis 49999 und 400001 bis 465536 sowie für das **Schreiben von Wörtern** in die Modbus-Adressen 40001 bis 49999 und 400001 bis 465536.
 - Standard-Array der Datentypen WORD, UINT oder INT
 - Benannte Struktur des Typs WORD, UINT oder INT, bei der jedes Element einen eindeutigen Namen und einen 16-Bit-Datentyp hat.
 - Benannte komplexe Struktur, bei der jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Zum **Lesen** und Schreiben von Bits für die Modbus-Adressen 00001 bis 09999 und zum Lesen von Bits von 10001 bis 19999.
 - Standardfeld aus Booleschen Datentypen.
 - Benannte Boolesche Struktur aus eindeutig benannten Booleschen Variablen.

13.5 Modbus-Kommunikation

- Es ist zwar nicht erforderlich, doch empfehlenswert, dass jede Anweisung Modbus_Master einen eigenen getrennten Speicherbereich hat. Grund hierfür ist, dass die Möglichkeit der Datenzerstörung größer ist, wenn mehrere Anweisungen Modbus_Master in demselben Speicherbereich lesen und schreiben.
- Es ist nicht erforderlich, dass die Datenbereiche für DATA_PTR in demselben globalen Datenbaustein liegen. Sie können einen Datenbaustein mit mehreren Bereichen für Modbus-Lesevorgänge, einen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jede Slave-Station anlegen.

Variablen im Datenbaustein (DB) der Anweisung Modbus_Master

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von Modbus_Master, die Sie in Ihrem Programm verwenden können:

Tabelle 13-83 Statische Variablen im Instanz-DB von Modbus_Master

Variable	Datentyp	Standardwert	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), während der auf eine blockierte Instanz von Modbus_Master gewartet wird, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Modbus_Master-Anforderung ausgegeben wurde und das Programm dann stoppt und die Modbus_Master-Funktion aufruft, bevor es die Anforderung vollständig beendet hat. Der Zeitwert muss größer als 0 und kleiner als 55 Sekunden sein, ansonsten tritt ein Fehler auf.
Extended_Addresssing	Bool	FALSCH	Konfiguriert die Adressierung der Slave-Station als einzelnes oder doppeltes Byte: <ul style="list-style-type: none"> • FALSCH = 1-Byte-Adresse, 0 bis 247 • WAHR = 2-Byte-Adresse (entspricht der erweiterten Adressierung), 0 bis 65535
MB_DB	MB_BASE	-	Der Parameter MB_DB der Anweisung Modbus_Comm_Load muss mit dem Parameter MB_DB der Anweisung Modbus_Master verbunden sein.

Ihr Programm kann in die Variablen Blocked_Proc_Timeout und Extended_Addresssing Werte schreiben, um die Modbus_Master-Operationen zu steuern. Im Abschnitt über Modbus_Slave finden Sie in der Beschreibung zu HR_Start_Offset (Seite 1103) und Extended_Addresssing (Seite 1103) ein Beispiel dafür, wie Sie diese Variablen im Programmiereditor nutzen, sowie weitere Einzelheiten zu erweiterter Modbus-Adressierung.

Bedingungscode

Tabelle 13-84 Bedingungscode für die Ausführung von Modbus_Master (Kommunikations- und Konfigurationsfehler) ¹

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
80C8	Slave-Timeout. Der angegebene Slave hat nicht während der angegebenen Zeit geantwortet. Prüfen Sie die Baudrate, Parität und Verdrahtung des Slavegeräts. Dieser Fehler wird erst gemeldet, nachdem alle konfigurierten Wiederholungen durchgeführt wurden.

STATUS (W#16#)	Beschreibung
80C9	<p>Die Zeit der Anweisung Modbus_Master ist aus einem der folgenden Gründe abgelaufen:</p> <ul style="list-style-type: none"> Die Anweisung wartet auf eine Antwort von dem Modul, das für die Kommunikation verwendet wird. Der Wert von Blocked_Proc_Timeout ist zu klein festgelegt. <p>Dieser Fehler wird gemeldet, wenn ein dezentrales PROFIBUS oder PROFINET E/A-Gerät aus einem der folgenden Zustände zurückkehrt:</p> <ul style="list-style-type: none"> Eine Unterbrechung der Spannung oder Kommunikation Ein Ziehen/Stecken-Ereignis bei einem Kommunikationsmodul <p>In diesen Fällen wird die Hardwarekonfiguration vom PLC neu geladen und die Anweisung Modbus_Comm_Load muss erneut ausgeführt werden, um das Kommunikationsmodul richtig zu konfigurieren.</p>
80D1	<p>Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert.</p> <p>Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.</p>
80D2	Die Sendeaufforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung Modbus_Comm_Load
8186	Ungültige Modbus-Stationsadresse
8188	Ungültiger Modus für die Broadcast-Anforderung angegeben
8189	Ungültiger Wert für die Datenadresse.
818A	Ungültiger Wert für die Datenlänge.
818B	Ungültiger Pointer auf die lokale Datenquelle bzw. das lokale Datenziel: Größe nicht richtig
818C	<p>Ungültiger Pointer für DATA_PTR oder Blocked_Proc_Timeout ungültig. Der Datenbereich muss einer der folgenden sein:</p> <ul style="list-style-type: none"> Klassischer DB Array elementarer Datentypen in einem symbolischen oder remanenten DB Merker
8200	Port ist durch Bearbeitung einer Sendeaufforderung belegt.
8280	Negative Quittierung beim Lesen aus Modul. Prüfen Sie den Eingang am Parameter PORT. Dieser Fehler kann durch Verlust eines dezentralen PROFIBUS oder PROFINET E/A-Moduls entweder aufgrund eines Spannungs- oder Kommunikationsverlusts oder aufgrund des Ziehens eines Moduls verursacht werden.
8281	Negative Quittierung beim Schreiben ins Modul. Prüfen Sie den Eingang am Parameter PORT. Dieser Fehler kann durch Verlust eines dezentralen PROFIBUS oder PROFINET E/A-Moduls entweder aufgrund eines Spannungs- oder Kommunikationsverlusts oder aufgrund des Ziehens eines Moduls verursacht werden.

Tabelle 13-85 Bedingungs-codes für die Ausführung von Modbus_Master (Modbus-Protokollfehler) ¹

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	-	CRC-Fehler
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	Größer als 03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)
8386	-	Funktionscode in der Antwort entspricht nicht dem Code in der Anforderung.
8387	-	Der falsche Slave hat reagiert.
8388	-	Die Slave-Antwort auf eine Schreibanforderung ist inkorrekt. Die vom Slave ausgegebene Schreibanforderung entspricht nicht dem, was der Master gesendet hat.

¹ Neben den oben aufgeführten Fehlern der Anweisung Modbus_Master können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

Hinweis

Einstellen der maximalen Länge des Datensatzes für die Profibus-Kommunikation

Wenn das Kommunikationsmodul CM1243-5 als Profibus-Master zur Steuerung eines ET200SP- oder ET200MP-Profibus-Geräts dient, das wiederum ein RS232-, RS422- oder RS485-Punkt-zu-Punkt-Modul verwendet, müssen Sie die Datenbausteinvariable "max_record_len" wie folgt explizit auf den Wert 240 setzen:

Setzen Sie nach Ausführung von Modbus_Comm_Load den Parameter max_record_len im Abschnitt Send_P2P des Instanz-DBs (beispielsweise "Modbus_Master_DB".Send_P2P.max_record_len) auf 240.

Die explizite Zuweisung des Werts für max_record_len ist nur bei Profibus-Verbindungen erforderlich. Bei Profinet-Verbindungen wird bereits ein gültiger Wert für max_record_len verwendet.

Anweisung Modbus_Slave (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Slave kommunizieren)

Tabelle 13-86 Anweisung Modbus_Slave

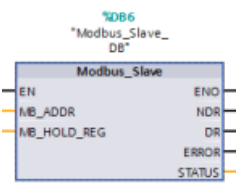
KOP/FUP	SCL	Beschreibung
	<pre>"Modbus_Slave_DB" (MB_ADDR:= uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>Mit der Anweisung Modbus_Slave kann Ihr Programm auf zweierlei Art kommunizieren:</p> <ul style="list-style-type: none"> Als Modbus RTU-Slave über einen PtP-Port des CM (RS485 oder RS232) und CB (RS485) Als Modbus RTU-Slave über die Modbus RTU SIPLUS I/O-Optionen: <ul style="list-style-type: none"> ET 200MP S7-1500CM PtP (RS485 / 422 oder RS232) installieren. ET 200SP S7-1500 CM PtP (RS485 / 422 oder RS232) installieren. <p>Wenn ein dezentraler Modbus-RTU-Master eine Anforderung ausgibt, reagiert Ihr Anwenderprogramm auf die Anforderung, indem es die Anweisung Modbus_Slave ausführt. STEP 7 erstellt automatisch einen Instanz-DB, wenn Sie die Anweisung einfügen. Sie geben mit diesem Modbus_Slave_DB-Namen den Parameter MB_DB für die Anweisung Modbus_Comm_Load an.</p>

Tabelle 13-87 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
MB_ADDR	IN	V1.0: USInt V2.0: UInt	Stationsadresse des Modbus-Slaves: Standard-Adressierungsbereich (1 bis 247) Erweiterter Adressierungsbereich (0 bis 65535)
MB_HOLD_REG	IN_OUT	Variant	Pointer auf den Modbus-Halteregister-DB: Bei dem Modbus-Halteregister kann es sich um den Speicherbereich der Merker oder um einen Datenbaustein handeln.
NDR	OUT	Bool	Neue Daten bereit: <ul style="list-style-type: none"> 0 – Keine neuen Daten 1 – Gibt an, dass neue Daten vom Modbus-Master geschrieben wurden
DR	OUT	Bool	Daten lesen: <ul style="list-style-type: none"> 0 – Keine Daten gelesen 1 – Gibt an, dass Daten vom Modbus-Master gelesen wurden
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Wenn die Ausführung mit einem Fehler beendet wird, ist der Fehlercode im Parameter STATUS nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Fehlercode der Ausführung

Die Funktionscodes der Modbus-Kommunikation (1, 2, 4, 5 und 15) können Bits und Wörter direkt im Prozessabbild der Eingänge und im Prozessabbild der Ausgänge der CPU lesen und

13.5 Modbus-Kommunikation

schreiben. Bei diesen Funktionscodes muss der Parameter MB_HOLD_REG als Datentyp größer als ein Byte definiert werden. Die folgende Tabelle zeigt die Beispielzuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 13-88 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	bis	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	bis	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Die Funktionscodes der Modbus-Kommunikation (3, 6, 16) nutzen ein Modbus-Halteregister, bei dem es sich um einen Adressbereich im Speicherbereich der Merker oder um einen Datenbaustein handeln kann. Der Typ des Halteregisters wird vom Parameter MB_HOLD_REG der Anweisung Modbus_Slave angegeben.

Hinweis

Typ des Datenbausteins MB_HOLD_REG

Ein Datenbaustein mit Modbus-Halteregister muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

Ab Version 4.0 der Modbus_Slave-Anweisung können Sie das Datenbaustein-Attribut "Optimierter Bausteinzugriff" aktivieren. Sie können nur ein einzelnes Element oder ein Element-Array im optimierten Speicher mit folgenden Datentypen verwenden: Bool, Byte, Char, Word, Int, DWord, Dint, Real, USInt, UInt, UDIInt, SInt oder WChar.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 13-89 Zuordnung von Modbus-Adressen zum CPU-Speicher

Modbus-Master-Adresse	Beispiele für Parameter von MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Rezept".Inhaltsstoff[5]

Tabelle 13-90 Diagnosefunktionen

Modbus-Diagnosefunktionen von Modbus_Slave der S7-1200		
Codes	Teilfunktion	Beschreibung
08	0000H	Echotest Abfragedaten ausgeben: <ul style="list-style-type: none"> • Vor STEP 7 V15.1 gibt die Anweisung Modbus_Slave einem Modbus-Master das Echo eines empfangenen Datenworts zurück. • Ab STEP 7 V15.1 gibt die Anweisung Modbus_Slave V4.1 oder höher Echos eines oder mehrerer empfangener Datenwörter zurück.
08	000AH	Kommunikationsereigniszähler löschen: Die Anweisung Modbus_Slave löscht den Kommunikationsereigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationsereigniszähler abrufen: Die Anweisung Modbus_Slave nutzt einen internen Kommunikationsereigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und -Schreibanforderungen, die an den Modbus_Slave gesendet werden, zu erfassen. Der Zähler wird bei Funktion 8, Funktion 11 und Broadcast-Anforderungen nicht hochgezählt. Außerdem wird bei Anforderungen, die zu Kommunikationsfehlern führen (z.B. Paritäts- oder CRC-Fehler), nicht hochgezählt.

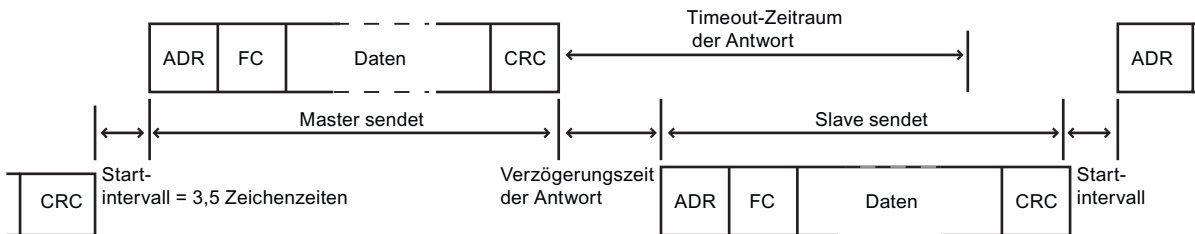
Die Anweisung Modbus_Slave unterstützt Broadcast-Schreibanforderungen von Modbus-Mastern, solange die Anforderungen den Zugriff auf gültige Adressen umfassen. Für Funktionscodes, die bei der Broadcast-Funktion nicht unterstützt werden, erzeugt Modbus_Slave den Fehlercode "0x8188".

Regeln für die Modbus_Slave-Kommunikation

- Modbus_Comm_Load muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung Modbus_Slave über diesen Port kommunizieren kann.
- Wenn ein Port als Slave auf einen Modbus_Master reagieren soll, darf dieser Port nicht mit der Anweisung Modbus_Master programmiert werden.
- Nur eine Instanz von Modbus_Slave kann mit einem bestimmten Port verwendet werden, andernfalls kann unerwartetes Verhalten auftreten.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss den Kommunikationsprozess durch Abfrage der Anweisung Modbus_Slave auf abgeschlossene Sende- und Empfangsvorgänge steuern.
- Die Anweisung Modbus_Slave muss regelmäßig mit einer Häufigkeit ausgeführt werden, die eine zeitnahe Antwort auf eingehende Anforderungen eines Modbus_Masters ermöglicht. Es wird empfohlen, Modbus_Slave in jedem Zyklus aus einem Programmzyklus-OB auszuführen. Modbus_Slave kann aus einem Weckalarm-OB ausgeführt werden, doch ist dies nicht empfehlenswert, weil übermäßige Zeitverzögerungen im Interruptprogramm die Ausführung anderer Interruptprogramme temporär blockieren können.

Zeitsteuerung des Modbus-Signals

Modbus_Slave muss regelmäßig ausgeführt werden, um jede Anforderung des Modbus_Masters zu empfangen und entsprechend zu antworten. Die Häufigkeit der Ausführung von Modbus_Slave richtet sich nach dem vom Modbus_Master vorgegebenen Timeout-Wert für die Antwort. Dies ist in der nachstehenden Abbildung dargestellt.



Der Timeout-Zeitraum für die Antwort RESP_TO ist die Zeitdauer, die ein Modbus_Master auf den Beginn einer Antwort von einem Modbus_Slave wartet. Dieser Zeitraum wird nicht vom Modbus-Protokoll definiert, sondern von einem Parameter des jeweiligen Modbus_Masters. Die Häufigkeit der Ausführung (die Zeit zwischen einer Ausführung und der nächsten Ausführung) von Modbus_Slave muss sich nach den jeweiligen Parametern des Modbus_Masters richten. Sie sollten Modbus_Slave mindestens zweimal während des Timeout-Zeitraums für die Antwort des Modbus_Masters ausführen.

Variablen im Datenbaustein (DB) der Anweisung Modbus_Slave

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von Modbus_Slave, die Sie in Ihrem Programm verwenden können:

Tabelle 13-91 Statische Variablen im Instanz-DB von Modbus_Slave

Variable	Datentyp	Standardwert	Beschreibung
HR_Start_Offset	Word	0	Weist die Anfangsadresse des Modbus-Halteregisters zu (Standard = 0)
Extended_Adressing	Bool	FALSCH	Konfiguriert die Slave-Adressierung als einzelnes oder doppeltes Byte: <ul style="list-style-type: none"> FALSCH = 1-Byte-Adresse WAHR = 2-Byte-Adresse
Request_Count	Word	0	Anzahl aller von diesem Slave empfangenen Anforderungen
Slave_Message_Count	Word	0	Anzahl der für diesen spezifischen Slave empfangenen Anforderungen
Bad_CRC_Count	Word	0	Anzahl der empfangenen Anforderungen, die einen CRC-Fehler aufweisen
Broadcast_Count	Word	0	Anzahl der empfangenen Broadcast-Anforderungen
Exception_Count	Word	0	Modbus-spezifische Fehler, die eine Quittierung mit einer an den Master zurückgegebenen Ausnahme benötigen
Success_Count	Word	0	Anzahl der für diesen spezifischen Slave empfangenen Anforderungen ohne Protokollfehler
MB_DB	MB_BASE	-	Der Parameter MB_DB der Anweisung Modbus_Comm_Load muss mit dem Parameter MB_DB der Anweisung Modbus_Slave verbunden sein.
QB_Start	UInt	0	Startadresse der Ausgangsbytes, in die die CPU schreiben kann (A0 bis AB65535)

Variable	Datentyp	Standardwert	Beschreibung
QB_Count	UInt	65535	Anzahl Bytes, in die ein entferntes Gerät schreiben kann. Ist QB_Count = 0, kann ein entferntes Gerät nicht in die Ausgänge schreiben. Beispiel: Wenn nur AB10 bis AB17 beschreibbar sein sollen, muss QB_Start = 10 und QB_Count = 8 sein.
QB_Read_Start	UInt	0	Startadresse der Ausgangsbytes, aus denen die CPU lesen kann (AB0 bis AB65535)
QB_Read_Count	UInt	65535	Anzahl der Ausgangsbytes, aus denen ein entferntes Gerät lesen kann. Ist QB_Count = 0, kann ein entferntes Gerät nicht aus den Ausgängen lesen. Beispiel: Wenn nur AB10 bis AB17 lesbar sein sollen, muss QB_Start = 10 und QB_Count = 8 sein.
IB_Read_Start	UInt	0	Startadresse der Eingangsbytes, aus denen die CPU lesen kann (EB0 bis EB65535)
IB_Read_Count	UInt	65535	Anzahl der Eingangsbytes, aus denen ein entferntes Gerät lesen kann. Ist IB_Count = 0, kann ein entferntes Gerät nicht aus den Eingängen lesen. Beispiel: Wenn nur EB10 bis EB17 lesbar sein sollen, muss IB_Start = 10 und IB_Count = 8 sein.

Ihr Programm kann Daten in die steuernden Modbus-Server-Operationen und die folgenden Variablen schreiben:

- HR_Start_Offset
- Extended_Addressing
- QB_Start
- QB_Count
- QB_Read_Start
- QB_Read_Count
- IB_Read_Start
- IB_Read_Count

Es gelten folgende Versionsanforderungen für die Verfügbarkeit der Variablen im Datenbaustein (DB) der Modbus_Slave-Anweisung:

Tabelle 13-92 Versionsanforderungen für die Verfügbarkeit der Variablen im Datenbaustein (DB) der Modbus_Slave-Anweisung: Anweisung, TIA Portal und S7-1200 CPU

Version der Modbus_Slave-Anweisung	Version von TIA Portal	Version der S7-1200 CPU Firmware (FW)	Datenbausteinvariablen
3.0	V14 SP1	CPU FW V4.0 oder höher	QB_Start
			QB_Count
4.0 oder höher	V15 oder höher	CPU FW V4.2 oder höher	QB_Start
			QB_Count
			QB_Read_Start
			QB_Read_Count
			IB_Read_Start
			IB_Read_Count

HR_Start_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001 oder 400001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR_Start_Offset konfigurieren, um eine andere Anfangsadresse als 40001 oder 400001 für das Modbus-Halteregister zu konfigurieren.

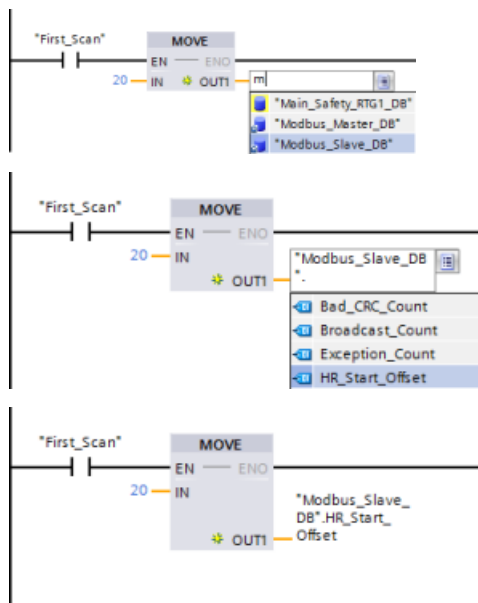
Sie können z. B. ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 400119 führt zu einem Adressierungsfehler.

Tabelle 13-93 Beispiel für die Adressierung des Modbus-Halteregisters

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR_Start_Offset ist ein Wortwert, der die Anfangsadresse des Modbus-Halteregisters angibt und im Instanz-Datenbaustein Modbus_Slave gespeichert ist. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie Modbus_Slave in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise Modbus_Slave in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und den Wert HR_Start_Offset zuweisen. Der Wert muss vor der Ausführung von Modbus_Slave zugewiesen werden.



Modbus-Slave-Variable über den Standard-DB-Namen eingeben:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste "Modbus_Slave_DB" aus.
3. Positionieren Sie den Cursor rechts vom DB-Namen (nach den Anführungszeichen) und geben Sie einen Punkt ein.
4. Wählen Sie in der Klappliste "Modbus_Slave_DB.HR_Start_Offset" aus.

Extended Addressing

Auf die Extended Addressing-Variable wird in ähnlicher Weise zugegriffen wie auf die oben beschriebene HR_Start_Offset-Referenz, außer dass es sich bei der Extended Addressing-Variable um einen Booleschen Wert handelt. Der Boolesche Wert muss von einer Ausgangsspule und kann nicht von einer Box "Verschieben" geschrieben werden.

Für die Modbus-Slave-Adressierung kann ein einzelnes Byte (dies ist der Modbus-Standard) oder ein doppeltes Byte konfiguriert werden. Die erweiterte Adressierung dient zur Adressierung von mehr als 247 Geräten in einem einzigen Netzwerk. Wenn Sie sich für die erweiterte Adressierung entscheiden, können Sie maximal 64.000 Adressen ansprechen. Im Folgenden wird ein Telegramm der Modbus-Funktion 1 als Beispiel gezeigt.

Tabelle 13-94 Slave-Adresse mit einem Byte (Byte 0)

Funktion 1	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	
Anforderung	Slave-Adresse	F-Code	Anfangsadresse		Länge der Spulen		
Gültige Antwort	Slave-Adresse	F-Code	Länge	Spulendaten			
Fehlerantwort	Slave-Adresse	0x81	E-Code				

Tabelle 13-95 Slave-Adresse mit zwei Bytes (Byte 0 und Byte 1)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Anforderung	Slave-Adresse		F-Code	Anfangsadresse		Länge der Spulen	
Gültige Antwort	Slave-Adresse		F-Code	Länge	Spulendaten		
Fehlerantwort	Slave-Adresse		0x81	E-Code			

Zugriff auf Datenbereiche in Datenbausteinen (DB) statt des direkten Zugriffs auf Modbus-Adressen

Ab der Version V4.0 der Modbus_Slave -Anweisung und Firmware-Version V4.2 der S7-1200 CPU kann auf Datenbereiche in DBs zugegriffen werden, statt direkt auf Prozessabbilder und Haltereister zuzugreifen. Hierfür müssen auf der Eigenschaftenseite "Attribute" der globalen DB die Kontrollkästchen "Nur in Ladespeicher ablegen" und "Optimierter Bausteinzugriff" abgewählt werden.

Wenn eine Modbus-Anforderung eintrifft und kein Datenbereich für den Modbus-Datentyp des entsprechenden Funktionscodes festgelegt wurde, behandelt die Anweisung Modbus_Slave die Anforderung wie in früheren Anweisungsversionen: Sie können direkt auf Prozessabbilder und Haltereister zugreifen.

Wenn ein Datenbereich für den Modbus-Datentyp des Funktionscodes festgelegt wurde, wird von der Modbus_Slave -Anweisung aus diesem Datenbereich gelesen bzw. in diesen

Datenbereich geschrieben. Ob gelesen oder geschrieben wird, hängt von der Art des Auftrags ab.

Hinweis

Wenn ein Datenbereich konfiguriert ist, ignoriert die Modbus_Slave -Anweisung die von den statischen Variablen in dem Instanz-Datenbaustein, der dem Datentyp des Datenbereichs entspricht, konfigurierten Offsets oder Bereiche. Diese Offsets und Bereiche gelten nur für das Prozessabbild oder den Speicher, auf das/den MB_HOLD_REG verweist. Die Start- und Längenparameter des Datenbereichs bieten eine eigene Art der Definition von Offsets und Bereichen.

Bei einer einzelnen Modbus-Anforderung kann nur in einen Datenbereich geschrieben bzw. daraus gelesen werden. Sollen beispielsweise Haltereister gelesen werden, die sich über mehrere Datenbereiche erstrecken, sind mehrere Modbus-Anforderungen notwendig.

Für die Festlegung der Datenbereiche gelten folgende Regeln:

- Sie können bis zu acht Datenbereiche in verschiedenen DBs festlegen, wobei jeder DB nur einen Datenbereich enthalten darf. Eine einzelne MODBUS-Anforderung kann nur aus genau einem Datenbereich lesen oder in genau einen Datenbereich schreiben. Jeder Datenbereich entspricht einem MODBUS-Adressbereich. Die Datenbereiche legen Sie in der statischen Variablen "Data_Area_Array" des Instanz-DB fest.
- Sollen weniger als acht Datenbereiche verwendet werden, müssen die benötigten Datenbereiche lückenlos nacheinander angeordnet werden. Beim ersten leeren Eintrag in den Datenbereichen wird die Datenbereichssuche bei der Verarbeitung abgebrochen. Werden zum Beispiel die Feldelemente 1, 2, 4 und 5 als Datenbereiche festgelegt, erkennt "Data_Area_Array" nur die Feldelemente 1 und 2, da Feldelement 3 leer ist.

- Feld Data_Area_Array besteht aus acht Elementen: Data_Area_Array[1] bis Data_Area_Array[8]
- Jedes Feldelement Data_Area_Array[x], $1 \leq x \leq 8$ ist ein UDT des Typs MB_DataArea und wie folgt strukturiert:

Parameter	Datentyp	Bedeutung
data_type	UInt	<p>Kennung für den MODBUS-Datentyp, der auf diesen Datenbereich abgebildet ist:</p> <ul style="list-style-type: none"> • 0: Kennung für ein leeres Feldelement oder einen unbenutzten Datenbereich. Die Werte für db, start und length sind dann ohne Bedeutung. • 1: Prozessabblausgang (in Verbindung mit Funktionscodes 1, 5 und 15) • 2: Prozessabbildeingang (in Verbindung mit Funktionscode 2) • 3: Halteregeister (in Verbindung mit Funktionscodes 3, 6 und 16) • 4: Eingangsregister (in Verbindung mit Funktionscode 4) <p>Hinweis: Wenn Sie einen Datenbereich für einen MODBUS-Datentyp festgelegt haben, kann die Anweisung Modbus_Slave nicht mehr direkt auf diesen MODBUS-Datentyp zugreifen. Wenn die Adresse einer MODBUS-Anforderung für einen solchen Datentyp nicht einem festgelegten Datenbereich entspricht, wird in STATUS der Wert W#16#8383 zurückgegeben.</p>
db	UInt	<p>Nummer des Datenbausteins, auf dem das MODBUS-Register oder die anschließend festgelegten Bits abgebildet sind</p> <p>Die DB-Nummer muss in den Datenbereichen einmalig vorhanden sein. Die DB-Nummer darf nicht in mehreren Datenbereichen vorhanden sein.</p> <p>Auf der Eigenschaftenseite "Attribute" der globalen DB müssen die Kontrollkästchen "Nur in Ladespeicher ablegen" und "Optimierter Bausteinzugriff" abgewählt werden.</p> <p>Die Datenbereiche beginnen auch mit der Byte-Adresse 0 des DB.</p> <p>Zulässige Werte: 1 bis 60999</p>
start	UInt	<p>Erste MODBUS-Adresse, die auf den Datenbaustein abgebildet ist, beginnend mit Adresse 0.0</p> <p>Zulässige Werte: 0 bis 65535</p>
length	UInt	<p>Anzahl Bits (für die Werte 1 und 2 von data_type) oder Anzahl Register (für die Werte 3 und 4 von data_type)</p> <p>Die MODBUS-Adressbereiche eines MODBUS-Datentyps dürfen sich nicht überlappen.</p> <p>Zulässige Werte: 1 bis 65535</p>

13.5 Modbus-Kommunikation

Beispiele für die Festlegung von Datenbereichen:

- Erstes Beispiel: data_type = 3, db = 1, start = 10, length = 6
Die CPU bildet die Haltereister (data_type = 3) in Datenbaustein 1 (db = 1) ab und legt die Modbus-Adresse 10 (start = 10) auf Datenwort 0 und die letzte gültige Modbus-Adresse 15 (length = 6) auf Datenwort 5.
- Zweites Beispiel: data_type = 2, db = 15, start = 1700, length = 112
Die CPU bildet die Eingänge (data_type = 2) in Datenbaustein 15 (db = 15) ab und legt die Modbus-Adresse 1700 (start = 1700) auf Datenwort 0 und die letzte gültige Modbus-Adresse 1811 (length = 112) auf Datenwort 111.

Bedingungs-codes

Tabelle 13-96 Bedingungs-codes für die Ausführung von Modbus_Slave (Kommunikations- und Konfigurationsfehler) ¹

STATUS (W#16#)	Beschreibung
80D1	Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert. Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.
80D2	Die Sende-anforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung Modbus_Comm_Load
8186	Ungültige Modbus-Stationsadresse
8187	Ungültiger Pointer auf MB_HOLD_REG-DB: Bereich ist zu klein
818C	Ungültiger Pointer MB_HOLD_REG. Der Datenbereich muss einer der folgenden sein: <ul style="list-style-type: none"> • Klassischer DB • Array elementarer Datentypen in einem symbolischen oder remanenten DB • Merker

Tabelle 13-97 Bedingungs-codes für die Ausführung von Modbus_Slave (Modbus-Protokollfehler) ¹

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	Keine Antwort	CRC-Fehler
8381	01	Funktioncode nicht unterstützt oder nicht unterstützt in Broadcasts
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	03	Fehler im Datenwert

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)
8389		Ungültige Datenbereichsdefinition: <ul style="list-style-type: none"> • Ungültiger Wert für den Datentyp • DB-Nummer ist ungültig oder nicht vorhanden: <ul style="list-style-type: none"> – Ungültiger DB-Wert – DB-Nummer ist nicht vorhanden – DB-Nummer wird bereits von einem anderen Datenbereich verwendet – DB mit optimiertem Zugriff – DB befindet sich nicht im Arbeitsspeicher • Ungültiger Längenwert • Überschneidung von MODBUS-Adressbereichen, die zu demselben MODBUS-Datentyp gehören

¹ Neben den oben aufgeführten Fehlern der Anweisung Modbus_Slave können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

Hinweis

Einstellen der maximalen Länge des Datensatzes für die PROFIBUS-Kommunikation

Wenn das Kommunikationsmodul CM1243-5 als PROFIBUS-Master zur Steuerung eines ET200SP- oder ET200MP-PROFIBUS-Geräts dient, das wiederum ein RS232-, RS422- oder RS485-Punkt-zu-Punkt-Modul verwendet, müssen Sie die Datenbausteinvariable "max_record_len" wie folgt explizit auf den Wert 240 setzen:

Setzen Sie nach Ausführung von Modbus_Comm_Load den Parameter max_record_len im Abschnitt Send_P2P des Instanz-DBs (beispielsweise "Modbus_Slave_DB".Send_P2P.max_record_len) auf 240.

Die explizite Zuweisung des Werts für max_record_len ist nur bei PROFIBUS-Verbindungen erforderlich. Bei PROFINET-Verbindungen wird bereits ein gültiger Wert für max_record_len verwendet.

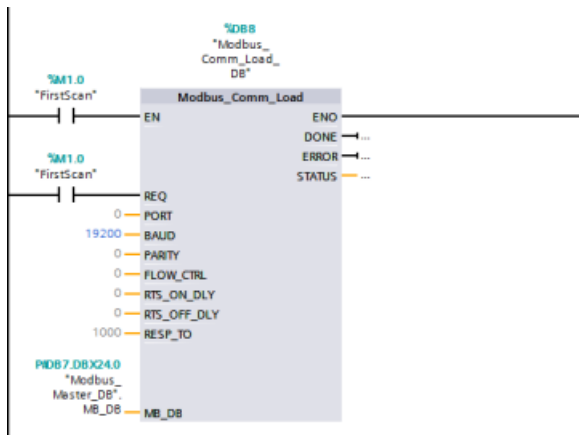
13.5.3.5 Modbus RTU Beispiele

Beispiel: Beispielprogramm für einen Modbus RTU-Master

Modbus_Comm_Load wird während des Anlaufs über den Merker des ersten Zyklus initialisiert. Die Ausführung von Modbus_Comm_Load auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports während der Laufzeit nicht ändert.

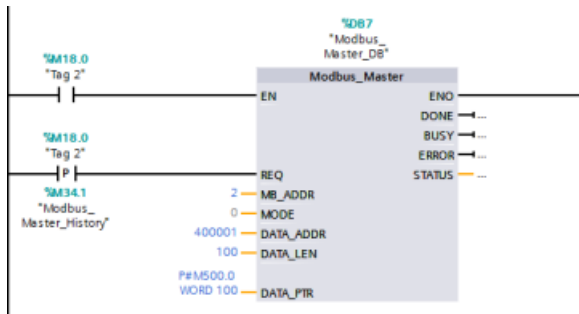
Netzwerk 1: Kommunikationsport des RS485-Moduls nur einmal während des ersten Zyklus konfigurieren/initialisieren.

13.5 Modbus-Kommunikation

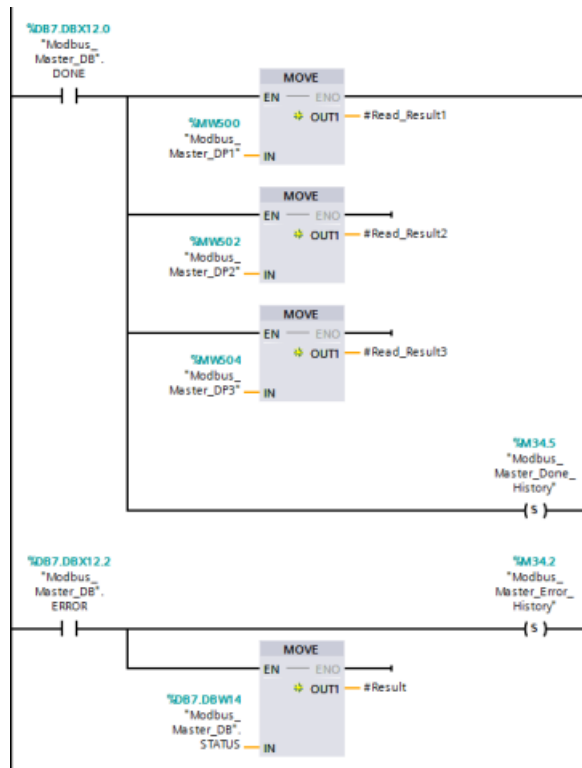


Eine Anweisung Modbus_Master dient im Programmzyklus-OB zur Kommunikation mit einem Slave. Weitere Anweisungen Modbus_Master können im Programmzyklus-OB für die Kommunikation mit anderen Slaves genutzt werden, oder es kann ein Modbus_Master-FB wiederverwendet werden, um mit weiteren Slaves zu kommunizieren.

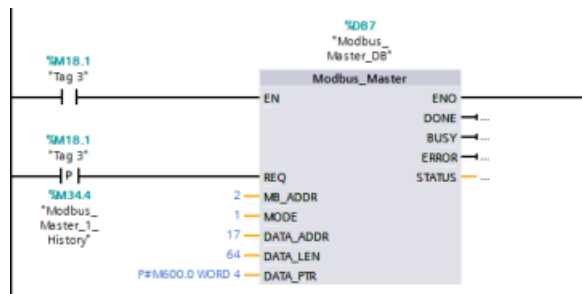
Netzwerk 2: 100 Wörter an Haltereisterdaten von Adresse 400001 in Slave Nr. 2 in Adresse MW500-MW698 auslesen.



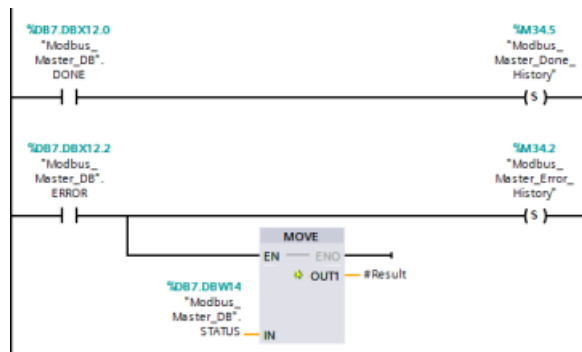
Netzwerk 3: Die ersten drei Wörter der an eine andere Adresse ausgelesenen Haltereisterdaten verschieben und ein Verlaufsbit DONE setzen. Dieses Netzwerk setzt bei Auftreten eines Fehlers auch ein Verlaufsbit ERROR und speichert das Wort STATUS an einer anderen Adresse.



Netzwerk 4: 64 Bit an Daten aus MW600-MW607 in die Ausgangsbitadressen 00017 bis 00081 in Slave Nr. 2 schreiben.



Netzwerk 5: Verlaufsbit DONE setzen, wenn der Schreibvorgang beendet ist. Tritt ein Fehler auf, setzt das Programm das Verlaufsbit ERROR und speichert den STATUS-Code.

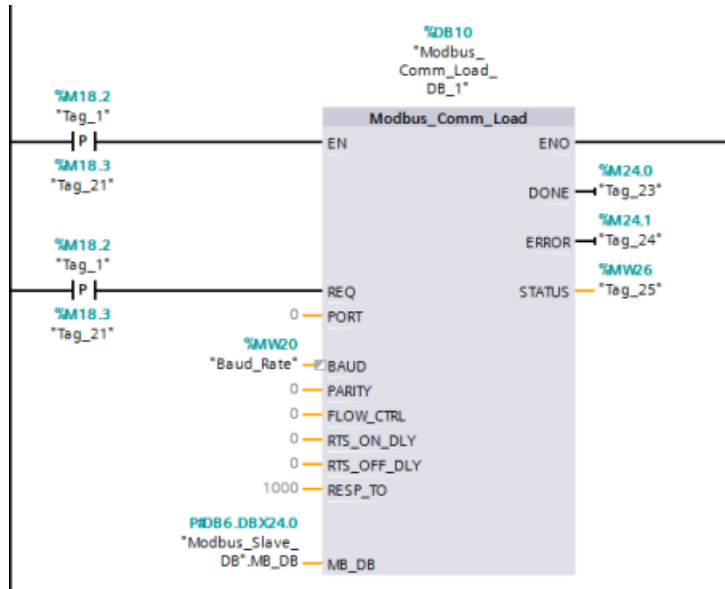


Beispiel: Beispielprogramm für einen Modbus RTU-Slave

Die im Folgenden gezeigte Anweisung MB_COMM_LOAD wird bei jeder Aktivierung von "Variable_1" initialisiert.

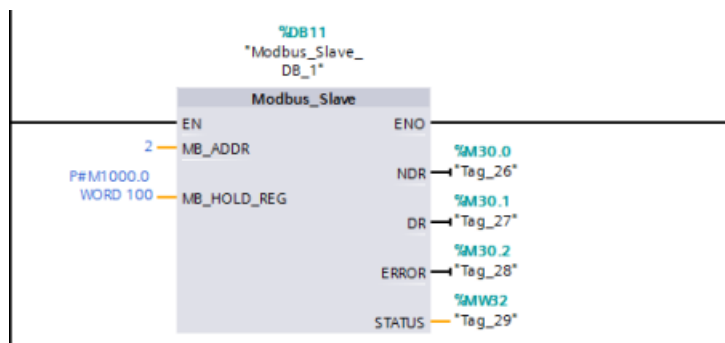
Die Ausführung von MB_COMM_LOAD auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports als Folge der HMI-Konfiguration während der Laufzeit ändert.

Netzwerk 1: RS485-Modulparameter bei jeder Änderung durch ein HMI-Gerät initialisieren.



Die im Folgenden gezeigte Anweisung MB_SLAVE befindet sich in einem zyklischen OB, der alle 10 ms ausgeführt wird. Dies ermöglicht zwar nicht die absolut schnellste Reaktion durch den Slave, doch es bietet ein gutes Betriebsverhalten bei 9600 Baud für kurze Meldungen (bis zu 20 Byte in der Anforderung).

Netzwerk 2: In jedem Zyklus auf Anforderungen des Modbus-Masters prüfen. Für das Modbus-Halteregister sind 100 Wörter mit Beginn an MW1000 konfiguriert.



13.6 PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)

Vor STEP 7 V13 SP1 und den S7-1200 V4.1-CPU's hatten die PtP-Kommunikationsanweisungen andere Namen und teilweise auch etwas andere Schnittstellen. Die allgemeinen Konzepte für die Punkt-zu-Punkt-Kommunikation (Seite 943) sowie die Port- (Seite 947) und Parameterkonfiguration (Seite 963) gelten für beide Anweisungsarten. Informationen zur Programmierung finden sich in den Beschreibungen zu den PtP-Anweisungen der älteren Versionen.

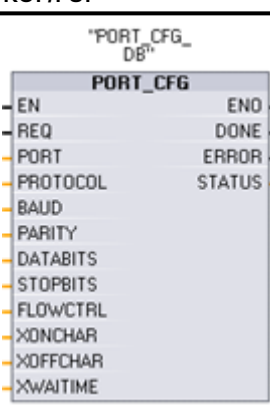
Tabelle 13-98 Gemeinsame Fehlerklassen

Beschreibung der Klasse	Fehlerklassen	Beschreibung
Portkonfiguration	80Ax	Für die Beschreibung häufiger Fehler in der Schnittstellenkonfiguration
Sendekonfiguration	80Bx	Für die Beschreibung häufiger Fehler in der Sendekonfiguration
Empfangskonfiguration	80Cx	Für die Beschreibung häufiger Fehler in der Empfangskonfiguration
Sende-Laufzeit	80Dx	Für die Beschreibung häufiger Laufzeitfehler beim Senden
Empfangs-Laufzeit	80Ex	Für die Beschreibung häufiger Laufzeitfehler beim Empfangen
Signalverarbeitung	80Fx	Für die Beschreibung häufiger Fehler in Verbindung mit der Signalverarbeitung

13.6.1 Ältere Punkt-zu-Punkt-Anweisungen

13.6.1.1 PORT_CFG (Kommunikationsparameter dynamisch konfigurieren)

Tabelle 13-99 Anweisung PORT_CFG (Port-Konfiguration)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>Mit PORT_CFG können Sie Portparameter wie die Baudrate über Ihr Programm ändern.</p> <p>Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit PORT_CFG in Ihrem Programm die Konfiguration ändern.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Die Konfigurationsänderungen von PORT_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere Informationen finden

13.6 PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)

Sie unter Kommunikationsports konfigurieren (Seite 947) und Flusskontrolle verwalten (Seite 949).

Tabelle 13-100 Datentypen für die Parameter

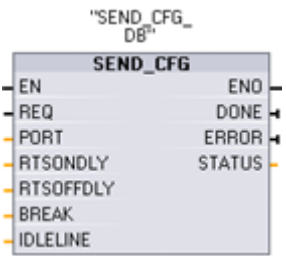
Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke dieses Eingangs. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
PROTOCOL	IN	UInt	0 - Protokoll der Punkt-zu-Punkt-Kommunikation (Standardwert) 1..n - künftige Definition spezifischer Protokolle
BAUD	IN	UInt	Baudrate des Ports (Standardwert: 6): 1 = 300 Baud, 2 = 600 Baud, 3 = 1200 Baud, 4 = 2400 Baud, 5 = 4800 Baud, 6 = 9600 Baud, 7 = 19200 Baud, 8 = 38400 Baud, 9 = 57600 Baud, 10 = 76800 Baud, 11 = 115200 Baud
PARITY	IN	UInt	Parität des Ports (Standardwert: 1): 1 = Keine Parität, 2 = Gerade Parität, 3 = Ungerade Parität, 4 = Mark-Parität, 5 = Space-Parität
DATABITS	IN	UInt	Bits pro Zeichen (Standardwert):1): 1 = 8 Datenbits, 2 = 7 Datenbits
STOPBITS	IN	UInt	Stoppbits (Standardwert: 1): 1 = 1 Stoppbit, 2 = 2 Stoppbits
FLOWCTRL	IN	UInt	Flusskontrolle (Standardwert: 1): 1 = Keine Flusskontrolle, 2 = XON/XOFF, 3 = Hardware-RTS immer EIN, 4 = Hardware-RTS geschaltet
XONCHAR	IN	Char	Gibt das Zeichen an, das als XON-Zeichen dient. Dies ist typischerweise ein DC1-Zeichen (16#11). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 16#11)
XOFFCHAR	IN	Char	Gibt das Zeichen an, das als XOFF-Zeichen dient. Dies ist typischerweise ein DC3-Zeichen (116#3). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 16#13)
XWAITIME	IN	UInt	Gibt an, wie lange nach dem Empfang eines XON-Zeichens auf ein XOFF-Zeichen gewartet werden soll bzw. wie lange nach Aktivierung von RTS auf das CTS-Signal gewartet werden soll (0 bis 65535 ms). Dieser Parameter wird nur ausgewertet, wenn die Flusskontrolle aktiviert ist. (Standardwert: 2000)
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 13-101 Bedingungscode

STATUS (W#16#....)	Beschreibung
80A0	Angegebenes Protokoll ist nicht vorhanden.
80A1	Angegebene Baudrate ist nicht vorhanden.
80A2	Angegebene Paritätsoption ist nicht vorhanden.
80A3	Angegebene Anzahl Datenbits ist nicht vorhanden.
80A4	Angegebene Anzahl Stopbits ist nicht vorhanden.
80A5	Angegebene Art der Flusskontrolle ist nicht vorhanden.
80A6	Wartezeit ist 0 und Flusskontrolle ist aktiviert
80A7	XON und XOFF sind unzulässige Werte (z. B. der gleiche Wert)

13.6.1.2 SEND_CFG (Sendekonfiguration)

Tabelle 13-102 Anweisung SEND_CFG (Sendekonfiguration)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	SEND_CFG ermöglicht das dynamische Konfigurieren serieller Übertragungsparameter für eine PtP-Kommunikationsschnittstelle. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn SEND_CFG ausgeführt wird.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Ports in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit SEND_CFG in Ihrem Programm die Konfiguration ändern.

Die Konfigurationsänderungen von SEND_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Siehe Sendeparameter konfigurieren.

Tabelle 13-103 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
PORT	IN	PORT

13.6 PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)

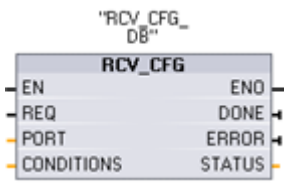
Parameter und Datentyp		Datentyp	Beschreibung
RTSONDLY	IN	UInt	Anzahl der Millisekunden, die nach Aktivierung von RTS gewartet werden soll, bevor eine Übertragung von Tx-Daten durchgeführt wird. Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich liegt zwischen 0 und 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
RTSOFFDLY	IN	UInt	Anzahl der Millisekunden, die nach der Übertragung von Tx-Daten gewartet werden soll, bevor RTS deaktiviert wird: Dieser Parameter ist nur gültig, wenn die Hardwareflusskontrolle aktiviert ist. Der gültige Bereich liegt zwischen 0 und 65535 ms. Der Wert 0 deaktiviert die Funktion. (Standardwert: 0)
BREAK	IN	UInt	Dieser Parameter gibt an, dass beim Start jeder Meldung für die angegebene Anzahl von Bitzeiten eine Pause gesendet wird. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 12)
IDLELINE	IN	UInt	Dieser Parameter gibt an, dass vor dem Start jeder Meldung für die angegebene Anzahl von Bitzeiten die Leitung im Leerlauf bleibt. Maximal sind 65535 Bitzeiten bis maximal 8 Sekunden möglich. Der Wert 0 deaktiviert die Funktion. (Standardwert: 12)
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle Bedingungscode
13-104

STATUS (W#16#....)	Beschreibung
80B0	Konfiguration eines Sendearms ist nicht zulässig.
80B1	Pausenzeit ist größer als der maximal zulässige Wert.
80B2	Leerlaufzeit ist größer als der maximal zulässige Wert.

13.6.1.3 RCV_CFG (Empfangskonfiguration)

Tabelle Anweisung RCV_CFG (Empfangskonfiguration)
13-105

KOP/FUP	SCL	Beschreibung
	<pre>"RCV_CFG_DB" (REQ:=_bool_in_, PORT:=_uint_in_, CONDITIONS:=_struct_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	<p>RCV_CFG führt die dynamische Konfiguration serieller Empfangsparameter für eine PtP-Kommunikationsschnittstelle durch. Diese Anweisung konfiguriert die Bedingungen, die den Beginn und das Ende einer empfangenen Meldung kennzeichnen. In einem CM oder CB anstehende Meldungen werden gelöscht, wenn RCV_CFG ausgeführt wird.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Sie können die statische Anfangskonfiguration des Kommunikationsanschlusses in den Eigenschaften der Gerätekonfiguration einrichten oder einfach die Standardwerte nutzen. Dann können Sie mit RCV_CFG in Ihrem Programm die Konfiguration ändern.

Die Konfigurationsänderungen von RCV_CFG werden in der CPU nicht dauerhaft gespeichert. Die in der Gerätekonfiguration gespeicherten Parameter werden nicht wiederhergestellt, wenn die CPU von RUN in STOP wechselt oder neu eingeschaltet wird. Weitere Informationen finden Sie unter Empfangsparameter konfigurieren (Seite 952).

Tabelle 13-106 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Aktiviert die Konfigurationsänderung bei steigender Flanke an diesem Eingang. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
CONDITIONS	IN	CONDITIONS	Die Datenstruktur von CONDITIONS gibt die Anfangs- und Endbedingungen der Meldung wie im Folgenden beschrieben an.
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Anfangsbedingungen für die Anweisung RCV_PTP

Die Anweisung RCV_PTP nutzt die von der Anweisung RCV_CFG angegebene Konfiguration, um Anfang und Ende von Meldungen der Punkt-zu-Punkt-Kommunikation zu ermitteln. Der Meldungsbeginn wird von den Startbedingungen festgelegt. Der Meldungsbeginn kann anhand von einer oder mehreren Startbedingungen ermittelt werden. Sind mehrere Startbedingungen angegeben, müssen alle Bedingungen erfüllt sein, damit die Meldung gestartet wird.

Eine Beschreibung der Bedingungen für den Meldungsbeginn finden Sie unter "Empfangsparameter konfigurieren (Seite 952)".

Datentypstruktur des Parameters CONDITIONS, Teil 1 (Startbedingungen)

Tabelle 13-107 Struktur von CONDITIONS für START-Bedingungen

Parameter und Datentyp		Datentyp	Beschreibung
STARTCOND	IN	UInt	Angabe der Startbedingung (Standardwert: 1) <ul style="list-style-type: none"> • 01H - Startzeichen • 02H - Beliebiges Zeichen • 04H - Leitungspause • 08H - Leitung im Leerlauf • 10H - Zeichenfolge 1 • 20H - Zeichenfolge 2 • 40H - Zeichenfolge 3 • 80H - Zeichenfolge 4
IDLETIME	IN	UInt	Die Anzahl der erforderlichen Bitzeiten für Timeout des Leitungsleerlaufs. (Standardwert: 40). Nur in Verbindung mit der Bedingung "Leitung im Leerlauf". 0 bis 65535
STARTCHAR	IN	Byte	Das Startzeichen für die Bedingung "Startzeichen". (Standardwert: B#16#2)
SEQ[1].CTL	IN	Byte	Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/vergleichen: (Standardwert: B#16#0) Dies sind die Aktivierungsbits für jedes Zeichen der Startzeichenfolge. <ul style="list-style-type: none"> • 01H - Zeichen 1 • 02H - Zeichen 2 • 04H - Zeichen 3 • 08H - Zeichen 4 • 10H - Zeichen 5 Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zeichenfolge jedes Zeichen eine Übereinstimmung darstellt.
SEQ[1].STR	IN	Char[5]	Zeichenfolge 1, Startzeichen (5 Zeichen). Standardwert: 0
SEQ[2].CTL	IN	Byte	Zeichenfolge 2, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0)
SEQ[2].STR	IN	Char[5]	Zeichenfolge 2, Startzeichen (5 Zeichen). Standardwert: 0
SEQ[3].CTL	IN	Byte	Zeichenfolge 3, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0
SEQ[3].STR	IN	Char[5]	Zeichenfolge 3, Startzeichen (5 Zeichen). Standardwert: 0
SEQ[4].CTL	IN	Byte	Zeichenfolge 4, Steuerung für jedes Zeichen ignorieren/vergleichen. Standardwert: B#16#0
SEQ[4].STR	IN	Char[5]	Zeichenfolge 4, Startzeichen (5 Zeichen), Standardwert: 0

Beispiel

Sehen Sie sich die folgende empfangene hexadezimal-codierte Meldung an: "**68** 10 aa **68** bb 10 aa 16". Die konfigurierten Startzeichenfolgen finden Sie in der folgenden Tabelle. Startzeichenfolgen werden ausgewertet, nachdem das erste Zeichen 68H erfolgreich empfangen wurde. Nach erfolgreichem Empfang des vierten Zeichens (das zweite 68H) ist die Startbedingung 1 erfüllt. Wenn die Startbedingungen erfüllt sind, beginnt die Auswertung der Endbedingungen.

Die Verarbeitung der Startzeichenfolge kann aufgrund verschiedener Fehler bei Parität, Framing oder Zeitabständen zwischen den Zeichen beendet werden. Diese Fehler führen dazu, dass die Meldung nicht empfangen wird, weil die Startbedingung nicht erfüllt wurde.

Tabelle Startbedingungen
13-108

Startbedingung	Erstes Zeichen	Erstes Zeichen +1	Erstes Zeichen +2	Erstes Zeichen +3	Erstes Zeichen +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

Endbedingungen für die Anweisung RCV_PTP

Das Ende einer Meldung wird von den angegebenen Endbedingungen festgelegt. Das Ende einer Meldung wird durch das erste Auftreten einer oder mehrerer konfigurierter Endbedingungen festgelegt. Im Abschnitt "Bedingungen für den Meldungsbeginn" unter "Empfangsparameter konfigurieren (Seite 952)" werden die Endbedingungen beschrieben, die Sie für die Anweisung RCV_CFG konfigurieren können.

Sie können die Endbedingungen entweder in den Eigenschaften der Kommunikationsschnittstelle in der Gerätekonfiguration oder über die Anweisung RCV_CFG konfigurieren. Immer wenn die CPU von STOP in RUN wechselt, werden die Empfangsparameter (Start- und Endbedingungen) wieder auf die Einstellungen der Gerätekonfiguration gesetzt. Wenn das STEP 7 Anwenderprogramm RCV_CFG ausführt, werden die Einstellungen auf die Bedingungen von RCV_CFG gesetzt.

Datentypstruktur des Parameters CONDITIONS, Teil 2 (Endbedingungen)

Tabelle 13-109 Struktur von CONDITIONS für END-Bedingungen

Parameter	Parametertyp	Datentyp	Beschreibung
ENDCOND	IN	UInt 0	Dieser Parameter gibt die Bedingung für das Meldungsende an: <ul style="list-style-type: none"> • 01H - Antwortzeit • 02H - Meldungszeit • 04H - Zeichenabstand • 08H - Maximale Länge • 10H - N + LEN + M • 20H - Zeichenfolge
MAXLEN	IN	UInt 1	Maximale Meldungslänge: Wird nur verwendet, wenn die Endbedingung "Maximale Länge" ausgewählt ist. 1 bis 1024 Bytes
N	IN	UInt 0	Byteposition des Längensfelds in der Meldung. Wird nur bei der Endbedingung N + LEN + M verwendet. 1 bis 1022 Bytes
LENGTHSIZE	IN	UInt 0	Größe des Längensfelds (1, 2 oder 4 Byte). Wird nur bei der Endbedingung N + LEN + M verwendet.
LENGTHM	IN	UInt 0	Geben Sie die Anzahl der Zeichen nach dem Längensfeld an, die nicht im Wert des Längensfelds enthalten sind. Diese Angabe wird nur bei der Endbedingung N + LEN + M verwendet. 0 bis 255 Bytes
RCVTIME	IN	UInt 200	Geben Sie an, wie lange auf das erste empfangene Zeichen gewartet werden soll. Die Empfangsanweisung wird mit einem Fehler beendet, wenn nicht innerhalb der angegebenen Zeit ein Zeichen erfolgreich empfangen wird. Diese Angabe wird nur bei der Bedingung "Antwortzeit" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden) Bei diesem Parameter handelt es sich nicht um eine Endbedingung, weil die Auswertung beendet wird, wenn das erste Zeichen einer Antwort empfangen wird. Es handelt sich nur in dem Sinn um eine Endbedingung, als dass eine Empfangsanweisung beendet wird, weil bei Erwartung einer Antwort keine Antwort empfangen wird. Sie müssen eine getrennte Endbedingung definieren.
MSGTIME	IN	UInt 200	Geben Sie an, wie lange nach dem Empfang des ersten Zeichens auf den vollständigen Empfang der gesamten Meldung gewartet werden soll. Dieser Parameter wird nur verwendet, wenn die Bedingung "Timeout der Meldung" ausgewählt ist. (0 bis 65535 ms)
CHARGAP	IN	UInt 12	Geben Sie die Anzahl der Bitzeiten zwischen den Zeichen an. Wenn die Anzahl der Bitzeiten zwischen den Zeichen den angegebenen Wert überschreitet, ist die Endbedingung erfüllt. Diese Angabe wird nur bei der Bedingung "Zeichenabstand" verwendet. (0 bis 65535 Bitzeiten, maximal 8 Sekunden)

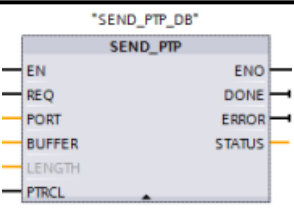
Parameter	Parametertyp	Datentyp	Beschreibung
SEQ.CTL	IN	Byte B#16#0	Zeichenfolge 1, Steuerung für jedes Zeichen ignorieren/vergleichen: Dies sind die Aktivierungsbits für jedes Zeichen der Endezeichenfolge. Zeichen 1 ist Bit 0, Zeichen 2 ist Bit 1, ..., Zeichen 5 ist Bit 4. Wird ein Bit für ein bestimmtes Zeichen deaktiviert, bedeutet dies, dass an dieser Position der Zeichenfolge jedes Zeichen eine Übereinstimmung darstellt.
SEQ.STR	IN	Char[5] 0	Zeichenfolge 1, Startzeichen (5 Zeichen)

Tabelle 13-110 Bedingungscode

STATUS (W#16#...)	Beschreibung
80C0	Unzulässige Startbedingung ausgewählt
80C1	Unzulässige Endebedingung ausgewählt, keine Endebedingung ausgewählt
80C2	Empfangsalarm aktiviert und dies ist nicht möglich.
80C3	Endebedingung "Maximale Länge" ist aktiviert und die maximale Länge ist 0 oder > 1024.
80C4	Berechnete Länge ist aktiviert und N ist >= 1023.
80C5	Berechnete Länge ist aktiviert und Länge ist nicht 1, 2 oder 4.
80C6	Berechnete Länge ist aktiviert und der Wert von M ist > 255.
80C7	Berechnete Länge ist aktiviert und die berechnete Länge ist > 1024.
80C8	Timeout der Antwort ist aktiviert und das Antwort-Timeout ist null.
80C9	Timeout für den Zeichenabstand ist aktiviert und das Timeout ist null.
80CA	Timeout für den Leitungslerlauf ist aktiviert und das Timeout ist null.
80CB	Endezeichenfolge ist aktiviert, doch alle Zeichen sind "nicht relevant".
80CC	Startzeichenfolge (eine von 4) ist aktiviert, doch alle Zeichen sind "nicht relevant".

13.6.1.4 SEND_PTP (Sendepufferdaten übertragen)

Tabelle 13-111 Anweisung SEND_PTP (Punkt-zu-Punkt-Daten senden)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>SEND_PTP startet die Übertragung der Daten und überträgt den zugewiesenen Puffer zur Kommunikationsschnittstelle. Das Programm der CPU wird weiterhin ausgeführt, während das CM oder CB die Daten mit der zugewiesenen Baudrate sendet. Es darf zu jeder Zeit nur eine Sendeanweisung anstehen. Das CM oder CB gibt einen Fehler aus, wenn eine zweite Anweisung SEND_PTP ausgeführt wird, während das CM oder CB bereits eine Meldung sendet.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

13.6 PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)

Tabelle 13-112 Datentypen für die Parameter

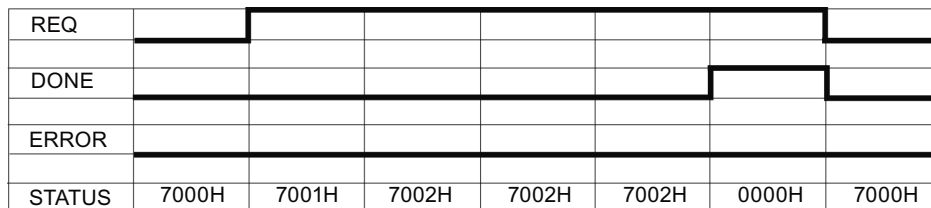
Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Freigabe der angeforderten Übertragung bei steigender Flanke dieses Freigabeingangs. Dadurch wird der Inhalt des Puffers zur Punkt-zu-Punkt-Kommunikationsschnittstelle übertragen. (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Sendepuffers. (Standardwert: 0) Hinweis: Boolesche Daten oder Boolesche Felder werden nicht unterstützt.
LENGTH ¹	IN	UInt	Übertragene Rahmenlänge in Byte (Standardwert: 0) Verwenden Sie beim Senden einer komplexen Struktur immer die Länge 0.
PTRCL	IN	Bool	Reserviert für künftige Verwendung
DONE	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

¹ Optionaler Parameter: Klicken Sie auf den Pfeil im unteren Bereich der KOP/FUP-Box, um die Box zu erweitern und diesen Parameter aufzunehmen.

Wenn eine Sendeanweisung in Bearbeitung ist, sind die Ausgänge DONE und ERROR im Zustand FALSCH. Nach dem Ende einer Sendeanweisung wird einer der Ausgänge DONE oder ERROR auf WAHR gesetzt, um den Zustand der Sendeanweisung zu melden. Während DONE oder ERROR im Zustand WAHR ist, ist der Ausgang STATUS gültig.

Die Anweisung gibt den Status 16#7001 aus, wenn die Kommunikationsschnittstelle die Sendedaten annimmt. Nachfolgende Ausführungen von SEND_PTP geben den Wert 16#7002 aus, wenn das CM oder CB immer noch sendet. Nach dem Ende der Sendeanweisung gibt das CM oder CB den Status 16#0000 für die Sendeanweisung aus (sofern kein Fehler aufgetreten ist). Nachfolgende Ausführungen von SEND_PTP mit REQ = 0 geben den Status 16#7000 (frei) aus.

Das folgende Diagramm zeigt die Beziehung zwischen den Ausgangswerten und REQ. Hierbei wird vorausgesetzt, dass die Anweisung regelmäßig aufgerufen wird, um den Status des Sendevorgangs zu prüfen. In der folgenden Abbildung wird davon ausgegangen, dass die Anweisung in jedem Zyklus aufgerufen wird (dargestellt durch die STATUS-Werte).



Die folgende Abbildung zeigt, wie die Parameter DONE und STATUS nur einen Zyklus lang gültig sind, wenn an der REQ-Leitung (einen Zyklus lang) ein Impuls anliegt, um die Sendeanweisung anzustoßen.

REQ		█						
DONE						█		
ERROR								
STATUS	7000H	7001H	7002H	7002H	7002H	0000H	7000H	7000H

Die folgende Abbildung zeigt die Beziehung der Parameter DONE, ERROR und STATUS im Fehlerfall.

REQ		█						
DONE								
ERROR						█		
STATUS	7000H	7001H	7002H	7002H	7002H	80D1H	7000H	7000H

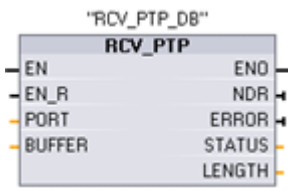
Die Werte DONE, ERROR und STATUS sind nur solange gültig, bis SEND_PTP erneut mit dem gleichen Instanz-DB ausgeführt wird.

Tabelle 13-113 Bedingungs-codes

STATUS (W#16#....)	Beschreibung
80D0	Neue Anforderung bei aktivem Sender
80D1	Sendung abgebrochen, weil kein CTS innerhalb der Wartezeit
80D2	Sendung abgebrochen, weil kein DSR vom DCE-Gerät
80D3	Sendung wegen Überlauf der Warteschlange abgebrochen (mehr als 1024 Bytes senden)
80D5	Umgekehrtes Biassignal (Drahtbruch)
833A	Der DB für den Parameter BUFFER ist nicht vorhanden.

13.6.1.5 RCV_PTP (Empfangsmeldungen aktivieren)

Tabelle Anweisung RCV_PTP (Punkt-zu-Punkt-Daten empfangen)
13-114

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>RCV_PTP prüft die Meldungen, die im CM oder CB empfangen wurden. Wenn eine Meldung verfügbar ist, wird sie vom CM oder CB zur CPU übertragen. Ein Fehler gibt den entsprechenden STATUS-Wert aus.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle Datentypen für die Parameter
13-115

Parameter und Datentyp	Datentyp	Beschreibung	
EN_R	IN	Bool	Wenn dieser Eingang WAHR ist und eine Meldung verfügbar ist, wird die Meldung vom CM oder DB in BUFFER übertragen. Wenn EN_R = FALSCH ist, wird das CM oder CB auf Meldungen geprüft und die Ausgänge NDR, ERROR und STATUS werden aktualisiert, doch die Meldung wird nicht in BUFFER übertragen. (Standardwert: 0)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen. (Standardwert: 0)
BUFFER	IN	Variante	Dieser Parameter zeigt auf die Anfangsadresse des Empfangspuffers. Dieser Puffer muss groß genug sein, um die maximale Meldungslänge zu empfangen. Boolesche Daten oder Boolesche Felder werden nicht unterstützt. (Standardwert: 0)
NDR	OUT	Bool	Eine Ausführung lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde.
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die Anweisung mit Fehler beendet wurde.
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
LENGTH	OUT	UInt	Länge der ausgegebenen Meldung in Byte (Standardwert: 0)

Beachten Sie den folgenden Zusammenhang zwischen dem Eingang EN_R und dem Meldungspuffer der Anweisung RCV_PTP:

Der Eingang EN_R steuert die Kopie einer empfangenen Meldung im BUFFER.

Wenn der Eingang EN_R gleich TRUE ist und eine Meldung verfügbar ist, überträgt die CPU die Meldung vom CM oder CB zum BUFFER und aktualisiert die Ausgänge NDR, ERROR, STATUS, und LENGTH.

Wenn EN_R gleich FALSE ist, prüft die CPU das CM oder CB auf Meldungen und aktualisiert die Ausgänge NDR, ERROR, und STATUS, überträgt die Meldung jedoch nicht zum BUFFER. (Beachten Sie, dass der Standardwert von EN_R gleich FALSE ist.)

Es wird empfohlen, EN_R auf TRUE zu setzen und die Ausführung der Anweisung RCV_PTP mit Eingang EN zu steuern.

Der STATUS-Wert ist gültig, wenn NDR oder ERROR im Zustand WAHR ist. Der STATUS-Wert liefert den Grund für die Beendigung der Empfangsoperation im CM oder CB. Dies ist typischerweise ein positiver Wert, der angibt, dass die Empfangsoperation erfolgreich war und dass der Empfangsvorgang normal beendet wurde. Ist der STATUS-Wert negativ (das höchstwertige Bit des Hexadezimalwerts ist gesetzt), wurde die Empfangsoperation wegen einer Fehlerbedingung wie Paritäts-, Framing- oder Überlauffehler beendet.


Jede PtP-Kommunikationsschnittstelle kann maximal 1024 Bytes puffern. Hierbei kann es sich um eine große oder mehrere kleinere Meldungen handeln. Sind mehrere Meldungen im CM oder CB verfügbar, gibt die Anweisung RCV_PTP die älteste verfügbare Meldung aus. Eine anschließend ausgeführte Anweisung RCV_PTP gibt die zweitälteste Meldung aus.

Tabelle 13-116 Bedingungscode

STATUS (W#16#...)	Beschreibung
0000	Kein Puffer vorhanden
0094	Meldung wurde beendet, weil die maximale Zeichenlänge empfangen wurde
0095	Meldung wurde wegen Meldungs-Timeout beendet
0096	Meldung wurde wegen Zeichenabstands-Timeout beendet
0097	Meldung wurde wegen Antwort-Timeout beendet
0098	Meldung wurde beendet, weil die Längenbedingung "N+LEN+M" erfüllt war
0099	Meldung wurde beendet, weil die Endezeichenfolge erfüllt war
80E0	Meldung wurde beendet, weil der Empfangspuffer voll ist
80E1	Meldung wurde wegen Paritätsfehler beendet
80E2	Meldung wurde wegen Framingfehler beendet
80E3	Meldung wurde wegen Überlauffehler beendet
80E4	Meldung wurde beendet, weil die berechnete Länge die Puffergröße überschreitet
80E5	Umgekehrtes Biassignal (Drahtbruch)
833A	Der DB für den Parameter BUFFER ist nicht vorhanden.

13.6.1.6 RCV_RST (Empfangspuffer löschen)

Tabelle 13-117 Anweisung RCV_RST (Empfänger zurücksetzen)

KOP/FUP	SCL	Beschreibung
	<pre>"RCV_RST_DB" (REQ:=_bool_in_, PORT:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_);</pre>	RCV_RST löscht den Empfangspuffer im CM oder CB.

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

13.6 PtP-Kommunikation in älteren Systemen (nur CM/CB 1241)

Tabelle 13-118 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Aktiviert das Löschen des Empfangspuffers bei steigender Flanke dieses Freigabeingangs (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen. (Standardwert: 0)
DONE	OUT	Bool	Einen Zyklus lang WAHR bedeutet, dass die letzte Anforderung fehlerfrei ausgeführt wurde.
ERROR	OUT	Bool	WAHR bedeutet, dass die letzte Anforderung mit Fehlern ausgeführt wurde. Ist dieser Ausgang WAHR, enthält Ausgang STATUS die zugehörigen Fehlercodes.
STATUS	OUT	Word	Fehlercode (Standardwert: 0) Siehe Gemeinsame Parameter für Punkt-zu-Punkt-Operationen (Seite 963) zu den Codes für den Kommunikationsstatus.

Hinweis

Mit der Anweisung RCV_RST können Sie sicherstellen, dass die Meldungspuffer nach einem Kommunikationsfehler oder nach dem Ändern eines Kommunikationsparameters wie z. B. der Baudrate gelöscht werden. Die Ausführung von RCV_RST bewirkt die Löschung aller internen Meldungspuffer des Moduls. Nach dem Löschen der Meldungspuffer können Sie sicher sein, dass Ihr Programm bei der Ausführung der nächsten Empfangsanweisung nur neue Meldungen zurückgibt und keine alten Meldungen aus dem Zeitraum vor dem Aufruf von RCV_RST.

13.6.1.7 SGN_GET (RS232-Signale abfragen)

Tabelle 13-119 Anweisung SGN_GET (RS232-Signale abrufen)

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>SGN_GET liest die aktuellen Zustände der RS232-Kommunikationssignale.</p> <p>Diese Funktion gilt nur beim RS232-CM.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

Tabelle 13-120 Datentypen für die Parameter


Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	RS232-Signalzustände werden bei der steigenden Flanke an diesem Eingang abgerufen (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablen-tabelle zugewiesen.
NDR	OUT	Bool	Einen Zyklus lang WAHR, wenn neue Daten bereit sind und die Anweisung fehlerfrei beendet wurde
ERROR	OUT	Bool	Einen Zyklus lang WAHR, nachdem die Anweisung mit Fehler beendet wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)
DTR	OUT	Bool	Datenterminal bereit, Modul bereit (Ausgang). Standardwert: False
DSR	OUT	Bool	Datensatz bereit, Kommunikationsteilnehmer bereit (Eingang). Standardwert: False
RTS	OUT	Bool	Sendeanforderung, Modul sendebereit (Ausgang). Standardwert: False
CTS	OUT	Bool	Sendebereit, Kommunikationsteilnehmer kann Daten empfangen (Eingang). Standardwert: False
DCD	OUT	Bool	Datenträger erkannt, Signalpegel empfangen (immer False, nicht unterstützt)
RING	OUT	Bool	Rufanzeige, Meldung eines eingehenden Rufs (immer False, nicht unterstützt)

Tabelle 13-121 Bedingungs-codes

STATUS (W#16#....)	Beschreibung
80F0	CM oder CB ist RS-485 und es sind keine Signale verfügbar

13.6.1.8 SGN_SET (RS-232-Signale einstellen)

Tabelle 13-122 Anweisung SGN_SET (RS232-Signale einstellen)

KOP/FUP	SCL	Beschreibung
	<pre> "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_); </pre>	<p>SGN_SET setzt die Zustände der RS232-Kommunikationssignale. Diese Funktion gilt nur beim RS232-CM.</p>

¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

13.7 USS-Kommunikation in älteren Systemen (nur CM/CB 1241)

Tabelle 13-123 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	IN	Bool	Die Anweisung zum Setzen der RS232-Signale wird bei der steigenden Flanke an diesem Eingang gestartet (Standardwert: Falsch)
PORT	IN	PORT	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen. (Standardwert: 0)
SIGNAL	IN	Byte	Gibt die festzusetzenden Signale an: (mehrere zulässig). Standardwert: 0 <ul style="list-style-type: none"> • 01H = RTS • 02H = DTR • 04H = DSR
RTS	IN	Bool	Sendeanforderung, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch), Standardwert: Falsch
DTR	IN	Bool	Datenterminal bereit, Modul bereit, zu setzenden Wert zu senden (wahr oder falsch). Standardwert: Falsch
DSR	IN	Bool	Datensatz bereit (gilt nur für Schnittstellentyp DCE), nicht verwendet.
DONE	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde
ERROR	OUT	Bool	Eine Ausführung lang WAHR, nachdem die letzte Anforderung mit Fehler ausgeführt wurde
STATUS	OUT	Word	Ausführungsbedingung (Standardwert: 0)

Tabelle 13-124 Bedingungscode

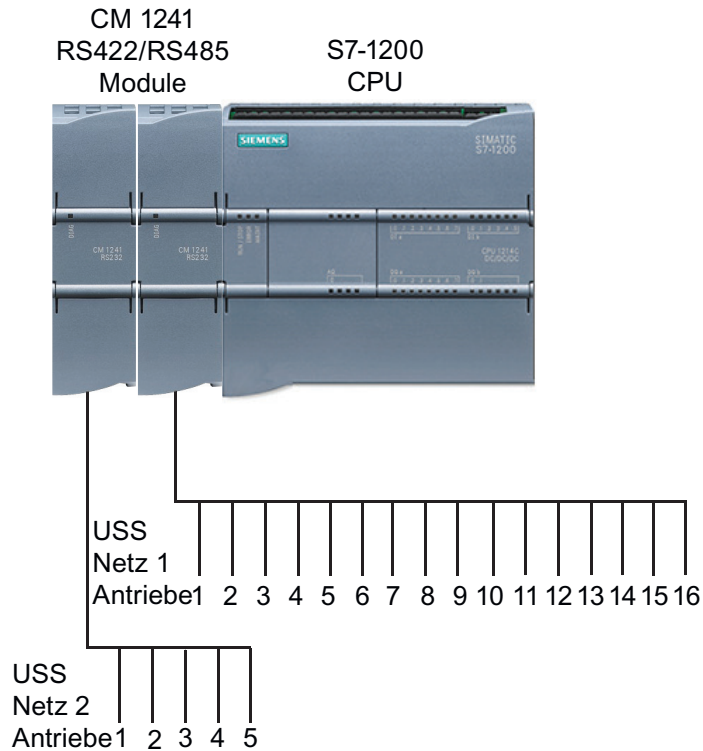
STATUS (W#16#....)	Beschreibung
80F0	CM oder CB ist RS485 und es können keine Signale gesetzt werden
80F1	Signale können wegen Hardwareflusskontrolle nicht gesetzt werden
80F2	DSR kann nicht gesetzt werden, weil das Modul ein DTE-Gerät ist
80F3	DTR kann nicht gesetzt werden, weil das Modul ein DCE-Gerät ist

13.7 USS-Kommunikation in älteren Systemen (nur CM/CB 1241)

Die USS-Anweisungen steuern den Betrieb von Motorantrieben, die das Protokoll der universellen seriellen Schnittstelle (USS) unterstützen. Mit den USS-Anweisungen können Sie über RS485-Verbindungen mit mehreren Antrieben mit CM 1241 RS485-Kommunikationsmodulen oder einem CB 1241 RS485-Kommunikationsboard kommunizieren. In einer S7-1200 CPU können bis zu drei CM 1241 RS422/RS485-Module und ein CB 1241 RS485-Board eingebaut werden. Jeder RS485-Port kann bis zu sechzehn Antriebe betreiben.

Das USS-Protokoll nutzt ein Master/Slave-Netzwerk für die Kommunikation über einen seriellen Bus. Der Master verwendet einen Adressparameter, um eine Meldung an einen ausgewählten

Slave zu senden. Ein Slave selbst kann niemals senden, ohne dafür zuvor eine Anforderung zu erhalten. Die direkte Meldungsübertragung zwischen den einzelnen Slaves ist nicht möglich. Die USS-Kommunikation funktioniert im Halbduplex-Betrieb. Die folgende USS-Abbildung zeigt ein Netzwerkdigramm für eine Beispielanwendung eines Antriebs.



Vor STEP 7 V13 SP1 und den S7-1200 V4.1-CPU's hatten die USS-Kommunikationsanweisungen andere Namen und teilweise auch etwas andere Schnittstellen. Für beide Anweisungsarten gelten die allgemeinen Konzepte. Informationen zur Programmierung finden sich in den Beschreibungen zu den einzelnen USS-Anweisungen.

13.7.1 Version der USS-Anweisungen auswählen

Es gibt zwei Versionen der USS-Anweisungen in STEP 7:

- Version 2.0 war anfangs in STEP 7 Basic/Professional V13 verfügbar.
- Version 2.1 und höher ist in STEP 7 Basic/Professional V13 SP1 und höher verfügbar.

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Sie können nicht beide Versionen der Anweisungen bei demselben Modul verwenden, Sie können jedoch für zwei verschiedene Module unterschiedliche Versionen der Anwendungen einsetzen.



Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.

USS		V1.1
USS_PORT	Edit communication via US...	V1.1
USS_DRV	Swap data with drive	V1.1
USS_RPM	Readout parameters from t...	V1.1
USS_WPM	Change parameters in the d...	V1.1

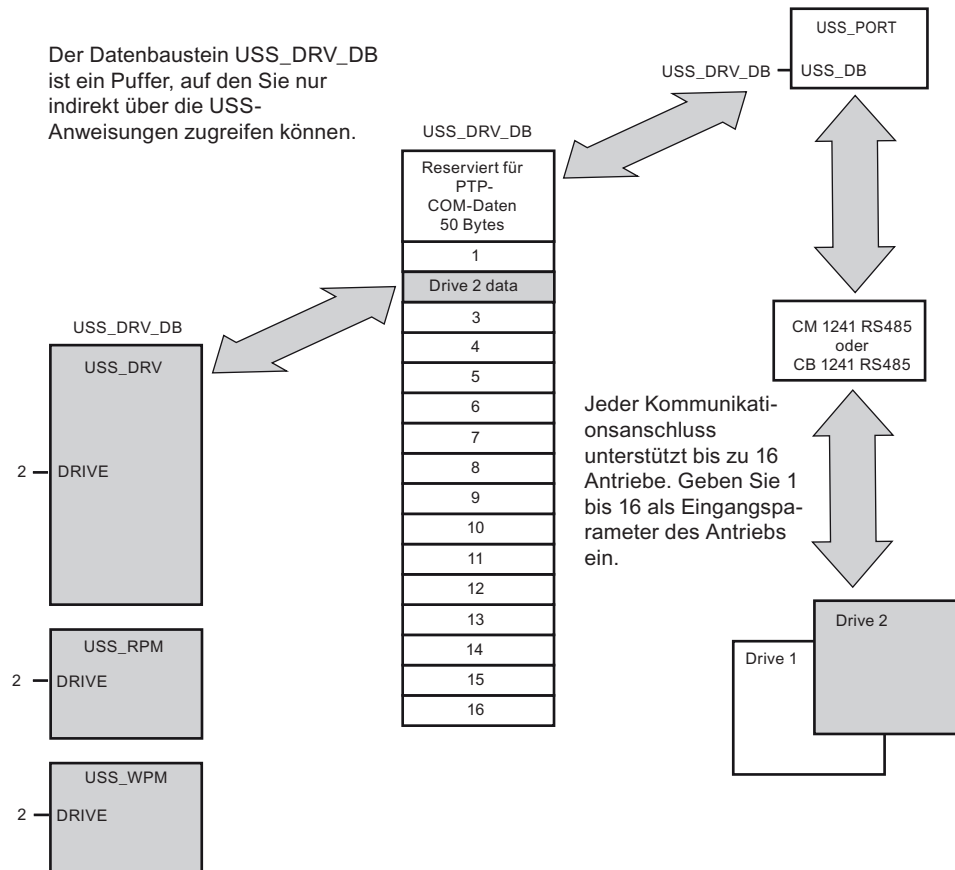
Um die Version einer USS-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine USS-Anweisung über das Anweisungsverzeichnis in Ihr Programm platzieren, wird je nach der ausgewählten USS-Anweisung eine neue FB- oder FC-Instanz in der Projektnavigation angelegt. Die neue FB- oder FC-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer USS-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmierer angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines USS-FBs oder FCs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der USS-Anweisung anzuzeigen.

13.7.2 Voraussetzungen für den Einsatz des USS-Protokolls

Die vier USS-Anweisungen verwenden 1 FB und 3 FCs zur Unterstützung des USS-Protokolls. Für jedes USS-Netzwerk wird ein Instanz-Datenbaustein (DB) USS_PORT verwendet. Der Instanz-Datenbaustein USS_PORT enthält temporäre Speicher und Puffer für alle Antriebe in dem USS-Netzwerk. Die USS-Anweisungen nutzen die Informationen in diesem Datenbaustein gemeinsam.



Alle Antriebe (max. 16), die an einen RS485-Port angeschlossen sind, sind Teil desselben USS-Netzwerks. Alle Antriebe, die an einen anderen RS485-Port angeschlossen sind, sind Teil eines anderen USS-Netzwerks. Jedes USS-Netzwerk wird mithilfe eines eindeutigen Datenbausteins verwaltet. Alle Anweisungen, die zu einem USS-Netzwerk gehören, müssen diesen Datenbaustein gemeinsam nutzen. Dies umfasst alle Anweisungen USS_DRV, USS_PORT, USS_RPM und USS_WPM für die Steuerung aller Antriebe in einem USS-Netzwerk.

Die Anweisung USS_DRV ist ein Funktionsbaustein (FB). Wenn Sie die Anweisung USS_DRV in den Programmiereditor einfügen, werden Sie im Dialog "Aufrufoptionen" aufgefordert, einen DB für diesen FB zuzuweisen. Wenn es sich um die erste Anweisung USS_DRV in diesem Programm für dieses USS-Netzwerk handelt, können Sie die DB-Standardzuweisung übernehmen (oder ggf. den Namen ändern), und der neue DB wird für Sie erstellt. Wenn es sich jedoch nicht um die erste Anweisung USS_DRV für diesen Kanal handelt, müssen Sie im Dialog "Aufrufoptionen" in der Klappliste den DB auswählen, der diesem USS-Netzwerk bereits zuvor zugewiesen wurde.

Bei allen Anweisungen USS_PORT, USS_RPM und USS_WPM handelt es sich um Funktionen (FCs). Wenn Sie diese FCs im Editor einfügen, wird kein DB zugewiesen. Stattdessen müssen Sie

dem Eingang USS_DB dieser Anweisungen den jeweiligen DB zuweisen. Doppelklicken Sie auf das Parameterfeld und klicken Sie dann auf das Symbol, um die verfügbaren DBs anzuzeigen.

Die Funktion USS_PORT steuert die Kommunikation zwischen der CPU und den Antrieben über den Punkt-zu-Punkt(PtP)-RS485-Kommunikationsport. Bei jedem Aufruf dieser Funktion wird eine Kommunikation mit einem Antrieb bearbeitet. Ihr Programm muss diese Funktion schnell genug aufrufen, so dass die Antriebe keine Zeitüberschreitung melden. Diese Funktion kann aus dem Zyklus-OB des Hauptprogramms oder aus einem beliebigen Alarm-OB aufgerufen werden.

In der Regel wird die Funktion USS_PORT aus einem Weckalarm-OB aufgerufen. Stellen Sie die Zykluszeit des Weckalarm-OBs etwa auf die Hälfte eines Mindestaufrufintervalls ein (beispielsweise sollte für die Kommunikation mit 1200 Baud eine Zykluszeit von maximal 350 ms verwendet werden).

Der Funktionsbaustein USS_DRV gibt Ihrem Programm Zugriff auf einen angegebenen Antrieb im USS-Netzwerk. Seine Ein- und Ausgänge entsprechen den Zuständen und den Bedienfunktionen des Antriebs. Sind 16 Antriebe im Netzwerk vorhanden, so muss USS_DRV in Ihrem Programm mindestens 16mal aufgerufen werden, also jeweils einmal für jeden Antrieb. Wie schnell diese Bausteine aufgerufen werden, hängt von der erforderlichen Geschwindigkeit für die Steuerung des Antriebsbetriebs ab.

Sie können den Funktionsbaustein USS_DRV nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.



VORSICHT

Beim Aufruf von USS-Anweisungen aus OBs zu beachten

Rufen Sie USS_DRV, USS_RPM und USS_WPM nur aus einem Zyklus-OB des Hauptprogramms auf. Die Funktion USS_PORT kann aus einem beliebigen OB aufgerufen werden, üblicherweise wird sie aus einem Weckalarm-OB aufgerufen.

Verwenden Sie die Anweisungen USS_DRV, USS_RPM und USS_WPM nicht in einem OB mit einer höheren Priorität als die entsprechende Anweisung USS_PORT. Fügen Sie beispielsweise USS_PORT nicht in das Hauptprogramm und USS_RPM nicht in einen Weckalarm-OB ein. Wird die Unterbrechung der Ausführung von USS_PORT nicht verhindert, kann es zu unerwarteten Fehlern kommen, die zu Verletzungen führen können.

Mit den Funktionen USS_RPM und USS_WPM werden die Betriebsparameter des entfernten Antriebs gelesen und geschrieben. Diese Parameter steuern die interne Funktionsweise des Antriebs. Eine Definition dieser Parameter finden Sie im Handbuch des Antriebs. Ihr Programm kann eine beliebige Anzahl dieser Funktionen enthalten, es kann jedoch immer nur eine Lese- oder Schreibanforderung für einen Antrieb aktiv sein. Sie dürfen die Funktionen USS_RPM und USS_WPM nur aus dem Zyklus-OB eines Hauptprogramms aufrufen.

Zeit für die Kommunikation mit dem Antrieb berechnen

Die Kommunikation mit dem Antrieb läuft asynchron zum Zyklus der S71200 ab. Die S7-1200 durchläuft üblicherweise mehrere Zyklen, bevor die Kommunikation mit einem Antrieb beendet ist.

Das Intervall USS_PORT ist die Zeit, die für eine Transaktion des Antriebs erforderlich ist. Die folgende Tabelle zeigt die Mindestintervalle für USS_PORT für jede Baudrate der Kommunikation. Wenn Sie die Funktion USS_PORT häufiger aufrufen, als es das USS_PORT-Intervall vorgibt, wird die Anzahl der Transaktionen nicht erhöht. Das Timeout-Intervall des

Antriebs ist die Zeitdauer, die für eine Transaktion zur Verfügung steht, wenn zur Fertigstellung der Transaktion aufgrund von Kommunikationsfehlern 3 Versuche nötig sind. Standardmäßig führt die Bibliothek für das USS-Protokoll bei jeder Transaktion bis zu 2 Wiederholungen durch.

Tabelle 13-125 Zeitbedarf berechnen

Baudrate	Berechnetes Mindestintervall für Aufruf von USS_PORT (ms)	Intervall-Timeout für Antriebsmeldung pro Antrieb (ms)
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 Ältere USS-Anweisungen

13.7.3.1 Anweisung USS_PORT (Kommunikation über USS-Netzwerk bearbeiten)

Tabelle 13-126 Anweisung USS_PORT

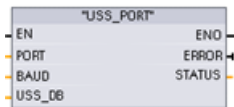
KOP/FUP	SCL	Beschreibung
	<pre>USS_PORT (PORT:=_uint_in_, BAUD:=_dint_in_, ERROR=>_bool_out_, STATUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung USS_PORT bearbeitet die Kommunikation über ein USS-Netzwerk.</p>

Tabelle 13-127 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
PORT	IN	Port
BAUD	IN	DInt
USS_DB	INOUT	USS_BASE



Parameter und Datentyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Wenn WAHR, weist dieser Ausgang darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt des Ergebnis des Zyklus oder der Initialisierung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

Normalerweise ist nur eine Anweisung USS_PORT pro PtP-Kommunikationsport im Programm vorhanden und jeder Aufruf dieser Anweisung steuert eine Übertragung zu oder von einem einzigen Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Instanz-DB nutzen.

Ihr Programm muss die Anweisung USS_PORT oft genug ausführen, damit kein Timeout im Antrieb auftritt. Die Anweisung USS_PORT wird üblicherweise aus einem Weckalarm-OB aufgerufen, um Antriebs-Timeouts zu verhindern und die letzten USS-Datenaktualisierungen für Aufrufe von USS_DRV verfügbar zu haben.

13.7.3.2 Anweisung USS_DRV (Daten mit Antrieb tauschen)

Tabelle 13-128 Anweisung USS_DRV

KOP/FUP	SCL	Beschreibung
<p>Standardansicht</p>  <p>Erweiterte Ansicht</p> 	<pre>"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_);</pre>	<p>Die Anweisung USS_DRV tauscht Daten mit einem Antrieb aus, indem Anfragemeldungen erzeugt und die Antwortmeldungen des Antriebs ausgewertet werden. Es sollte für jeden Antrieb ein eigener Funktionsbaustein verwendet werden, jedoch müssen alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationssport zugewiesen sind, den gleichen Instanz-Datenbaustein verwenden. Sie müssen den DB-Namen eingeben, wenn Sie die erste Anweisung USS_DRV einfügen. Dann verweisen Sie auf diesen DB, der beim Einfügen der ersten Anweisung angelegt wurde.</p> <p>STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.</p>

¹ KOP und FUP: Erweitern Sie die Box, um alle Parameter anzuzeigen. Klicken Sie dazu auf den unteren Bereich der Box. Die Parameteranschlüsse, die grau dargestellt sind, sind optional, eine Parametrierung ist nicht erforderlich.

Tabelle 13-129 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
RUN	IN	Bool
OFF2	IN	Bool
OFF3	IN	Bool

13.7 USS-Kommunikation in älteren Systemen (nur CM/CB 1241)

Parameter und Datentyp		Datentyp	Beschreibung
F_ACK	IN	Bool	Fehlerquittierungsbit: Mit diesem Bit wird das Fehlerbit eines Antriebs zurückgesetzt. Das Bit wird nach dem Löschen des Fehlers gesetzt und der Antrieb erkennt damit, dass der vorherige Fehler nicht mehr gemeldet werden muss.
DIR	IN	Bool	Richtungssteuerung des Antriebs: Dieses Bit wird gesetzt, wenn der Antrieb in Vorwärtsrichtung laufen soll (wenn SPEED_SP positiv ist).
DRIVE	IN	USInt	Adresse des Antriebs: Dieser Eingang ist die Adresse des USS-Antriebs. Der gültige Bereich liegt zwischen Antrieb 1 und Antrieb 16.
PZD_LEN	IN	USInt	Wortlänge: Dies ist die Anzahl der PZD-Datenwörter. Gültige Werte sind 2, 4, 6 oder 8 Wörter. Der Standardwert ist 2.
SPEED_SP	IN	Real	Drehzahlsollwert: Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Frequenz. Ein positiver Wert bedeutet, dass der Antrieb vorwärts läuft (wenn DIR wahr ist). Gültig ist der Bereich von 200,00 bis -200,00.
CTRL3	IN	Word	Steuerwort 3: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL4	IN	Word	Steuerwort 4: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL5	IN	Word	Steuerwort 5: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL6	IN	Word	Steuerwort 6: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL7	IN	Word	Steuerwort 7: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
CTRL8	IN	Word	Steuerwort 8: Wert, der in einen benutzerkonfigurierbaren Parameter des Antriebs geschrieben wird. Sie müssen dies im Antrieb konfigurieren (optionaler Parameter).
NDR	OUT	Bool	Neue Daten bereit: Ist dieser Parameter wahr, so meldet das Bit, dass am Ausgang Daten einer neuen Kommunikationsanforderung bereitstehen.
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist dies darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	Der Zustandswert der Anforderung zeigt des Ergebnis des Zyklus an. Dies ist kein vom Antrieb ausgegebenes Zustandswort.
RUN_EN	OUT	Bool	Betrieb freigegeben: Dieses Bit meldet, ob der Antrieb läuft.
D_DIR	OUT	Bool	Antriebsrichtung: Dieses Bit meldet, ob der Antrieb vorwärts läuft.
INHIBIT	OUT	Bool	Antrieb gesperrt: Dieses Bit meldet den Zustand des Sperrbits für den Antrieb.
FAULT	OUT	Bool	Antriebsfehler: Dieses Bit meldet, dass im Antrieb ein Fehler aufgetreten ist. Sie müssen die Störung beheben und Bit F_ACK setzen, um dieses Bit zu löschen.

Parameter und Datentyp		Datentyp	Beschreibung
SPEED	OUT	Real	Istwert Antriebsdrehzahl (skalierter Wert von Zustandswort 2 des Antriebs): Dies ist die Drehzahl des Antriebs prozentual zur konfigurierten Drehzahl.
STATUS1	OUT	Word	Zustandswort 1 des Antriebs: Dieser Wert enthält feste Zustandsbits eines Antriebs.
STATUS3	OUT	Word	Zustandswort 3 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS4	OUT	Word	Zustandswort 4 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS5	OUT	Word	Zustandswort 5 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS6	OUT	Word	Zustandswort 6 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS7	OUT	Word	Zustandswort 7 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.
STATUS8	OUT	Word	Zustandswort 8 des Antriebs: Dieser Wert enthält ein benutzerkonfigurierbares Zustandswort des Antriebs.

Wenn die erste Ausführung von USS_DRV erfolgt, wird der von der USS-Adresse (Parameter DRIVE) angegebene Antrieb im Instanz-DB initialisiert. Nach dieser Initialisierung können nachfolgende Anweisungen USS_PORT die Kommunikation mit dem Antrieb an dieser Antriebsnummer beginnen.

Wenn Sie die Antriebsnummer ändern, muss die CPU zunächst in STOP und dann wieder in RUN versetzt werden, damit der Instanz-DB initialisiert wird. Die Eingangsparameter werden im USS-Sendepuffer konfiguriert und die Ausgänge werden, sofern vorhanden, aus einem "vorherigen" gültigen Antwortpuffer gelesen. Während der Ausführung der Anweisung USS_DRV findet keine Datenübertragung statt. Nach der Ausführung von USS_PORT kommunizieren die Antriebe. USS_DRV konfiguriert nur die zu sendenden Meldungen und wertet Daten aus, die möglicherweise in einer vorherigen Anforderung empfangen wurden.

Sie können die Drehrichtung des Antriebs entweder über den Eingang DIR (Bool) oder über das Vorzeichen (positiv oder negativ) am Eingang SPEED_SP (Real) steuern. Die folgende Tabelle erläutert, wie diese Eingänge zusammen funktionieren, um die Drehrichtung des Antriebs zu bestimmen, vorausgesetzt der Motor dreht vorwärts.

Tabelle 13-130 Interaktion der Parameter SPEED_SP und DIR

SPEED_SP	DIR	Drehrichtung des Antriebs
Wert > 0	0	Rückwärts
Wert > 0	1	Vorwärts
Wert < 0	0	Vorwärts
Wert < 0	1	Rückwärts

13.7.3.3 Anweisung USS_RPM (Parameter aus dem Antrieb auslesen)

Tabelle Anweisung USS_RPM
13-131

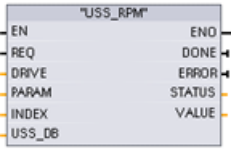
KOP/FUP	SCL	Beschreibung
	<pre> USS_RPM(REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, VALUE=>_variant_out_, USS_DB:=_fbtref_inout_); </pre>	<p>Die Anweisung USS_RPM liest einen Parameter aus einem Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Datenbaustein verwenden. USS_RPM muss aus einem Zyklus-OB des Hauptprogramms aufgerufen werden.</p>

Tabelle Datentypen für die Parameter
13-132

Parametertyp	Datentyp	Beschreibung
REQ	IN	Bool
DRIVE	IN	USInt
PARAM	IN	UInt
INDEX	IN	UInt
USS_DB	INOUT	USS_BASE
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real
DONE ¹	OUT	Bool

Parametertyp		Datentyp	Beschreibung
ERROR	OUT	Bool	Fehler aufgetreten: Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Leseanforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

¹ Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf von USS_RPM oder USS_WPM FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

13.7.3.4 Anweisung USS_WPM (Parameter im Antrieb ändern)

Hinweis

EEPROM-Schreibanweisungen (für den EEPROM in einem USS-Antrieb)

Übertreiben Sie die Verwendung der EEPROM-Schreiboperation nicht. Halten Sie die Anzahl der EEPROM-Schreiboperationen möglichst gering, um die Lebensdauer des EEPROM zu verlängern.

Tabelle 13-133 Anweisung USS_WPM

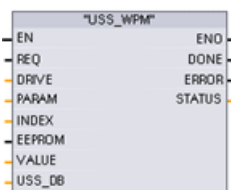
KOP/FUP	SCL	Beschreibung
	<pre>USS_WPM (REQ:=_bool_in_, DRIVE:=_usint_in_, PARAM:=_uint_in_, INDEX:=_uint_in_, EEPROM:=_bool_in_, VALUE:=_variant_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, USS_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung USS_WPM ändert einen Parameter im Antrieb. Alle USS-Funktionen, die einem USS-Netzwerk und einem PtP-Kommunikationsport zugewiesen sind, müssen den gleichen Datenbaustein nutzen.</p> <p>USS_WPM muss aus dem Zyklus-OB eines Hauptprogramms aufgerufen werden.</p>

Tabelle 13-134 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool
DRIVE	IN	USInt

13.7 USS-Kommunikation in älteren Systemen (nur CM/CB 1241)

Parameter und Datentyp		Datentyp	Beschreibung
PARAM	IN	UInt	Parameter Nummer: PARAM gibt an, welcher Antriebsparameter geschrieben wird. Der Bereich für diesen Parameter liegt zwischen 0 und 2047. Bei einigen Antrieben kann das höchstwertige Byte auf PARAM-Werte größer als 2047 zugreifen. Weitere Informationen für den Zugriff auf einen erweiterten Bereich finden Sie im Handbuch zu Ihrem Antrieb.
INDEX	IN	UInt	Parameterindex: INDEX gibt an, in welchen Antriebsparameterindex geschrieben werden soll. Es handelt sich um einen 16-Bit-Wert, bei dem das niederwertigste Byte der tatsächliche Indexwert ist, mit einem Bereich von (0 bis 255). Das höchstwertige Byte kann ebenfalls von dem Antrieb verwendet werden und ist antriebsspezifisch. Weitere Informationen finden Sie im Handbuch Ihres Antriebs.
EEPROM	IN	Bool	Im EEPROM des Antriebs speichern: Wenn WAHR, wird die Transaktion eines Parameters zum Schreiben in den Antrieb im EEPROM des Antriebs gespeichert. Wenn FALSCH, so wird der geschriebene Wert nur temporär gespeichert und geht beim nächsten Einschalten des Antriebs verloren.
VALUE	IN	Word, Int, UInt, DWord, DInt, UInt, Real	Wert des Parameters, in den geschrieben werden soll. Er muss beim Zustandswechsel von REQ gültig sein.
USS_DB	INOUT	USS_BASE	Der Name des Instanz-DBs, der erstellt und initialisiert wird, wenn eine Anweisung USS_DRV in Ihr Programm eingefügt wird.
DONE ¹	OUT	Bool	Wenn WAHR, weist DONE darauf hin, dass der Eingang VALUE in den Antrieb geschrieben wurde. Dieses Bit wird gesetzt, wenn die Anweisung USS_DRV die Schreibantwort des Antriebs erkennt. Dieses Bit wird zurückgesetzt, entweder wenn Sie die Antwortdaten über eine weitere Abfrage USS_WPM anfordern, oder beim zweiten der nächsten beiden Aufrufe von USS_DRV.
ERROR	OUT	Bool	Wenn WAHR, weist ERROR darauf hin, dass ein Fehler aufgetreten und Ausgang STATUS gültig ist. Alle anderen Ausgänge werden bei einem Fehler auf Null gesetzt. Kommunikationsfehler werden nur an den Ausgängen ERROR und STATUS der Anweisung USS_PORT gemeldet.
STATUS	OUT	Word	STATUS gibt das Ergebnis der Schreibforderung an. Weitere Informationen stehen für einige Statuscodes in der Variablen "USS_Extended_Error" zur Verfügung.

¹ Das Bit DONE weist darauf hin, dass gültige Daten aus dem referenzierten Motorantrieb ausgelesen und an die CPU geliefert wurden. Es weist nicht darauf hin, dass die USS-Bibliothek in der Lage ist, sofort einen weiteren Parameter auszulesen. Eine leere PKW-Anforderung muss an den Motorantrieb gesendet und auch von der Anweisung quittiert werden, bevor der Parameterkanal zur Verwendung durch den jeweiligen Antrieb frei wird. Der sofortige Aufruf von USS_RPM oder USS_WPM FC für den spezifischen Motorantrieb führt zu dem Fehler 0x818A.

13.7.4 Alte USS-Statuscodes

Statuscodes der USS-Anweisung werden im Ausgang STATUS der USS-Funktionen ausgegeben.

Tabelle STATUS-Codes ¹
13-135

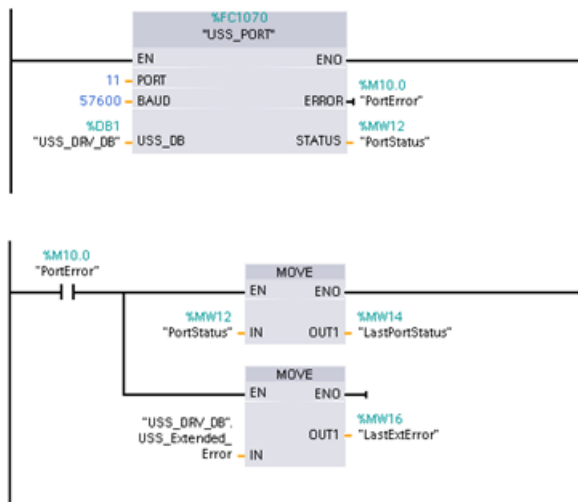
STATUS (W#16#....)	Beschreibung
0000	Kein Fehler
8180	Die Länge der Antwort des Antriebs entsprach nicht den vom Antrieb empfangenen Zeichen. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8181	Der Parameter VALUE gehört nicht zum Datentyp Wort, Real oder Doppelwort.
8182	Der Anwender hat einen Parameterwert vom Typ Wort eingegeben und die Antwort vom Antrieb im Format Doppelwort oder Real empfangen.
8183	Der Anwender hat einen Parameterwert vom Typ Doppelwort oder Real eingegeben und die Antwort vom Antrieb im Format Wort empfangen.
8184	Das Antworttelegramm des Antriebs hatte eine falsche Prüfsumme. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8185	Unzulässige Antriebsadresse (gültiger Adressbereich für Antriebe: 1 bis16)
8186	Der Drehzahlsollwert liegt außerhalb des gültigen Bereichs (gültiger Sollwertbereich für die Drehzahl: -200 % bis 200 %).
8187	Die falsche Antriebsnummer hat auf die gesendete Anforderung geantwortet. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
8188	Unzulässige PZD-Wortlänge angegeben (gültiger Bereich = 2, 4, 6 oder 8 Wörter)
8189	Unzulässige Baudrate angegeben.
818A	Der Anforderungskanal für Parameter wird von einer anderen Anforderung für diesen Antrieb verwendet.
818B	Der Antrieb hat nicht auf Anforderungen und Wiederholungen reagiert. Die Nummer des Antriebs, in dem der Fehler aufgetreten ist, wird in der Variablen "USS_Extended_Error" ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818C	Der Antrieb hat einen erweiterten Fehler zu einer Parameteranforderung ausgegeben. Die Beschreibung der erweiterten Fehler finden Sie unterhalb dieser Tabelle.
818D	Der Antrieb hat einen Fehler "Unzulässiger Zugriff" zu einer Parameteranforderung ausgegeben. Weitere Informationen dazu, weshalb der Parameterzugriff begrenzt sein kann, finden Sie im Handbuch zu Ihrem Antrieb.
818E	Der Antrieb wurde nicht initialisiert. Dieser Fehlercode wird an USS_RPM oder USS_WPM ausgegeben, wenn die Anweisung USS_DRV nicht mindestens einmal für diesen Antrieb aufgerufen wurde. Dadurch wird verhindert, dass die Initialisierung im ersten Zyklus von USS_DRV eine anstehende Anforderung zum Lesen oder Schreiben von Parametern überschreibt, weil dabei der Antrieb als neuer Eintrag initialisiert wird. Um diesen Fehler zu beheben, rufen Sie die Anweisung USS_DRV für diesen Antrieb auf.
80Ax-80Fx	Spezifische Fehler, die von den von der USS-Bibliothek aufgerufenen FBs für die PtP-Kommunikation zurückgegeben werden - Diese Fehlercodes werden von der USS-Bibliothek nicht verändert und sind in den Beschreibungen der PtP-Anweisung definiert.

¹ Neben den oben aufgeführten Fehlern der USS-Anweisungen können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

Für verschiedene STATUS-Codes werden weitere Informationen in der Variable "USS_Extended_Error" des Instanz-DB USS_DRV zur Verfügung gestellt. Für die STATUS-Codes hexadezimal 8180, 8184, 8187 und 818B, enthält USS_Extended_Error die Antriebsnummer des Antriebs, bei dem der Kommunikationsfehler auftrat. Für die STATUS-Codes hexadezimal 818C enthält USS_Extended_Error einen Antriebsfehlercode, der bei der Verwendung einer Anweisung USS_RPM oder USS_WPM vom Antrieb ausgegeben wird.

Beispiel: Meldung von Kommunikationsfehlern

Kommunikationsfehler (STATUS = 16#818B) werden nur bei der Anweisung USS_PORT und nicht bei der Anweisung USS_DRV gemeldet. Beispiel: Wenn das Netzwerk nicht ordnungsgemäß beendet wird, kann ein Antrieb in RUN wechseln, doch die Anweisung USS_DRV zeigt an allen Ausgangsparametern 0 an. In diesem Fall können Sie den Kommunikationsfehler nur über die Anweisung USS_PORT erkennen. Weil dieser Fehler nur einen Zyklus lang sichtbar ist, müssen Sie entsprechende Erfassungslgik einfügen. Dies wird in dem folgenden Beispiel dargestellt. In diesem Beispiel werden, wenn das Fehlerbit der Anweisung USS_PORT WAHR ist, die Werte STATUS und USS_Extended_Error im Speicherbereich der Merker abgelegt. Die Antriebsnummer wird in der Variable USS_Extended_Error abgelegt, wenn der Wert des STATUS-Codes hexadezimal 8180, 8184, 8187 oder 818B ist.



Netzwerk 1 Der Portzustand "PortStatus" und die erweiterten Fehlercode-werte "USS_DRV_DB".USS_Extended_Error sind nur einen Programm-zyklus lang gültig. Die Werte müssen zur späteren Bearbeitung erfasst werden.

Netzwerk 2 Der "PortError"-Kontakt löst die Speicherung des "PortStatus"-Werts in "LastPortStatus" und des "USS_DRV_DB".USS_Extended_Error-Werts in "LastExtError" aus.

Lese- und Schreibzugriff auf die internen Parameter eines Antriebs

USS-Antriebe unterstützen Lese- und Schreibzugriff auf die internen Parameter eines Antriebs. Diese Funktion ermöglicht die dezentrale Steuerung und Konfiguration des Antriebs. Zugriffe der Antriebsparameter können aufgrund von Fehlern wie Wert außerhalb des Bereichs oder unzulässige Anforderungen in der aktuellen Betriebsart des Antriebs fehlschlagen. Der Antrieb erzeugt einen Fehlercode, der in der Variablen "USS_Extended_Error" ausgegeben wird. Dieser Fehlercode gilt nur für die letzte Ausführung der Anweisung USS_RPM oder USS_WPM. Der Fehlercode des Antriebs wird in der Variablen "USS_Extended_Error" abgelegt, wenn der Wert von STATUS code hexadezimal 818C ist. Der Fehlercode von "USS_Extended_Error" richtet sich nach der Variante des Antriebs. Eine Beschreibung der erweiterten Fehlercodes von Lese- und Schreibfunktionen für Parameter finden Sie im Handbuch des Antriebs.

13.7.5 Allgemeine Voraussetzungen für die Antriebseinrichtung mit der alten Anweisung USS

Für die Antriebseinrichtung mit der alten Anweisung USS gelten die folgenden allgemeinen Voraussetzungen:

- Für die Antriebe muss die Verwendung von 4 PKW-Wörtern eingerichtet werden.
- Die Antriebe können für 2, 4, 6 oder 8 PZD-Wörter konfiguriert werden.
- Die Anzahl der PZD-Wörter im Antrieb muss dem Eingang PZD_LEN der Anweisung USS_DRV des Antriebs entsprechen.
- Die Baudrate aller Antriebe muss dem Eingang BAUD der Anweisung USS_PORT entsprechen.
- Der Antrieb muss für die Fernsteuerung eingerichtet sein.
- Für den Frequenzsollwert an der COM-Verbindung des Antriebs muss USS festgelegt sein.
- Für die Antriebsadresse muss 1 bis 16 festgelegt sein. Diese Adresse muss dem Eingang DRIVE am Baustein USS_DRV des Antriebs entsprechen.
- Für die Richtungssteuerung des Antriebs muss die Verwendung der Polarität des Antriebssollwerts eingerichtet werden.
- Das RS485-Netzwerk muss ordnungsgemäß abgeschlossen sein.

Die allgemeine USS-Antriebsverbindung und -einrichtung ist bei den USS-Anweisungen (V4.1) und bei den alten USS-Anweisungen (bis V4.0) identisch. Weitere Informationen finden Sie unter Beispiel: Allgemeine USS-Antriebsverbindung und -einrichtung (Seite 1014).

13.8 Modbus TCP-Kommunikation in älteren Systemen

13.8.1 Übersicht

Vor STEP 7 V13 SP1 und den S7-1200 V4.1-CPU's hatten die Modbus TCP-Kommunikationsanweisungen andere Namen und teilweise auch etwas andere Schnittstellen. Für beide Anweisungsarten gelten die allgemeinen Konzepte. Informationen zur Programmierung finden sich in den Beschreibungen zu den einzelnen Modbus TCP-Anweisungen.

13.8.2 Version der Modbus TCP-Anweisungen auswählen

Die folgenden Versionen der Modbus TCP-Anweisungen sind in STEP 7 verfügbar:

- Alte Version 2.1: Kompatibel mit allen CPU- und CM-Varianten
- Alte Version 3.1: Kompatibel mit allen CPU- und CM-Varianten
- Version 4.2: Kompatibel mit CPUs ab V4.0 und CMs ab V2.1
- Version 5.1: Kompatibel mit CPUs ab V4.2 und CMs ab V2.1
- Version 6.0: Kompatibel mit CPUs ab V4.2 und CMs ab V2.1

13.8 Modbus TCP-Kommunikation in älteren Systemen

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Rufen Sie in der Taskcard "Anweisung" die MODBUS TCP-Anweisungen in der Gruppe Kommunikationsprozessor unter "Andere" auf.

Um die Version einer Modbus TCP-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.



Wenn Sie eine Modbus TCP-Anweisung über das Anweisungsverzeichnis in Ihr Programm einfügen, wird in der Projektnavigation eine neue FB-Instanz angelegt. Die neue FB-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer Modbus TCP-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmeditor angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines Modbus TCP-FBs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der Modbus TCP-Anweisung anzuzeigen.

13.8.3 Ältere Modbus TCP-Anweisungen

13.8.3.1 MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren)

Tabelle 13-136 Anweisung MB_CLIENT

KOP/FUP	SCL	Beschreibung
	<pre>"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_);</pre>	<p>MB_CLIENT kommuniziert als Modbus TCP-Client über den PROFINET-Anschluss an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.</p> <p>MB_CLIENT kann eine Client-Server-Verbindung herstellen, eine Modbus-Funktionsanforderung senden, eine Antwort empfangen und das Trennen der Verbindung von einem Modbus TCP-Server steuern.</p>

Tabelle 13-137 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
REQ	In	Bool	FALSCH = Keine Modbus-Kommunikationsanforderung WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server
DISCONNECT	IN	Bool	Mit dem Parameter DISCONNECT kann Ihr Programm den Verbindungsaufbau und -abbau zu einem Modbus-Servergerät steuern. Ist DISCONNECT = 0 und keine Verbindung vorhanden, versucht MB_CLIENT eine Verbindung zur zugewiesenen IP-Adresse und Portnummer aufzubauen. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird versucht, die Verbindung zu trennen. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
CONNECT_ID	IN	UInt	Der Parameter CONNECT_ID muss jede Verbindung im PLC-Gerät eindeutig identifizieren. Jede eindeutige Instanz der Anweisung MB_CLIENT oder MB_SERVER muss einen eindeutigen Parameter CONNECT_ID enthalten.
IP_OCTET_1	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 1 8-Bit-Anteil der 32-Bit-IP-Adresse (IPv4) des Modbus TCP-Servers, mit dem der Client über das Modbus TCP-Protokoll eine Verbindung herstellt und kommuniziert.
IP_OCTET_2	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 2
IP_OCTET_3	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 3
IP_OCTET_4	IN	USInt	IP-Adresse des Modbus TCP-Servers: Oktett 4
IP_PORT	IN	UInt	Standardwert = 502: Die IP-Portnummer des Servers, mit dem der Client versucht, über das TCP/IP-Protokoll eine Verbindung herzustellen und anschließend zu kommunizieren.
MB_MODE	IN	USInt	Auswahl Modus: Weist die Art der Anforderung (Lesen, Schreiben oder Diagnose) zu. Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_ADDR	IN	UDInt	Modbus-Anfangsadresse: Weist die Anfangsadresse der Daten zu, auf die MB_CLIENT zugreifen soll. Die gültigen Adressen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_LEN	IN	UInt	Modbus-Datenlänge: Weist die Anzahl der Bits oder Wörter zu, auf die diese Anforderung zugreifen soll. Die gültigen Längen finden Sie in der Tabelle der Modbus-Funktionen unten.
MB_DATA_PTR	IN_OUT	Variant	Pointer auf das Modbus-Datenregister: Das Register puffert Daten, die an einen Modbus-Server gesendet oder von dort empfangen werden. Der Pointer muss einen nicht optimierten globalen DB oder eine Adresse im Speicherbereich der Merker zuweisen.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool	<ul style="list-style-type: none"> • 0 - Keine Anweisung MB_CLIENT in Bearbeitung • 1 - Anweisung MB_CLIENT in Bearbeitung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_CLIENT mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

Parameter REQ

FALSCH = Keine Modbus-Kommunikationsanforderung

WAHR = Anforderung für die Kommunikation mit einem Modbus TCP-Server

Wenn keine Instanz von MB_CLIENT aktiv ist und der Parameter DISCONNECT = 0 ist, wird bei REQ = 1 eine neue Modbus-Anforderung gestartet. Ist die Verbindung noch nicht hergestellt, so wird eine neue Verbindung aufgebaut.

Wird dieselbe Instanz von MB_CLIENT erneut mit DISCONNECT = 0 and REQ = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Modbus-Übertragungen durchgeführt. Sobald die aktuelle Anforderung jedoch abgearbeitet ist, kann eine neue Anforderung verarbeitet werden, sofern MB_CLIENT mit REQ = 1 ausgeführt wird.

Wenn die aktuelle MB_CLIENT-Kommunikationsanforderung abgearbeitet ist, ist das Bit DONE einen Zyklus lang WAHR. Das Bit DONE kann als Zeitfenster für die Sequenzierung mehrerer MB_CLIENT-Anforderungen verwendet werden.

Hinweis

Konsistenz der Eingangsdaten während der Verarbeitung von MB_CLIENT

Nachdem ein Modbus-Client eine Modbus-Operation initiiert, werden alle Eingangszustände intern gespeichert und dann bei jedem nachfolgenden Aufruf verglichen. Der Vergleich dient dazu, festzustellen, ob ein bestimmter Aufruf der ursprüngliche Auslöser der aktiven Client-Anforderung war. Mehrere Aufrufe von MB_CLIENT können mit Hilfe eines gemeinsamen Instanz-DBs durchgeführt werden.

Deshalb ist es wichtig, dass die Eingänge während des Zeitraums, in dem eine MB_CLIENT - Operation aktiv verarbeitet wird, nicht geändert werden. Wird diese Regel missachtet, kann MB_CLIENT nicht feststellen, ob es die aktive Instanz ist.

Über die Parameter MB_MODE und MB_DATA_ADDR wählen Sie die Modbus-Kommunikationsfunktion aus.

MB_DATA_ADDR weist die Modbus-Anfangsadresse der Daten zu, auf die zugegriffen werden soll. Die Anweisung MB_CLIENT nutzt statt eines Funktionscodeeingangs einen Eingang MB_MODE.

Die Kombination aus MB_MODE- und MB_DATA_ADDR -Werten legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MB_MODE, der Modbus-Funktion und dem Modbus-Adressbereich.

Tabelle Modbus-Funktionen
13-138

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
0	01	1 bis 2000	Ausgangsbits lesen: 1 bis 2000 Bit pro Anforderung	1 bis 9999
0	02	1 bis 2000	Eingangsbits lesen: 1 bis 2000 Bit pro Anforderung	10001 bis 19999

MB_MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	MB_DATA_ADDR
0	03	1 bis 125	Halteregister lesen: 1 bis 125 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
0	04	1 bis 125	Eingangswörter lesen: 1 bis 125 Wörter pro Anforderung	30001 bis 39999
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
1	15	2 bis 1968	Mehrere Ausgangsbits schreiben: 2 bis 1968 Bits pro Anforderung	1 bis 9999
1	16	2 bis 123	Mehrere Halteregister schreiben: 2 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
2	15	1 bis 1968	Ein oder mehrere Ausgangsbits schreiben: 1 bis 1968 Bit pro Anforderung	1 bis 9999
2	16	1 bis 123	Ein oder mehrere Halteregister schreiben: 1 bis 123 Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
11	11	0	Statuswort und Ereigniszähler der Serverkommunikation lesen. Das Statuswort zeigt beschäftigt an (0 – nicht beschäftigt, 0xFFFF - beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert. Sowohl Parameter MB_DATA_ADDR als auch Parameter MB_DATA_LEN von MB_CLIENT wird bei dieser Funktion ignoriert.	
80	08	1	Serverzustand über Datendiagnosecode 0x0000 prüfen (Prüfschleifentest – Server gibt ein Echo der Anforderung zurück) 1 Wort pro Anforderung	
81	08	1	Server-Ereigniszähler über Datendiagnosecode 0x000A zurücksetzen 1 Wort pro Anforderung	
3 bis 10, 12 bis 79, 82 bis 255			Reserviert	

Hinweis

MB_DATA_PTR weist einen Puffer zu, um aus einem Modbus TCP-Server gelesene bzw. in den Server geschriebene Daten zu speichern

Der Datenpuffer kann sich in einem nicht optimierten globalen DB oder in einer Adresse im Speicherbereich der Merker befinden.

Verwenden Sie für einen Puffer im Speicherbereich der Merker das Standardformat des Any-Pointer. Hierbei handelt es sich um das Format P#"Bitadresse" "Datentyp" "Länge". Ein Beispiel: P#M1000.0 WORD 500.

Zuweisung eines Kommunikationspuffers durch MB_DATA_PTR

- Kommunikationsfunktionen von MB_CLIENT:
 - 1-Bit-Daten aus Modbus-Serveradressen lesen und schreiben (00001 bis 09999)
 - 1-Bit-Daten aus Modbus-Serveradressen lesen (10001 bis 19999)
 - 16-Bit-Wortdaten aus Modbus-Serveradressen lesen (30001 bis 39999) und (40001 bis 49999)
 - 16-Bit-Wortdaten in Modbus-Serveradressen schreiben (40001 bis 49999)
- Daten in Wort- oder Bitgröße werden in den oder aus dem von MB_DATA_PTR zugewiesenen Puffer im DB oder Merkerbereich übertragen.
- Wenn von MB_DATA_PTR ein DB als Puffer zugewiesen ist, müssen Sie allen DB-Datenelementen Datentypen zuweisen.
 - Der 1-Bit-Datentyp Bool stellt eine Modbus-Bitadresse dar.
 - Datentypen mit einzelnen 16-Bit-Wörtern wie WORD, UInt und Int stellen eine Modbus-Wortadresse dar.
 - Datentypen mit 32-Bit-Doppelwörtern wie DWORD, DInt und Real stellen zwei Modbus-Wortadressen dar.
- Komplexe DB-Elemente können von MB_DATA_PTR zugewiesen werden, z.B.
 - Standard-Arrays
 - Benannte Strukturen, in denen jedes Element eindeutig ist.
 - Benannte komplexe Strukturen, in denen jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Es ist nicht erforderlich, dass die Datenbereiche für MB_DATA_PTR in demselben globalen Datenbaustein (oder Speicherbereich der Merker) liegen. Sie können einen Datenbaustein für Modbus-Lesevorgänge, einen anderen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jede MB_CLIENT-Station anlegen.

Mehrere Clientverbindungen

Ein Modbus TCP-Client unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communications-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und -Server, darf die maximale Anzahl der unterstützten Open User Communications-Verbindungen nicht überschreiten (Seite 595). Die Modbus TCP-Verbindungen können von Client- und/oder Serververbindungen gemeinsam genutzt werden.

Einzelne Clientverbindungen müssen die folgenden Regeln einhalten:

- Jede MB_CLIENT-Verbindung muss einen unterschiedlichen Instanz-DB nutzen.
- Jede MB_CLIENT-Verbindung muss eine eindeutige IP-Adresse eines Servers angeben.
- Jede MB_CLIENT-Verbindung muss eine eindeutige Verbindungs-ID angeben.
- Eindeutige IP-Portnummern sind möglicherweise je nach Serverkonfiguration erforderlich.

Die Verbindungs-ID muss für jede einzelne Verbindung eindeutig sein. Das bedeutet, dass für die einzelnen Instanz-DBs nur jeweils eine einzelne, eindeutige Verbindungs-ID verwendet werden

darf. Zusammengefasst heißt dies, dass Instanz-DB und Verbindungs-ID gepaart sind und für jede Verbindung eindeutig sein müssen.

Tabelle 13-139 Für den Benutzer zugänglich statische Variablen des Instanz-Datenbausteins MB_CLIENT

Variable	Datentyp	Voreinstellung	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), die auf eine blockierte Modbus-Client-Instanz gewartet werden soll, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Clientanforderung ausgegeben wurde und die Anwendung dann aufhört, die Clientfunktion auszuführen, bevor die Anforderung vollständig abgearbeitet wurde. Maximaler Grenzwert bei der S7-1200 sind 55 Sekunden.
MB_Unit_ID	Wort	255	Modbus-Geräteerkennung: Ein Modbus TCP-Server wird über seine IP-Adresse angesprochen. Deshalb wird der Parameter MB_UNIT_ID bei der Modbus TCP-Adressierung nicht verwendet. Der Parameter MB_UNIT_ID entspricht dem Feld der Slaveadresse beim Modbus RTU-Protokoll. Wenn ein Modbus TCP-Server als Gateway zu einem Modbus RTU-Protokoll verwendet wird, kann das Slavegerät im seriellen Netzwerk über MB_UNIT_ID identifiziert werden. Der Parameter MB_UNIT_ID würde in diesem Fall die Anforderung an die richtige Modbus RTU-Slaveadresse weiterleiten. Einige Modbus TCP-Geräte benötigen den Parameter MB_UNIT_ID möglicherweise für die Initialisierung innerhalb eines eingeschränkten Wertebereichs.
RCV_TIMEOUT	Real	2,0	Zeit in Sekunden, die MB_CLIENT auf die Antwort eines Servers auf eine Anforderung wartet.
Verbunden	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Server verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden

Tabelle 13-140 MB_CLIENT-Protokollfehler

STATUS (W#16#)	Antwortcode an Modbus-Client (B#16#)	Modbus-Protokollfehler
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Zugriff außerhalb der Grenzen des Adressbereichs von MB_HOLD_REG
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

13.8 Modbus TCP-Kommunikation in älteren Systemen

Tabelle 13-141 Bedingungs-codes der Ausführung von MB_CLIENT¹

STATUS (W#16#)	MB_CLIENT-Parameterfehler
7001	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Dies wird nur für die erste Ausführung eines Verbindungsaufbaus oder -abbaus gemeldet.
7002	MB_CLIENT wartet auf die Antwort eines Modbus-Servers auf die Anforderung eines Verbindungsaufbaus oder -abbaus am zugewiesenen TCP-Port. Dies wird für alle nachfolgenden Ausführungen gemeldet, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.
7003	Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
80C8	Der Server hat nicht während der festgelegten Zeit reagiert. MB_CLIENT muss innerhalb der zugewiesenen Zeit über die ursprünglich gesendete Transaktions-ID eine Antwort erhalten oder es wird dieser Fehler ausgegeben. Prüfen Sie die Verbindung zum Modbus-Servergerät. Dieser Fehler wird erst gemeldet, nachdem konfigurierte Wiederholungen (sofern zutreffend) durchgeführt wurden.
8188	Moduswert ungültig
8189	Ungültiger Wert für die Datenadresse
818A	Ungültiger Wert für die Datenlänge
818B	Ungültiger Pointer auf den DATA_PTR-Bereich. Hierbei kann es sich um die Kombination aus MB_DATA_ADDRESS + MB_DATA_LEN handeln.
818C	Pointer auf einen optimierten DATA_PTR-Bereich (hierbei muss es sich um einen nicht optimierten DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln)
8200	Der Port ist durch die Verarbeitung einer vorhandenen Modbus-Anforderung belegt.
8380	Der empfangene Modbus-Rahmen hat ein fehlerhaftes Format oder es wurden zu wenige Bytes empfangen.
8387	Der zugewiesene Parameter der Verbindungs-ID unterscheidet sich von der für vorherige Anforderungen verwendeten ID. Es darf in jedem Instanz-DB von MB_CLIENT nur eine einzige Verbindungs-ID verwendet werden. Dies ist auch ein interner Fehler, wenn die von einem Server empfangene Modbus TCP-Protokoll-ID nicht 0 ist.
8388	Ein Modbus-Server hat eine Datenmenge zurückgegeben, die sich von der angeforderten Menge unterscheidet. Dies gilt nur für die Modbus-Funktionen 15 und 16.

¹ Neben den oben aufgeführten MB_CLIENT-Fehlern können auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV (Seite 657)) Fehler gemeldet werden.

13.8.3.2 MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren)

MB_SERVER kommuniziert als Modbus TCP-Server über den PROFINET-Anschluss an der S7-1200 CPU. MB_SERVER verarbeitet Verbindungsanforderungen eines Modbus TCP-Clients, empfängt und verarbeitet Modbus-Anforderungen und sendet Antworten.

Für die Verwendung der Anweisung wird kein zusätzliches Hardwaremodul benötigt.

<p>ACHTUNG</p> <p>Sicherheitshinweise</p> <p>Jeder Client im Netzwerk hat Lese- und Schreibzugriff auf die Ein- und Ausgänge im Prozessbild und auf den Datenbaustein- oder Bitspeicherbereich, die im Modbus-Halteregister festgelegt sind.</p> <p>Diese Option soll den Zugriff auf eine IP-Adresse einschränken und unbefugtes Lesen und Schreiben verhindern. Zu beachten ist jedoch, dass die geteilte Adresse auch für den unbefugten Zugriff verwendet werden kann.</p>
--

Tabelle 13-142 Anweisung MB_SERVER

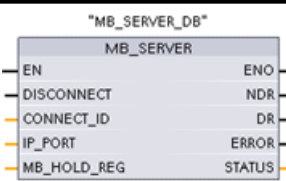
KOP/FUP	SCL	Beschreibung
	<pre>"MB_SERVER_DB" (DISCONNECT:=_bool_in_, CONNECT_ID:=_uint_in_, IP_PORT:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>MB_SERVER kommuniziert als Modbus TCP-Server über den PROFINET-Anschluss an der S7-1200 CPU. Es ist kein zusätzliches Hardwaremodul für die Kommunikation erforderlich.</p> <p>MB_SERVER kann eine Anforderung für den Verbindungsaufbau mit einem Modbus TCP-Client annehmen, eine Modbus-Funktionsanforderung empfangen und eine Antwortmeldung senden.</p>

Tabelle 13-143 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
DISCONNECT IN	Bool	MB_SERVER versucht, eine "passive" Verbindung mit einem Partnergerät aufzubauen. Das bedeutet, dass der Server passiv überwacht, ob es TCP-Verbindungsanforderungen von IP-Adressen gibt, die Anforderungen senden. Ist DISCONNECT = 0 und keine Verbindung vorhanden, kann eine passive Verbindung aufgebaut werden. Ist DISCONNECT = 1 und eine Verbindung vorhanden, wird versucht, die Verbindung zu trennen. Auf diese Weise kann Ihr Programm steuern, wann eine Verbindung angenommen wird. Immer wenn dieser Eingang aktiviert ist, wird kein anderer Vorgang eingeleitet.
CONNECT_ID IN	UInt	CONNECT_ID identifiziert jede Verbindung im PLC-Gerät eindeutig. Jede eindeutige Instanz der Anweisung MB_CLIENT oder MB_SERVER muss einen eindeutigen Parameter CONNECT_ID enthalten.
IP_PORT IN	UInt	Standardwert = 502: Die IP-Portnummer, die den Port identifiziert, der von einem Modbus-Client auf eine Verbindungsanforderung hin überwacht wird.
MB_HOLD_REG IN_OUT	Variant	Pointer auf das Modbus-Halteregister von MB_SERVER: Bei dem Halteregister muss es sich um einen nicht optimierten globalen DB oder um eine Adresse im Speicherbereich der Merker handeln. Dieser Speicherbereich dient dazu, die Werte zu speichern, auf die ein Modbus-Client mit den Modbus-Registerfunktionen 3 (Lesen), 6 (Schreiben) und 16 (Schreiben) zugreifen kann.

Parameter und Datentyp		Datentyp	Beschreibung
NDR	OUT	Bool	Neue Daten bereit: 0 = Keine neuen Daten, 1 = Gibt an, dass von einem Modbus-Client neue Daten geschrieben wurden
DR	OUT	Bool	Daten lesen: 0 = Daten nicht gelesen, 1 = Gibt an, dass die Daten von einem Modbus-Client gelesen wurden
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die Ausführung von MB_SERVER mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

Bei MB_SERVER können eingehende Modbus-Funktionscodes (1, 2, 4, 5 und 15) Bits und Wörter direkt in den Prozessabbildern der Eingänge und Ausgänge der S7-1200 CPU lesen und schreiben. Bei den Funktionscodes für die Datenübertragung (3, 6 und 16) muss der Parameter MB_HOLD_REG als Datentyp größer als ein Byte definiert sein. Die folgende Tabelle zeigt die Zuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 13-144 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	An	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	An	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	An	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Funktionscodes eingehender Modbus-Meldungen (3, 6 und 16) lesen oder schreiben Wörter in einem Modbus-Halteregister, bei dem es sich um einen Adressbereich im Speicherbereich der Merker oder um einen Datenbaustein handeln kann. Der Typ des Halteregisters wird vom Parameter MB_HOLD_REG angegeben.

Hinweis

Zuweisung des Parameters MB_HOLD_REG

Bei dem Modbus-Halteregister kann es sich um einen nicht optimierten globalen DB oder um eine Adresse im Speicherbereich der Merker handeln.

Für ein Modbus-Halteregister im Speicherbereich der Merker verwenden Sie das Standardformat des Any-Pointers. Dies ist das Format P#"Bitadresse" "Datentyp" "Länge".
Beispiel: P#M1000.0 WORD 500.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom

maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 13-145 Beispiele für die Zuordnung von Modbus-Adressen zu Adressen im Speicher der CPU

Modbus-Adresse	Beispiele für Parameter von MB_HOLD_REG		
	P#M100.0 Word 5	P#DB10.DBx0.0 Word 5	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	"Rezept".Inhaltsstoff[5]

Mehrere Serververbindungen

Es können mehrere Serververbindungen hergestellt werden. Dadurch ist es möglich, dass ein einziges PLC-Gerät gleichzeitig Verbindungen mit mehreren Modbus TCP-Clients herstellt.

Ein Modbus TCP-Server unterstützt gleichzeitige Verbindungen bis zur maximalen für das PLC-Gerät zulässigen Anzahl von Open User Communication-Verbindungen. Die Gesamtzahl der Verbindungen für ein PLC-Gerät, einschließlich Modbus TCP-Clients und -Server, darf die maximale Anzahl der unterstützten Open User Communication-Verbindungen nicht überschreiten (Seite 595). Die Modbus TCP-Verbindungen können von Client- und/oder Serververbindungen gemeinsam genutzt werden.

Einzelne Serververbindungen müssen die folgenden Regeln einhalten:

- Jede MB_SERVER-Verbindung muss einen unterschiedlichen Instanz-DB nutzen.
- Jede MB_SERVER-Verbindung muss mit einer eindeutigen IP-Portnummer hergestellt werden. Je Port wird nur eine Verbindung unterstützt.
- Jede MB_SERVER-Verbindung muss eine eindeutige Verbindungs-ID verwenden.
- MB_SERVER muss für jede Verbindung einzeln aufgerufen werden (mit dem entsprechenden Instanz-DB).

Die Verbindungs-ID muss für jede einzelne Verbindung eindeutig sein. Das bedeutet, dass nur mit jedem einzelnen Instanz-DB eine einzelne, eindeutige Verbindungs-ID verwendet werden darf. Zusammengefasst heißt dies, dass Instanz-DB und Verbindungs-ID gepaart sind und für jede Verbindung eindeutig sein müssen.

Tabelle 13-146 Funktionscodes der Modbus-Diagnose

Modbus-Diagnosefunktionen von MB_SERVER		
Codes	Teilfunktion	Beschreibung
08	0x0000	Echotest Abfragedaten ausgeben: Die Anweisung MB_SERVER gibt einem Modbus-Client das Echo eines empfangenen Datenworts zurück.

Modbus-Diagnosefunktionen von MB_SERVER		
08	0x000A	Kommunikationsereigniszähler löschen: Die Anweisung MB_SERVER löscht den Kommunikationsereigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationsereigniszähler abrufen: Die Anweisung MB_SERVER nutzt einen internen Kommunikationsereigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und Modbus-Schreibanforderungen, die an den Modbus-Server gesendet werden, zu erfassen. Der Zähler wird bei Anforderungen der Funktion 8 oder Funktion 11 nicht hochgezählt. Auch bei Anforderungen, die zu einem Kommunikationsfehler führen, wird der Zähler nicht inkrementiert. Die Broadcast-Funktion ist bei Modbus TCP nicht verfügbar, weil nur eine Client-Server-Verbindung zur Zeit vorhanden ist.

Variablen von MB_SERVER

Diese Tabelle zeigt die öffentlichen statischen Variablen im Instanz-Datenbaustein von MB_SERVER, die in Ihrem Programm verwendet werden können.

Tabelle Öffentliche statische Variablen von MB_SERVER
13-147

Variable	Datentyp	Standardwert	Beschreibung
HR_Start_Offset	Wort	0	Weist die Anfangsadresse des Modbus-Halteregisters zu
Request_Count	Wort	0	Die Anzahl aller von diesem Server empfangenen Anforderungen
Server_Message_Count	Wort	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen
Xmt_Rcv_Count	Wort	0	Die Anzahl der Sendungen oder Datenempfänge, bei denen ein Fehler aufgetreten ist. Wird auch inkrementiert, wenn eine Meldung empfangen wird, bei der es sich um eine ungültige Modbus-Meldung handelt.
Exception_Count	Wort	0	Modbus-spezifische Fehler, die eine zurückgegebene Ausnahme erfordern
Success_Count	Wort	0	Die Anzahl der für diesen spezifischen Server empfangenen Anforderungen ohne Protokollfehler.
Verbunden	Bool	0	Gibt an, ob die Verbindung zum zugewiesenen Client verbunden oder nicht verbunden ist: 1 = verbunden, 0 = nicht verbunden

Ihr Programm kann in HR_Start_Offset Werte schreiben und die Modbus-Server-Operationen steuern. Die anderen Variablen können gelesen werden, um den Modbus-Zustand zu überwachen.

HR_Start_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR_Start_Offset konfigurieren, um eine andere Anfangsadresse als 40001 für das Modbus-Halteregister zu konfigurieren.

Sie können zum Beispiel ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des

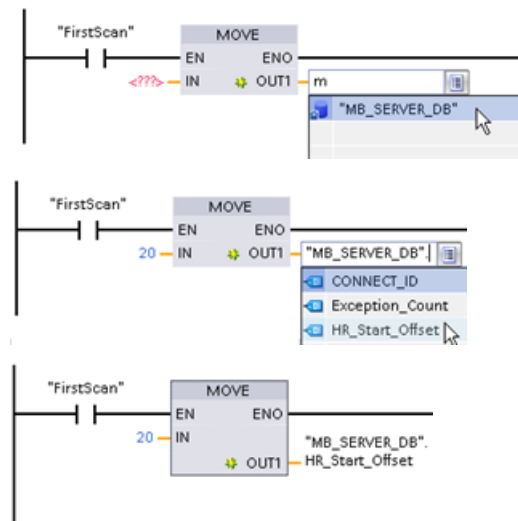
Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 40119 führt zu einem Adressierungsfehler.

Tabelle 13-148 Beispiel für die Adressierung des Modbus-Halteregisters

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR_Start_Offset ist ein Wortwert, der die Anfangsadresse des Modbus-Halteregisters angibt und im Instanz-Datenbaustein MB_SERVER gespeichert ist. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie MB_SERVER in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise MB_SERVER in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und den Wert HR_Start_Offset zuweisen. Der Wert muss vor der Ausführung von MB_SERVER zugewiesen werden.



Eingabe einer Variable für den Modbus-Server über den Namen des Standard-DB:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste der DB-Namen "MB_SERVER_DB" aus.
3. Wählen Sie in der Klappliste der DB-Variablen "MB_SERVER_DB.HR_Start_Offset" aus.

Tabelle 13-149 Bedingungscode der Ausführung von MB_SERVER ¹

STATUS (W#16#)	Antwortcode an Modbus-Server (B#16#)	Modbus-Protokollfehler
7001		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird bei der ersten Ausführung eines Verbindungsaufbaus oder -abbaus gemeldet.
7002		MB_SERVER wartet darauf, dass ein Modbus-Client die Verbindung zum zugewiesenen TCP-Port herstellt. Dieser Code wird für alle nachfolgenden Ausführungen gemeldet, während auf die Fertigstellung eines Verbindungsaufbaus oder -abbaus gewartet wird.

13.8 Modbus TCP-Kommunikation in älteren Systemen

STATUS (W#16#)	Antwortcode an Modbus-Server (B#16#)	Modbus-Protokollfehler
7003		Ein Verbindungsabbau wurde erfolgreich durchgeführt (nur einen PLC-Zyklus lang gültig).
8187		Ungültiger Pointer auf MB_HOLD_REG: Bereich ist zu klein
818C		Pointer auf einen optimierten MB_HOLD_REG-Bereich (hierbei muss es sich um einen nicht optimierten DB-Bereich oder um einen Bereich im Speicherbereich der Merker handeln) oder Timeout für gesperrten Prozess überschreitet den Grenzwert von 55 Sekunden. (spezifisch für die S7-1200)
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Zugriff außerhalb der Grenzen des Adressbereichs von MB_HOLD_REG
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

¹ Neben den oben aufgeführten MB_SERVER-Fehlern können auch von den zugrunde liegenden T-Bausteinanweisungen für die Kommunikation (TCON, TDISCON, TSEND und TRCV (Seite 657)) Fehler gemeldet werden.

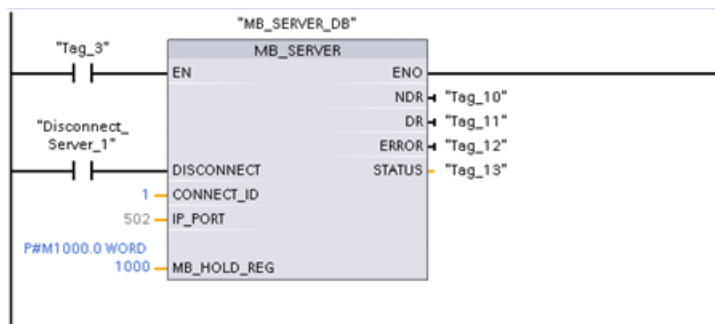
13.8.4 Ältere Modbus TCP-Beispiele

13.8.4.1 Beispiel: MB_SERVER für mehrere TCP-Verbindungen in älteren Systemen

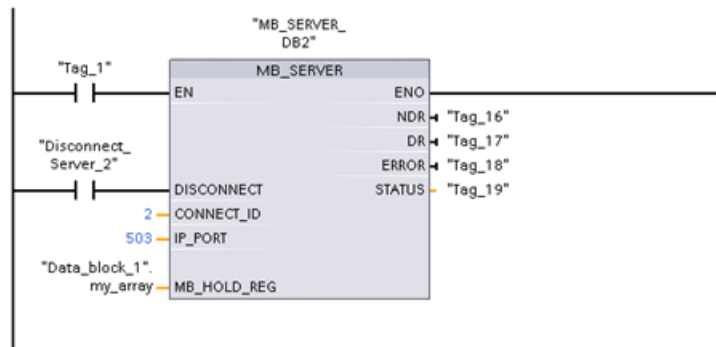
Sie können mehrere Verbindungen zum Modbus TCP-Server haben. Hierfür muss MB_SERVER für jede Verbindung unabhängig ausgeführt werden. Jede Verbindung muss einen unabhängigen Instanz-DB, eine Verbindungs-ID und einen IP-Port verwenden. Bei der S7-1200 ist nur je eine Verbindung pro IP-Port zulässig.

Um optimales Betriebsverhalten zu erzielen, muss MB_SERVER in jedem Programmzyklus für jede Verbindung ausgeführt werden.

Netzwerk 1: Verbindung Nr. 1 mit unabhängigem IP_PORT, Verbindungs-ID und Instanz-DB



Netzwerk 2: Verbindung Nr. 2 mit unabhängigem IP_PORT, Verbindungs-ID und Instanz-DB



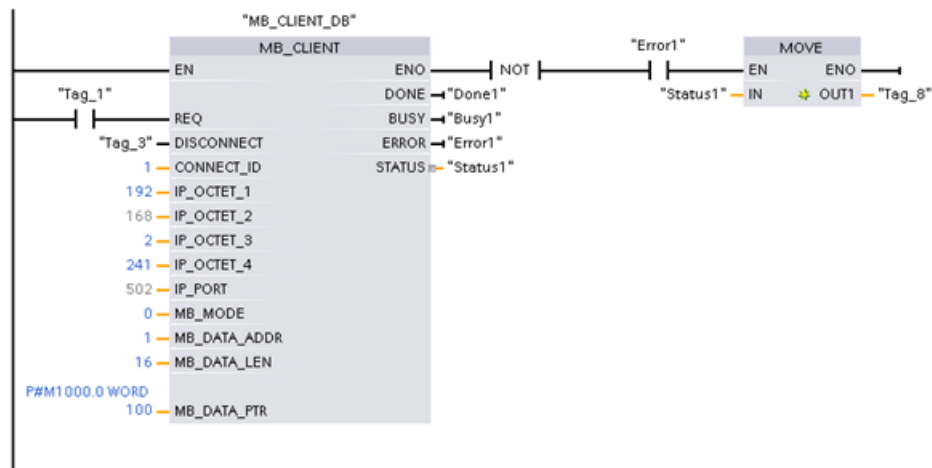
13.8.4.2 Beispiel: MB_CLIENT 1 in älteren Systemen: Mehrere Anforderungen mit gemeinsamer TCP-Verbindung

Mehrere Modbus-Clientanforderungen können über die gleiche Verbindung gesendet werden. Hierfür verwenden Sie den gleichen Instanz-DB, die gleiche Verbindungs-ID und Portnummer.

Es kann jeweils nur 1 Client aktiv sein. Nachdem ein Client seine Ausführung beendet hat, beginnt der nächste Client mit der Ausführung. Ihr Programm ist für die Reihenfolge der Ausführung verantwortlich.

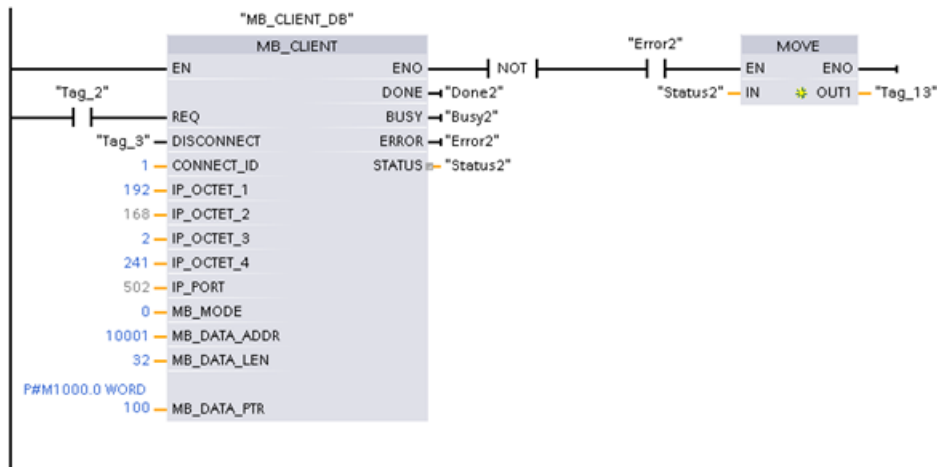
Das Beispiel zeigt beide Clients, die in denselben Speicherbereich schreiben. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

Netzwerk 1: Modbus-Funktion 1 - 16 Bits im Prozessabbild der Ausgänge lesen



Netzwerk 2: Modbus-Funktion 2 - 32 Bits im Prozessabbild der Eingänge lesen

13.8 Modbus TCP-Kommunikation in älteren Systemen



13.8.4.3 Beispiel: MB_CLIENT 2 in älteren Systemen: Mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen

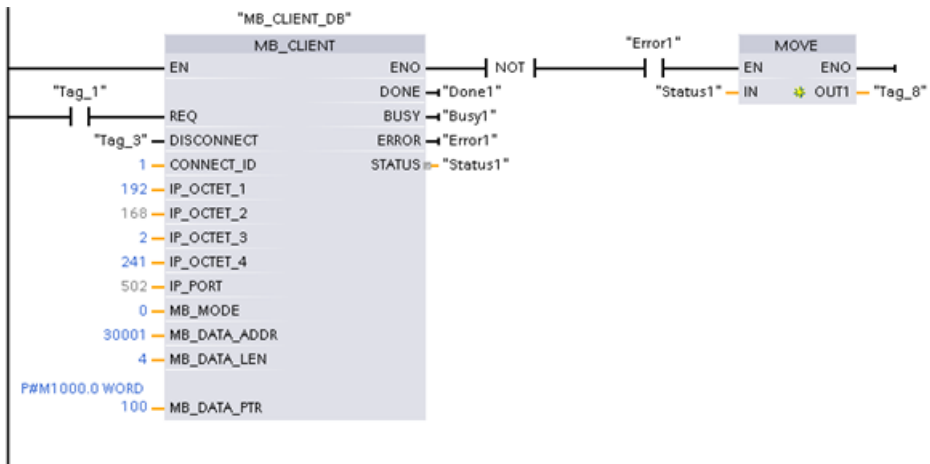
Mehrere Modbus-Clientanforderungen können über verschiedene Verbindungen gesendet werden. Hierfür müssen unterschiedliche Instanz-DBs, IP-Adressen und Verbindungs-IDs verwendet werden.

Die Portnummer muss unterschiedlich sein, wenn die Verbindungen mit demselben Modbus-Server aufgebaut werden. Bei Verbindungen zu unterschiedlichen Servern gibt es keine Einschränkungen hinsichtlich der Portnummer.

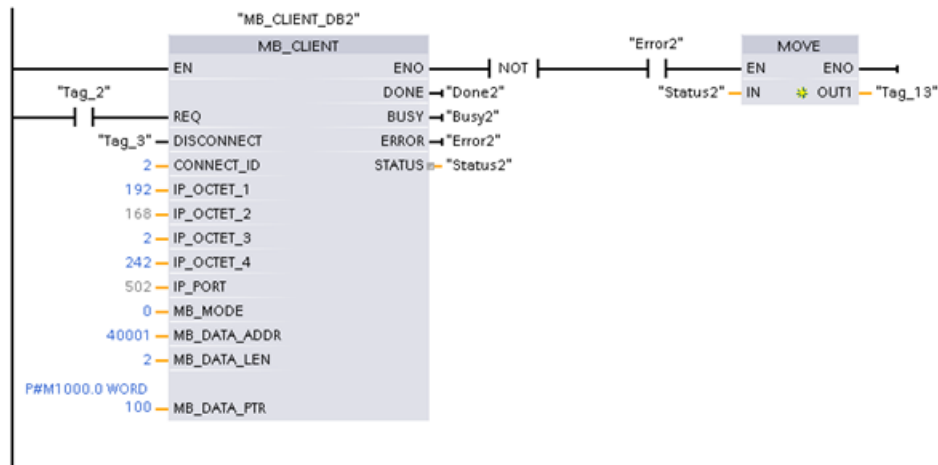
Das Beispiel zeigt beide Clients, die in denselben Speicherbereich schreiben. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

Netzwerk 1:

Modbus-Funktion 4 - Eingangswörter lesen (im Speicher der S7-1200)



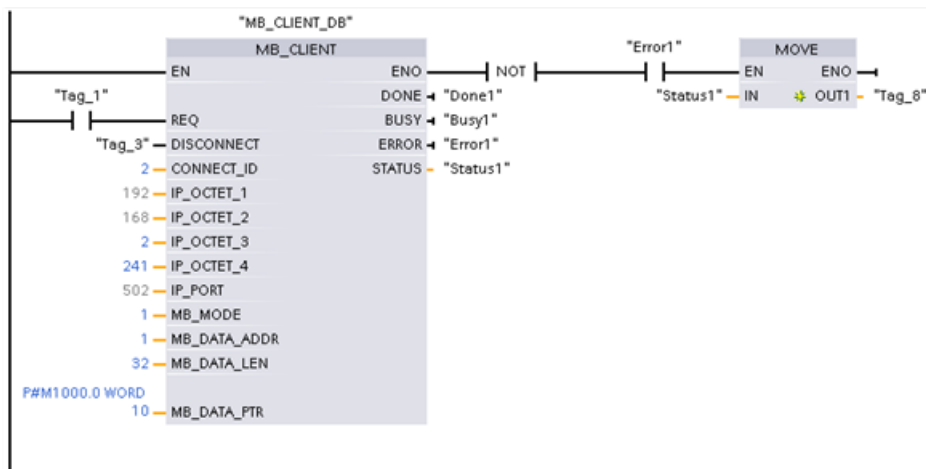
Netzwerk 2: Modbus-Funktion 3 - Wörter im Haltereister von einem Modbus-TCP-Server lesen



13.8.4.4 Beispiel: MB_CLIENT 3 in älteren Systemen: Schreibenanforderung für das Prozessabbild der Ausgänge

Dieses Beispiel zeigt die Anforderung eines Modbus-Clients zum Schreiben in das Prozessabbild der Ausgänge der S7-1200.

Netzwerk 1: Modbus-Funktion 15 - Bits in das Prozessabbild der Ausgänge der S7-1200 schreiben



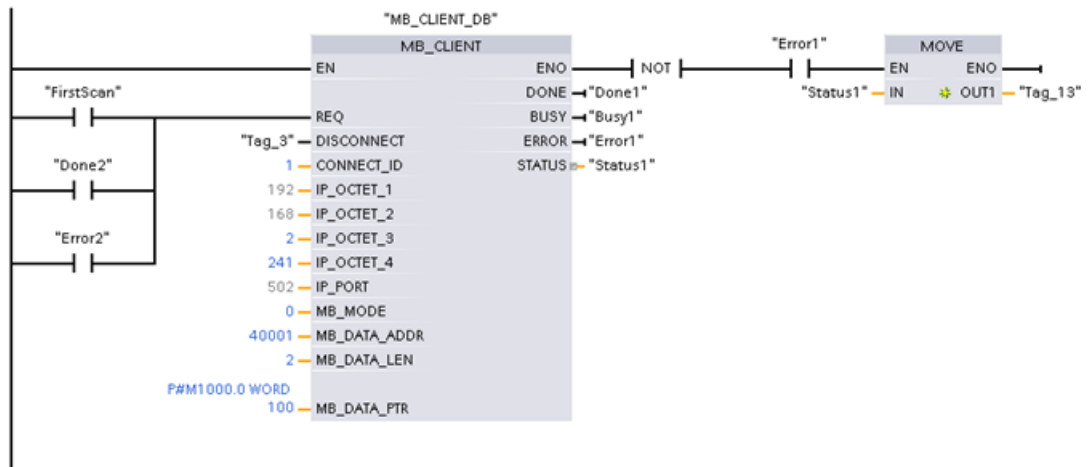
13.8.4.5 Beispiel: MB_CLIENT 4 in älteren Systemen: Mehrere Anforderungen koordinieren

Sie müssen sicherstellen, dass die Ausführung jeder einzelnen Modbus TCP-Anforderung abgeschlossen wird. Diese Koordination muss durch Ihr Programm erfolgen. Das folgende Beispiel zeigt, wie die Ausführung mit Hilfe der Ausgänge der ersten und der zweiten Clientanforderung koordiniert werden kann.

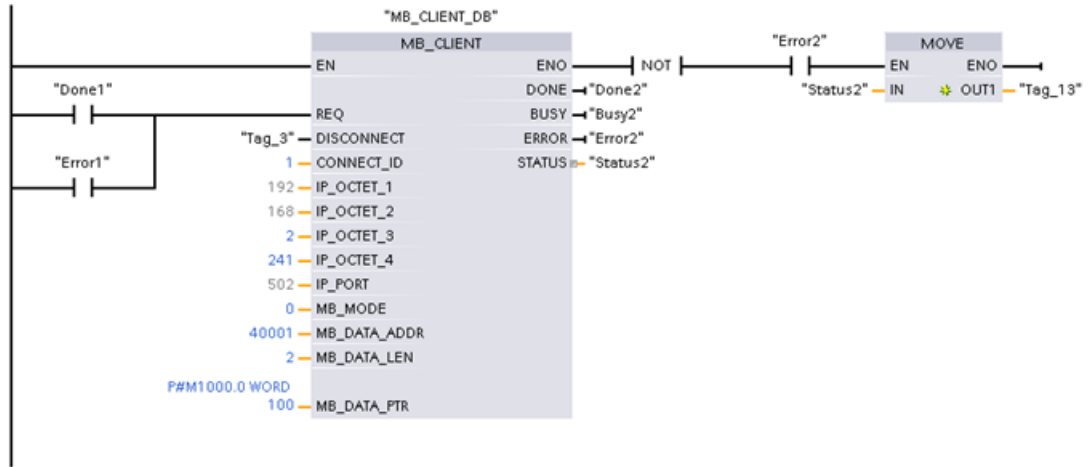
Das Beispiel zeigt beide Clients, die in denselben Speicherbereich schreiben. Außerdem wird ein ausgegebener Fehler erfasst (optionale Funktion).

Netzwerk 1: Modbus-Funktion 3 - Wörter im Haltereister lesen

13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)



Netzwerk 2: Modbus-Funktion 3 - Wörter im Halteregeister lesen



13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)

13.9.1 Übersicht

Vor STEP 7 V13 SP1 und den S7-1200 V4.1-CPU's hatten die Modbus RTU-Kommunikationsanweisungen andere Namen und teilweise auch etwas andere Schnittstellen. Für beide Anweisungsarten gelten die allgemeinen Konzepte. Informationen zur Programmierung finden sich in den Beschreibungen zu den einzelnen älteren Modbus RTU-Anweisungen.

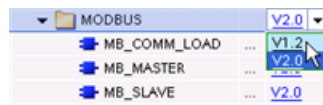
13.9.2 Version der Modbus RTU-Anweisungen auswählen

Es gibt zwei Versionen der alten Modbus RTU-Anweisungen in STEP 7:

- Alte Version 1.3: Kompatibel mit allen CPU- und CP-Varianten
- Alte Version 2.2: Kompatibel mit allen CPU- und CP-Varianten
(Hinweis: Version 2.2 bietet bei MB_COMM_LOAD die zusätzlichen Parameter REQ und DONE. In V2.2 gestattet der Parameter MB_ADDR bei MB_MASTER und MB_SLAVE nun einen UInt-Wert für die erweiterte Adressierung.)

Zum Zweck der Kompatibilität und zur einfacheren Migration können Sie wählen, welche Version der Anweisungen Sie in Ihr Anwenderprogramm einfügen.

Rufen Sie in der Taskcard "Anweisung" die MODBUS-Anweisungen in der Gruppe Kommunikationsprozessor auf.



Um die Version einer Modbus-Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus. Sie können die Gruppe oder einzelne Anweisungen auswählen.

Wenn Sie eine Modbus-Anweisung über das Anweisungsverzeichnis in Ihr Programm einfügen, wird in der Projektnavigation eine neue FB-Instanz angelegt. Die neue FB-Instanz finden Sie in der Projektnavigation unter PLC_x > Programmbausteine > Systembausteine > Programmressourcen.

Um die Version einer Modbus-Anweisung in einem Programm zu ermitteln, müssen Sie die Eigenschaften in der Projektnavigation aufrufen, nicht die Eigenschaften einer im Programmierer angezeigten Box. Wählen Sie in der Projektnavigation die Instanz eines Modbus-FBs aus, klicken Sie mit der rechten Maustaste, wählen Sie "Eigenschaften" und dann die Seite "Informationen", um die Versionsnummer der Modbus-Anweisung anzuzeigen.

13.9.3 Ältere Modbus RTU-Anweisungen

13.9.3.1 MB_COMM_LOAD (Port am PtP-Modul für Modbus RTU konfigurieren)

Tabelle Anweisung MB_COMM_LOAD
13-150

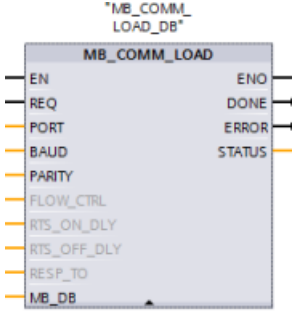
KOP/FUP	SCL	Beschreibung
	<pre>"MB_COMM_LOAD_DB" (REQ:=_bool_in_, PORT:=_uint_in_, BAUD:=_udint_in_, PARITY:=_uint_in_, FLOW_CTRL:=_uint_in_, RTS_ON_DLY:=_uint_in_, RTS_OFF_DLY:=_uint_in_, RESP_TO:=_uint_in_, DONE=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_DB:=_fbtref_inout_);</pre>	<p>Die Anweisung MB_COMM_LOAD konfiguriert einen PtP-Port für die Kommunikation über das Modbus-RTU-Protokoll. Hardwareoptionen des Modbus-Ports: Einbau von max. drei CMs (RS485 oder RS232), plus eines CB (R4845). Wenn Sie die Anweisung MB_COMM_LOAD in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen.</p>

Tabelle Datentypen für die Parameter
13-151

Parameter und Datentyp	Datentyp	Beschreibung	
REQ	IN	Bool	Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet. (Nur Version 2.0)
PORT	IN	Port	Nachdem Sie das CM oder CB eingebaut und konfiguriert haben, erscheint die Portkennung in der Parameter-Klappliste am Box-Anschluss PORT. Der zugewiesene CM- oder CB-Portwert ist die Eigenschaft "Hardwarekennung" der Gerätekonfiguration. Der symbolische Name des Ports wird im Register "Systemkonstanten" der PLC-Variablentabelle zugewiesen.
BAUD	IN	UDInt	Auswahl der Baudrate: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200. Alle anderen Werte sind ungültig.
PARITY	IN	UInt	Auswahl der Parität: <ul style="list-style-type: none"> 0 – Keine 1 – Ungerade 2 – Gerade
FLOW_CTRL ¹	IN	UInt	Auswahl der Flusskontrolle: <ul style="list-style-type: none"> 0 – (Standard) Keine Flusskontrolle 1 – Hardware-Flusskontrolle mit RTS immer EIN (gilt nicht bei RS485-Ports) 2 – Hardware-Flusskontrolle mit RTS geschaltet

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
RTS_ON_DLY ¹	IN	UInt	Auswahl RTS-Einschaltverzögerung: <ul style="list-style-type: none"> • 0 – (Standard) Keine Verzögerung von "RTS aktiv", bis das erste Zeichen der Meldung gesendet wird. • 1 bis 65535 – Verzögerung in Millisekunden von "RTS aktiv", bis das erste Zeichen der Meldung gesendet wird (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl FLOW_CTRL angewendet werden.
RTS_OFF_DLY ¹	IN	UInt	Auswahl RTS-Ausschaltverzögerung: <ul style="list-style-type: none"> • 0 – (Standard) Keine Verzögerung nach der Übertragung des letzten Zeichens bis "RTS inaktiv" • 1 bis 65535 – Verzögerung in Millisekunden von der Übertragung des letzten Zeichens bis "RTS inaktiv" (gilt nicht bei RS485-Ports). RTS-Verzögerungen müssen unabhängig von der Auswahl FLOW_CTRL angewendet werden.
RESP_TO ¹	IN	UInt	Antwort-Timeout: Zeit in Millisekunden, die MB_MASTER auf eine Antwort vom Slave wartet. Wenn der Slave während dieses Zeitraums nicht antwortet, wiederholt MB_MASTER die Anforderung oder beendet die Anforderung mit einem Fehler, wenn die angegebene Anzahl von Wiederholungen gesendet wurde. 5 ms bis 65535 ms (Standardwert = 1000 ms).
MB_DB	IN	Variant	Ein Verweis auf den Instanz-Datenbaustein der Anweisungen MB_MASTER oder MB_SLAVE. Nachdem Sie MB_SLAVE oder MB_MASTER in Ihr Programm eingefügt haben, ist die DB-Kennung in der Parameter-Klappliste am Anschluss der MB_DB-Box verfügbar.
DONE	OUT	Bool	Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde. (Nur Version 2.0)
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word	Ausführungsbedingung

¹ Optionale Parameter für MB_COMM_LOAD (V 2.x oder höher). Klicken Sie auf den Pfeil im unteren Bereich der KOP/FUP-Box, um die Box zu erweitern und diese Parameter aufzunehmen.

MB_COMM_LOAD wird ausgeführt, um einen Port für das Modbus-RTU-Protokoll zu konfigurieren. Nachdem ein Port für das Modbus-RTU-Protokoll konfiguriert ist, kann er nur von den Anweisungen MB_MASTER oder MB_SLAVE verwendet werden.

Für die Konfiguration jedes Kommunikationsports, der für die Modbus-Kommunikation eingesetzt wird, muss eine Ausführung von MB_COMM_LOAD genutzt werden. Sie müssen jedem Port, den Sie nutzen, einen eindeutigen Instanz-DB MB_COMM_LOAD zuweisen. Sie können bis zu drei Kommunikationsmodule (RS232 oder RS485) und ein Kommunikationsboard (RS485) in die CPU einbauen. Sie können MB_COMM_LOAD aus einem Anlauf-OB aufrufen und einmal ausführen, oder Sie können den Aufruf für die einmalige Ausführung mit Hilfe des Systemmerkers für den ersten Zyklus (Seite 92) initiieren. Führen Sie MB_COMM_LOAD nur dann erneut aus, wenn Kommunikationsparameter wie Baudrate oder Parität geändert werden müssen.

Wenn Sie MB_MASTER oder MB_SLAVE in Ihr Programm einfügen, wird der Anweisung ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein wird referenziert, wenn Sie in der Anweisung MB_COMM_LOAD den Parameter MB_DB angeben.

MB_COMM_LOAD Datenbausteinvariablen

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von MB_COMM_LOAD, die in Ihrem Programm verwendet werden können.

Tabelle 13-152 Statische Variablen im Instanz-DB

Variable	Datentyp	Beschreibung
ICHAR_GAP	UInt	Verzögerung für den Zeichenabstand zwischen den Zeichen. Dieser Parameter wird in Millisekunden angegeben und dient dazu, den erwarteten Zeitraum zwischen den empfangenen Zeichen zu erhöhen. Die entsprechende Anzahl Bitzeiten für diesen Parameter wird zum Modbus-Standardwert von 35 Bitzeiten (3,5 Zeichenzeiten) addiert.
RETRIES	UInt	Die Anzahl der wiederholten Versuche, die der Master durchführt, bevor der Fehlercode 0x80C8 für "Keine Antwort" zurückgegeben wird.
STOP_BITS	USInt	Anzahl Stoppbits für jeden Frame eines Zeichens. Gültig sind die Werte 1 und 2.

Tabelle 13-153 Bedingungscode der Ausführung von MB_COMM_LOAD ¹

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
8180	Ungültiger Wert der Port-ID (falscher Port/falsche Hardwarekennung des Kommunikationsmoduls)
8181	Ungültiger Wert für die Baudrate.
8182	Ungültiger Wert für die Parität.
8183	Ungültiger Wert für die Flusskontrolle.
8184	Ungültiger Wert für die Timeout-Zeit der Antwort (Antwort-Timeout ist kleiner als der Mindestwert von 5 ms)
8185	Der Parameter MB_DB ist kein Instanz-Datenbaustein einer Anweisung MB_MASTER oder MB_SLAVE.

¹ Neben den oben aufgeführten Fehlern der Anweisung MB_COMM_LOAD können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

13.9.3.2 MB_MASTER (Über den PtP-Port als Modbus RTU-Master kommunizieren)

Tabelle 13-154 Anweisung MB_MASTER

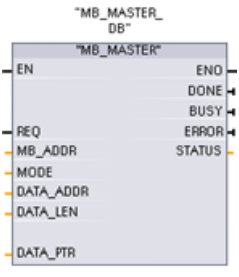
KOP/FUP	SCL	Beschreibung
	<pre>"MB_MASTER_DB" (REQ:=_bool_in_, MB_ADDR:=_uint_in_, MODE:=_usint_in_, DATA_ADDR:=_udint_in_, DATA_LEN:=_uint_in_, DONE=>_bool_out_, BUSY=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, DATA_PTR:=_variant_inout_);</pre>	<p>Die Anweisung MB_MASTER kommuniziert als Modbus-Master über einen Port, der von einer vorherigen Ausführung der Anweisung MB_COMM_LOAD konfiguriert wurde. Wenn Sie die Anweisung MB_MASTER in Ihr Programm einfügen, wird automatisch ein Instanz-Datenbaustein zugewiesen. Dieser Instanz-Datenbaustein MB_MASTER wird verwendet, wenn Sie in der Anweisung MB_COMM_LOAD den Parameter MB_DB angeben.</p>

Tabelle 13-155 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Beschreibung
REQ	IN	Bool 0 = Keine Anforderung 1 = Anforderung, Daten an den Modbus-Slave zu senden
MB_ADDR	IN	V1.0: USInt V2.0: UInt Modbus-RTU-Stationsadresse: Standard-Adressierungsbereich (1 bis 247) Erweiterter Adressierungsbereich (1 bis 65535) Der Wert 0 ist für den Broadcast einer Meldung an alle Modbus-Slaves reserviert. Für den Broadcast werden nur die Modbus-Funktionscodes 05, 06, 15 und 16 unterstützt.
MODE	IN	USInt Auswahl Modus: Gibt die Art der Anforderung an (Lesen, Schreiben oder Diagnose). Weitere Informationen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_ADDR	IN	UDInt Anfangsadresse im Slave: Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll. Die gültigen Adressen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_LEN	IN	UInt Datenlänge: Gibt die Anzahl der Bits oder Wörter an, auf die diese Anforderung zugreifen soll. Die gültigen Längen finden Sie in der Tabelle der Modbus-Funktionen unten.
DATA_PTR	IN	Variant Daten-Pointer: Zeigt auf die M- oder DB-Adresse (nicht optimierter DB) für die zu schreibenden oder zu lesenden Daten.
DONE	OUT	Bool Das Bit DONE ist einen Zyklus lang WAHR, nachdem die letzte Anforderung fehlerfrei ausgeführt wurde.
BUSY	OUT	Bool • 0 – Keine Anweisung MB_MASTER in Bearbeitung • 1 – Anweisung MB_MASTER in Bearbeitung
ERROR	OUT	Bool Das Bit ERROR ist einen Zyklus lang WAHR, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Parameter STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = WAHR ist.
STATUS	OUT	Word Ausführungsbedingung

Regeln für die Kommunikation des Modbus-Masters

- MB_COMM_LOAD muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung MB_MASTER mit diesem Port kommunizieren kann.
- Wenn ein Port zum Initiieren von Modbus-Masteranforderungen verwendet werden soll, darf dieser Port nicht von MB_SLAVE verwendet werden. Eine oder mehrere Instanzen von MB_MASTER können mit diesem Port verwendet werden. Doch alle Ausführungen von MB_MASTER müssen den gleichen Instanz-DB MB_MASTER für den Port verwenden.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss die Anweisung MB_MASTER auf abgeschlossene Sende- und Empfangsvorgänge abfragen.
- Rufen Sie alle Ausführungen von MB_MASTER für einen bestimmten Port aus einem Programmzyklus-OB auf. Modbus-Master-Anweisungen können nur in jeweils einem Programmzyklus oder in jeweils einer zyklischen/zeitverzögerten Ausführungsschicht ausgeführt werden. Sie dürfen nicht in beiden Prioritätsschichten der Ausführung bearbeitet werden. Die Vorrangunterbrechung einer Modbus-Master-Anweisung durch eine andere Modbus-Master-Anweisung in einer Ausführungsschicht mit höherer Priorität führt zu nicht ordnungsgemäßem Betrieb. Die Modbus-Master-Anweisungen dürfen nicht in Anlauf-, Diagnose- oder Zeitfehlerausführungsschichten bearbeitet werden.
- Nachdem eine Masteranweisung eine Übertragung ausgelöst hat, muss diese Instanz mit aktiviertem Eingang EN fortlaufend ausgeführt werden, bis der Zustand DONE = 1 oder ERROR = 1 ausgegeben wird. Eine bestimmte MB_MASTER-Instanz wird als aktiv betrachtet, bis eines dieser beiden Ereignisse auftritt. Während die ursprüngliche Instanz aktiv ist, führt jeder Aufruf einer weiteren Instanz mit aktiviertem Eingang REQ zu einem Fehler. Wenn die fortlaufende Ausführung der ursprünglichen Instanz stoppt, bleibt der Anforderungszustand für die von der statischen Variable Blocked_Proc_Timeout angegebene Zeitdauer aktiv. Nach Ablauf dieser Zeitdauer wird die nächste aufgerufene Master-Anweisung mit aktiviertem Eingang REQ zur aktiven Instanz. Dies verhindert, dass eine einzige Modbus-Master-Anweisung ein Monopol inne hat oder den Zugriff auf den Port sperrt. Wenn die ursprüngliche aktive Instanz nicht während des von der statischen Variablen "Blocked_Proc_Timeout" angegebenen Zeitraums aktiviert wird, löscht die nächste Ausführung durch diese Instanz (ohne gesetzten Eingang REQ) den aktiven Zustand. Falls (REQ gesetzt ist), löst diese Ausführung eine neue Master-Anforderung aus, als wenn keine andere Instanz aktiv wäre.

Parameter REQ

0 = Keine Anforderung; 1 = Anforderung, Daten an den Modbus-Slave zu senden

Sie können diesen Eingang über einen pegel- oder flankengesteuerten Kontakt steuern. Immer wenn dieser Eingang aktiviert wird, wird eine Zustandsmaschine gestartet, um sicherzustellen, dass eine andere Anweisung MB_MASTER, die denselben Instanz-DB verwendet, erst dann eine Anforderung abgeben kann, wenn die aktuelle Anforderung abgearbeitet ist. Alle anderen Eingangszustände werden erfasst und intern für die aktuelle Anforderung gespeichert, bis die Antwort empfangen oder ein Fehler erkannt wird.

Wird dieselbe Instanz von MB_MASTER erneut mit dem REQ-Eingang = 1 ausgeführt, bevor die aktuelle Anforderung abgearbeitet ist, werden keine nachfolgenden Übertragungen durchgeführt. Wenn die Anforderung jedoch abgearbeitet ist, wird zu dem Zeitpunkt, zu dem MB_MASTER erneut mit REQ-Eingang = 1 ausgeführt wird, eine neue Anforderung abgegeben.

Über die Parameter DATA_ADDR und MODE wählen Sie den Modbus-Funktionsstyp aus.

DATA_ADDR (Modbus-Anfangsadresse im Slave): Gibt die Anfangsadresse der Daten an, auf die im Modbus-Slave zugegriffen werden soll.

Die Anweisung MB_MASTER nutzt statt eines Funktionscodeeingangs einen Eingang MODE. Die Kombination aus MODE und Modbus-Adresse legt den Funktionscode fest, der in der eigentlichen Modbus-Meldung verwendet wird. Die folgende Tabelle zeigt die Entsprechung zwischen dem Parameter MODE, dem Modbus-Funktionscode und dem Modbus-Adressbereich.

Tabelle
13-156 Modbus-Funktionen

MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	Modbus-Adresse
0	01	1 bis 2000 1 bis 1992 ¹	Ausgangsbits lesen: 1 bis (1992 oder 2000) Bit pro Anforderung	1 bis 9999
0	02	1 bis 2000 1 bis 1992 ¹	Eingangsbits lesen: 1 bis (1992 oder 2000) Bit pro Anforderung	10001 bis 19999
0	03	1 bis 125 1 bis 124 ¹	Halteregister lesen: 1 bis (124 oder 125) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
0	04	1 bis 125 1 bis 124 ¹	Eingangswörter lesen: 1 bis (124 oder 125) Bits pro Anforderung	30001 bis 39999
1	05	1	Ein Ausgangsbit schreiben: Ein Bit pro Anforderung	1 bis 9999
1	06	1	Ein Halteregister schreiben: 1 Wort pro Anforderung	40001 bis 49999 oder 400001 bis 465535
1	15	2 bis 1968 2 bis 1960 ¹	Mehrere Ausgangsbits schreiben: 2 bis (1960 oder 1968) Bit pro Anforderung	1 bis 9999
1	16	2 bis 123 2 bis 122 ¹	Mehrere Halteregister schreiben: 2 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
2	15	1 bis 1968 2 bis 1960 ¹	Ein oder mehrere Ausgangsbits schreiben: 1 bis (1960 oder 1968) Bit pro Anforderung	1 bis 9999
2	16	1 bis 123 1 bis 122 ¹	Ein oder mehrere Halteregister schreiben: 1 bis (122 oder 123) Wörter pro Anforderung	40001 bis 49999 oder 400001 bis 465535
11	11	0	Statuswort und Ereigniszähler der Slave-Kommunikation lesen. Das Statuswort zeigt beschäftigt an (0 – nicht beschäftigt, 0xFFFF - beschäftigt). Der Ereigniszähler wird bei jeder erfolgreichen Abarbeitung einer Meldung inkrementiert. Beide Operanden DATA_ADDR und DATA_LEN von MB_MASTER werden bei dieser Funktion ignoriert.	
80	08	1	Slave-Zustand über Daten-Diagnosecode 0x0000 prüfen (Prüfschleifentest – Slave gibt ein Echo der Anforderung zurück) 1 Wort pro Anforderung	

MODE	Modbus-Funktion	Länge der Daten	Operation und Daten	Modbus-Adresse
81	08	1	Slave-Ereigniszähler über Daten-Diagnosecode 0x000A zurücksetzen 1 Wort pro Anforderung	
3 bis 10, 12 bis 79, 82 bis 255			Reserviert	

¹ Bei der erweiterten Adressierung ist die maximale Datenlänge je nach Datentyp der Funktion um 1 Byte oder 1 Wort geringer.

Parameter DATA_PTR

Der Parameter DATA_PTR zeigt auf die DB- oder M-Adresse, in die geschrieben oder aus der gelesen wird. Wenn Sie einen Datenbaustein verwenden, müssen Sie einen globalen Datenbaustein anlegen, der den Datenspeicher für Lese- und Schreibvorgänge auf Modbus-Slaves bereitstellt.

Hinweis

Der Datenbaustein DATA_PTR muss die direkte Adressierung zulassen

Der Datenbaustein muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

Datenbausteinstrukturen für den Parameter DATA_PTR

- Diese Datentypen gelten für das **Lesen von Wörtern** der Modbus-Adressen 30001 bis 39999, 40001 bis 49999 und 400001 bis 465536 sowie für das **Schreiben von Wörtern** in die Modbus-Adressen 40001 bis 49999 und 400001 bis 465536.
 - Standard-Array der Datentypen WORD, UINT oder INT
 - Benannte Struktur des Typs WORD, UINT oder INT, bei der jedes Element einen eindeutigen Namen und einen 16-Bit-Datentyp hat.
 - Benannte komplexe Struktur, bei der jedes Element einen eindeutigen Namen und einen 16- oder 32-Bit-Datentyp hat.
- Zum **Lesen** und Schreiben von Bits für die Modbus-Adressen 00001 bis 09999 und zum Lesen von Bits von 10001 bis 19999.
 - Standardfeld aus Booleschen Datentypen.
 - Benannte Boolesche Struktur aus eindeutig benannten Booleschen Variablen.

- Es ist zwar nicht erforderlich, doch empfehlenswert, dass jede Anweisung MB_MASTER einen eigenen getrennten Speicherbereich hat. Grund hierfür ist, dass die Möglichkeit der Datenzerstörung größer ist, wenn mehrere Anweisungen MB_MASTER in demselben Speicherbereich lesen und schreiben.
- Es ist nicht erforderlich, dass die Datenbereiche für DATA_PTR in demselben globalen Datenbaustein liegen. Sie können einen Datenbaustein mit mehreren Bereichen für Modbus-Lesevorgänge, einen Datenbaustein für Modbus-Schreibvorgänge oder einen Datenbaustein für jede Slave-Station anlegen.

Variablen im Datenbaustein des Modbus-Masters

Die folgende Tabelle zeigt die öffentlichen statischen Variablen im Instanz-DB von MB_MASTER, die in Ihrem Programm verwendet werden können.

Tabelle 13-157 Statische Variablen im Instanz-DB

Variable	Datentyp	Anfangswert	Beschreibung
Blocked_Proc_Timeout	Real	3.0	Zeitdauer (in Sekunden), die auf eine blockierte Modbus-Master-Instanz gewartet werden soll, bevor diese Instanz als AKTIV entfernt wird. Dies kann beispielsweise vorkommen, wenn eine Master-Anforderung ausgegeben wurde und das Programm dann aufhört, die Masterfunktion aufzurufen, bevor es die Anforderung vollständig beendet hat. Der Zeitwert muss größer als 0 und kleiner als 55 Sekunden sein, ansonsten tritt ein Fehler auf. Der Standardwert ist 0,5 Sekunden.
Extended_Addressing	Bool	Falsch	Konfiguriert die Slave-Adressierung als einzelnes oder doppeltes Byte. Der Standardwert ist 0. (0 = Ein-Byte-Adresse, 1 = Zwei-Byte-Adresse)

Ihr Programm kann in die Variablen Blocked_Proc_Timeout und Extended_Addressing Werte schreiben, um die Modbus-Master-Operationen zu steuern. Im Abschnitt zur Anweisung MB_SLAVE finden Sie in der Beschreibung der Variablen HR_Start_Offset und Extended_Addressing ein Beispiel dafür, wie Sie diese Variablen im Programmiereditor nutzen, sowie weitere Einzelheiten zu erweiterten Modbus-Adressierung (Seite 1175).

Bedingungscode

Tabelle 13-158 Bedingungscode für die Ausführung von MB_MASTER (Kommunikations- und Konfigurationsfehler) ¹

STATUS (W#16#)	Beschreibung
0000	Kein Fehler
80C8	Slave-Timeout. Prüfen Sie Baudrate, Parität und Verdrahtung des Slave.
80D1	Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert. Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.
80D2	Die Sende-anforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.

13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)

STATUS (W#16#)	Beschreibung
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung MB_COMM_LOAD
8186	Ungültige Modbus-Stationsadresse
8188	Ungültiger Modus für die Broadcast-Anforderung angegeben
8189	Ungültiger Wert für die Datenadresse.
818A	Ungültiger Wert für die Datenlänge.
818B	Ungültiger Pointer auf die lokale Datenquelle bzw. das lokale Datenziel: Größe nicht richtig
818C	Ungültiger Pointer für DATA_PTR oder Blocked_Proc_Timeout ungültig: Bei dem Datenbereich muss es sich um einen DB (der die symbolische und direkte Adressierung gestattet) oder um den Speicherbereich der Merker handeln.
8200	Port ist durch Bearbeitung einer Sendeanforderung belegt.

Tabelle 13-159 Bedingungs-codes für die Ausführung von MB_MASTER (Modbus-Protokollfehler) ¹

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	-	CRC-Fehler
8381	01	Funktionscode nicht unterstützt
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	Größer als 03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)
8386	-	Funktionscode in der Antwort entspricht nicht dem Code in der Anforderung.
8387	-	Der falsche Slave hat reagiert.
8388	-	Die Slave-Antwort auf eine Schreibanforderung ist inkorrekt. Die vom Slave ausgegebene Schreibanforderung entspricht nicht dem, was der Master gesendet hat.

¹ Neben den oben aufgeführten Fehlern der Anweisung MB_MASTER können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

13.9.3.3 MB_SLAVE (Über den PtP-Port als Modbus RTU-Slave kommunizieren)

Tabelle Anweisung MB_SLAVE
13-160

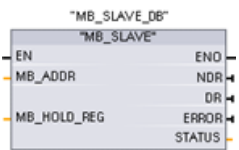
KOP/FUP	SCL	Beschreibung
	<pre>"MB_SLAVE_DB" (MB_ADDR:=_uint_in_, NDR=>_bool_out_, DR=>_bool_out_, ERROR=>_bool_out_, STATUS=>_word_out_, MB_HOLD_REG:=_variant_inout_);</pre>	<p>Mit der Anweisung MB_SLAVE kann Ihr Programm über einen PtP-Port eines CM (RS485 oder RS232) oder eines CB (RS485) als Modbus-Slave kommunizieren. Wenn ein dezentraler Modbus-RTU-Master eine Anforderung ausgibt, reagiert Ihr Anwenderprogramm auf die Anforderung, indem es die Anweisung MB_SLAVE ausführt. STEP 7 erstellt automatisch eine Instanz-DB, wenn Sie die Anweisung einfügen. Sie geben mit diesem MB_SLAVE_DB-Namen den Parameter MB_DB für die Anweisung MB_COMM_LOAD an.</p>

Tabelle Datentypen für die Parameter
13-161

Parameter und Datentyp	Datentyp	Beschreibung
MB_ADDR	IN	V1.0: USInt V2.0: UInt
MB_HOLD_REG	IN	Variant
NDR	OUT	Bool
DR	OUT	Bool
ERROR	OUT	Bool
STATUS	OUT	Word

Die Funktionscodes der Modbus-Kommunikation (1, 2, 4, 5 und 15) können Bits und Wörter direkt im Prozessabbild der Eingänge und im Prozessabbild der Ausgänge der CPU lesen und schreiben. Bei diesen Funktionscodes muss der Parameter MB_HOLD_REG als Datentyp größer

13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)

als ein Byte definiert werden. Die folgende Tabelle zeigt die Beispielzuordnung der Modbus-Adressen zum Prozessabbild in der CPU.

Tabelle 13-162 Zuordnung der Modbus-Adressen zum Prozessabbild

Modbus-Funktionen						S7-1200	
Codes	Funktion	Datenbereich	Adressbereich			Datenbereich	CPU-Adresse
01	Bits lesen	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
02	Bits lesen	Eingang	10001	bis	18192	Prozessabbild der Eingänge	E0.0 bis E1023.7
04	Wörter lesen	Eingang	30001	bis	30512	Prozessabbild der Eingänge	EW0 bis EW1022
05	Bit schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7
15	Bits schreiben	Ausgang	1	bis	8192	Prozessabbild der Ausgänge	A0.0 bis A1023.7

Die Funktionscodes der Modbus-Kommunikation (3, 6, 16) nutzen ein Modbus-Halteregister, bei dem es sich um einen Adressbereich im Speicherbereich der Merker oder um einen Datenbaustein handeln kann. Der Typ des Halteregisters wird vom Parameter MB_HOLD_REG der Anweisung MB_SLAVE angegeben.

Hinweis

Typ des Datenbausteins MB_HOLD_REG

Ein Datenbaustein mit Modbus-Halteregister muss die direkte (absolute) und die symbolische Adressierung zulassen. Wenn Sie den Datenbaustein anlegen, müssen Sie das Zugriffsattribut "Standard" auswählen.

Die folgende Tabelle zeigt Beispiele für die Zuordnung von Modbus-Adressen zum Halteregister, das für die Modbus-Funktionscodes 03 (Wörter lesen), 06 (Wort schreiben) und 16 (Wörter schreiben) verwendet wird. Die tatsächliche obere Grenze der DB-Adressen wird vom maximalen Arbeitsspeicher und vom maximalen M-Speicher des jeweiligen CPU-Modells festgelegt.

Tabelle 13-163 Zuordnung von Modbus-Adressen zum CPU-Speicher

Modbus-Master-Adresse	Beispiele für Parameter von MB_HOLD_REG				
	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff
40001	MW100	DB10.DBW0	MW120	DB10.DBW50	"Rezept".Inhaltsstoff[1]
40002	MW102	DB10.DBW2	MW122	DB10.DBW52	"Rezept".Inhaltsstoff[2]
40003	MW104	DB10.DBW4	MW124	DB10.DBW54	"Rezept".Inhaltsstoff[3]
40004	MW106	DB10.DBW6	MW126	DB10.DBW56	"Rezept".Inhaltsstoff[4]
40005	MW108	DB10.DBW8	MW128	DB10.DBW58	"Rezept".Inhaltsstoff[5]

Tabelle 13-164 Diagnosefunktionen

Modbus-Diagnosefunktionen von MB_SLAVE der S7-1200		
Codes	Teilfunktion	Beschreibung
08	0000H	Echotest Abfragedaten ausgeben: Die Anweisung MB_SLAVE gibt einem Modbus-Master das Echo eines empfangenen Datenworts zurück.
08	000AH	Kommunikationsereigniszähler löschen: Die Anweisung MB_SLAVE löscht den Kommunikationsereigniszähler, der für Modbus-Funktion 11 verwendet wird.
11		Kommunikationsereigniszähler abrufen: Die Anweisung MB_SLAVE nutzt einen internen Kommunikationsereigniszähler, um die Anzahl erfolgreicher Modbus-Lese- und Modbus-Schreibanforderungen, die an den Modbus-Slave gesendet werden, zu erfassen. Der Zähler wird bei Funktion 8, Funktion 11 und Broadcast-Anforderungen nicht hochgezählt. Außerdem wird bei Anforderungen, die zu Kommunikationsfehlern führen (z.B. Paritäts- oder CRC-Fehler), nicht hochgezählt.

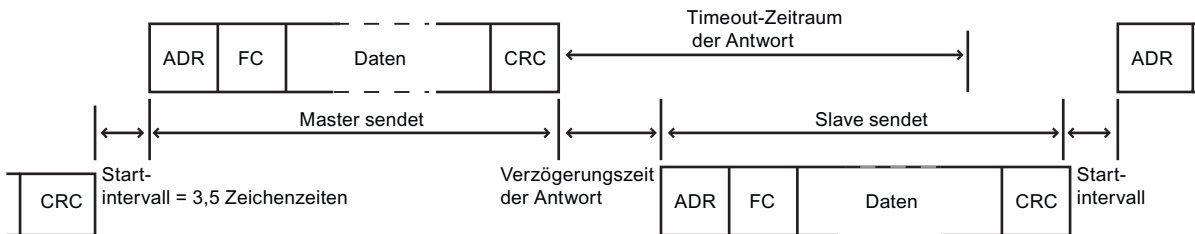
Die Anweisung MB_SLAVE unterstützt Broadcast-Schreibanforderungen von Modbus-Mastern, solange die Anforderungen den Zugriff auf gültige Adressen umfassen. Für Funktionscodes, die bei der Broadcast-Funktion nicht unterstützt werden, erzeugt MB_SLAVE den Fehlercode 0x8188.

Regeln für die Modbus-Slave-Kommunikation

- MB_COMM_LOAD muss ausgeführt werden, um einen Port zu konfigurieren, damit die Anweisung MB_SLAVE über diesen Port kommunizieren kann.
- Wenn ein Port als Slave auf einen Modbus-Master reagieren soll, darf dieser Port nicht mit der Anweisung MB_MASTER programmiert werden.
- Nur eine Instanz von MB_SLAVE kann mit einem bestimmten Port verwendet werden, andernfalls kann unerwartetes Verhalten auftreten.
- Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss den Kommunikationsprozess durch Abfrage der Anweisung MB_SLAVE auf abgeschlossene Sende- und Empfangsvorgänge steuern.
- Die Anweisung MB_SLAVE muss regelmäßig mit einer Häufigkeit ausgeführt werden, die eine zeitnahe Antwort auf eingehende Anforderungen eines Modbus-Masters ermöglicht. Es wird empfohlen, MB_SLAVE in jedem Zyklus aus einem Programmzyklus-OB auszuführen. MB_SLAVE kann aus einem Weckalarm-OB ausgeführt werden, doch ist dies nicht empfehlenswert, weil übermäßige Zeitverzögerungen im Interruptprogramm die Ausführung anderer Interruptprogramme temporär blockieren können.

Zeitsteuerung des Modbus-Signals

MB_SLAVE muss regelmäßig ausgeführt werden, um jede Anforderung des Modbus-Masters zu empfangen und entsprechend zu antworten. Die Häufigkeit der Ausführung von MB_SLAVE richtet sich nach dem vom Modbus-Master vorgegebenen Timeout-Wert für die Antwort. Dies ist in der nachstehenden Abbildung dargestellt.



Der Timeout-Zeitraum für die Antwort RESP_TO ist die Zeitdauer, die ein Modbus-Master auf den Beginn einer Antwort von einem Modbus-Slave wartet. Dieser Zeitraum wird nicht vom Modbus-Protokoll definiert, sondern von einem Parameter des jeweiligen Modbus-Masters. Die Häufigkeit der Ausführung (die Zeit zwischen einer Ausführung und der nächsten Ausführung) von MB_SLAVE muss sich nach den jeweiligen Parametern des Modbus-Masters richten. Sie sollten MB_SLAVE mindestens zweimal während des Timeout-Zeitraums für die Antwort des Modbus-Masters ausführen.

Variablen des Modbus-Slave

Diese Tabelle zeigt die öffentlichen statischen Variablen im Instanz-Datenbaustein von MB_SLAVE, die in Ihrem Programm verwendet werden können.

Tabelle 13-165 Variablen des Modbus-Slave

Variable	Datentyp	Beschreibung
Request_Count	Word	Die Anzahl aller von diesem Slave empfangenen Anforderungen
Slave_Message_Count	Word	Die Anzahl der für diesen spezifischen Slave empfangenen Anforderungen
Bad_CRC_Count	Word	Die Anzahl der empfangenen Anforderungen, die einen CRC-Fehler aufweisen
Broadcast_Count	Word	Die Anzahl der empfangenen Broadcast-Anforderungen
Exception_Count	Word	Modbus-spezifische Fehler, die eine zurückgegebene Ausnahme erfordern
Success_Count	Word	Die Anzahl der für diesen spezifischen Slave empfangenen Anforderungen ohne Protokollfehler
HR_Start_Offset	Word	Gibt die Anfangsadresse des Modbus-Halteregisters an (Standard = 0)
Extended_Addresssing	Bool	Konfiguriert die Slave-Adressierung als einzelnes oder doppeltes Byte (0 = einzelne Byte-Adresse, 1 = doppelte Byte-Adresse, Standard = 0)

Ihr Programm kann in die Variablen HR_Start_Offset und Extended_Addresssing Werte schreiben und die Modbus-Slave-Operationen steuern. Die anderen Variablen können gelesen werden, um den Modbus-Zustand zu überwachen.

HR_Start_Offset

Die Adressen des Modbus-Halteregisters beginnen bei 40001 oder 400001. Diese Adressen entsprechen der Anfangsadresse des Halteregisters im Zielsystemspeicher. Sie können jedoch die Variable HR_Start_Offset konfigurieren, um eine andere Anfangsadresse als 40001 oder 400001 für das Modbus-Halteregister zu konfigurieren.

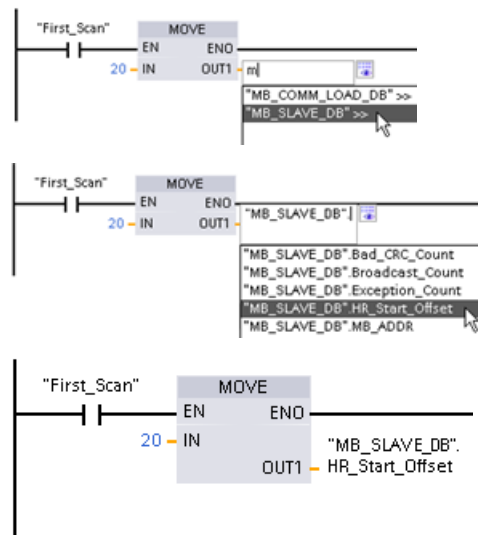
Sie können z. B. ein Halteregister mit Beginn an MW100 und einer Länge von 100 Wörtern konfigurieren. Mit einem Versatz von 20 geben Sie eine Anfangsadresse des Halteregisters von 40021 statt 40001 an. Jede Adresse unter 40021 und über 400119 führt zu einem Adressierungsfehler.

Tabelle Beispiel für die Adressierung des Modbus-Halteregisters
13-166

HR_Start_Offset	Adresse	Minimum	Maximum
0	Modbus-Adresse (Wort)	40001	40099
	S7-1200 Adresse	MW100	MW298
20	Modbus-Adresse (Wort)	40021	40119
	S7-1200 Adresse	MW100	MW298

HR_Start_Offset ist ein Wortwert, der die Anfangsadresse des Modbus-Halteregisters angibt und im Instanz-Datenbaustein MB_SLAVE gespeichert ist. Sie können diese öffentliche statische Variable über die Parameter-Klappliste auswählen, nachdem Sie MB_SLAVE in Ihr Programm eingefügt haben.

Wenn Sie beispielsweise MB_SLAVE in ein KOP-Netzwerk eingefügt haben, können Sie in ein vorheriges Netzwerk gehen und den Wert HR_Start_Offset zuweisen. Der Wert muss vor der Ausführung von MB_SLAVE zugewiesen werden.



Modbus-Slave-Variable über den Standard-DB-Namen eingeben:

1. Positionieren Sie den Cursor im Parameterfeld und geben Sie das Zeichen m ein.
2. Wählen Sie in der Klappliste "MB_SLAVE_DB" aus.
3. Positionieren Sie den Cursor rechts vom DB-Namen (nach den Anführungszeichen) und geben Sie einen Punkt ein.
4. Wählen Sie in der Klappliste "MB_SLAVE_DB.HR_Start_Offset" aus.

Extended Addressing

Auf die Extended Addressing-Variable wird in ähnlicher Weise zugegriffen wie auf die oben beschriebene HR_Start_Offset-Referenz, außer dass es sich bei der Extended Addressing-Variable um einen Booleschen Wert handelt. Der Boolesche Wert muss von einer Ausgangsspule und kann nicht von einer Box "Verschieben" geschrieben werden.

Für die Modbus-Slave-Adressierung kann ein einzelnes Byte (dies ist der Modbus-Standard) oder ein doppeltes Byte konfiguriert werden. Die erweiterte Adressierung dient zur Adressierung von mehr als 247 Geräten in einem einzigen Netzwerk. Wenn Sie sich für die erweiterte Adressierung entscheiden, können Sie maximal 64.000 Adressen ansprechen. Im Folgenden wird ein Telegramm der Modbus-Funktion 1 als Beispiel gezeigt.

Tabelle 13-167 Slave-Adresse mit einem Byte (Byte 0)

Funktion 1	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	
Anforderung	Slave-Adresse	F-Code	Anfangsadresse		Länge der Spulen		
Gültige Antwort	Slave-Adresse	F-Code	Länge	Spulendaten			
Fehlerantwort	Slave-Adresse	0x81	E-Code				

Tabelle 13-168 Slave-Adresse mit zwei Bytes (Byte 0 und Byte 1)

	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
Anforderung	Slave-Adresse		F-Code	Anfangsadresse		Länge der Spulen	
Gültige Antwort	Slave-Adresse		F-Code	Länge	Spulendaten		
Fehlerantwort	Slave-Adresse		0x81	E-Code			

BedingungsCodes

Tabelle 13-169 BedingungsCodes für die Ausführung von MB_SLAVE (Kommunikations- und Konfigurationsfehler) ¹

STATUS (W#16#)	Beschreibung
80D1	Der Empfänger hat eine Flusskontrolle zur Unterbrechung einer laufenden Sendung angefordert, aber die Sendung innerhalb der Wartezeit nicht wieder aktiviert. Dieser Fehler wird auch bei der Hardware-Flusskontrolle erzeugt, wenn der Empfänger innerhalb der Wartezeit kein CTS erkennt.
80D2	Die Sendeaufforderung wurde abgebrochen, weil vom DCE kein DSR-Signal empfangen wird.
80E0	Die Meldung wurde beendet, weil der Empfangspuffer voll ist.
80E1	Die Meldung wurde wegen eines Paritätsfehlers beendet.
80E2	Die Meldung wurde wegen eines Telegrammfehlers beendet.
80E3	Die Meldung wurde wegen eines Überlauffehlers beendet.
80E4	Die Meldung wurde abgebrochen, weil die angegebene Länge die Gesamtpuffergröße überschreitet.

STATUS (W#16#)	Beschreibung
8180	Ungültiger Wert für die Port-ID oder Fehler bei der Anweisung MB_COMM_LOAD
8186	Ungültige Modbus-Stationsadresse
8187	Ungültiger Pointer auf MB_HOLD_REG-DB: Bereich ist zu klein
818C	Ungültiger Pointer MB_HOLD_REG auf M-Speicher oder DB (DB-Bereich muss symbolische und direkte Adressen zulassen)

Tabelle 13-170 Bedingungs-codes für die Ausführung von MB_SLAVE (Modbus-Protokollfehler) ¹

STATUS (W#16#)	Antwortcode vom Slave	Modbus-Protokollfehler
8380	Keine Antwort	CRC-Fehler
8381	01	Funktioncode nicht unterstützt oder nicht unterstützt in Broadcasts
8382	03	Fehler in der Datenlänge
8383	02	Datenadressfehler oder Adresse außerhalb des gültigen Bereichs von DATA_PTR
8384	03	Fehler im Datenwert
8385	03	Wert des Datendiagnosecodes wird nicht unterstützt (Funktionscode 08)

¹ Neben den oben aufgeführten Fehlern der Anweisung MB_SLAVE können auch die zugrunde liegenden PtP-Kommunikationsanweisungen Fehler zurückgeben.

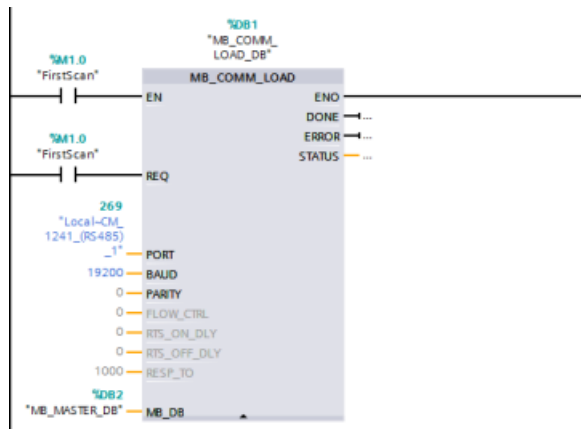
13.9.4 Beispiel für Modbus RTU in älteren Systemen

13.9.4.1 Beispiel: Beispielprogramm für einen Modbus RTU-Master in älteren Systemen

MB_COMM_LOAD wird während des Anlaufs über den Merker des ersten Zyklus initialisiert. Die Ausführung von MB_COMM_LOAD auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports während der Laufzeit nicht ändert.

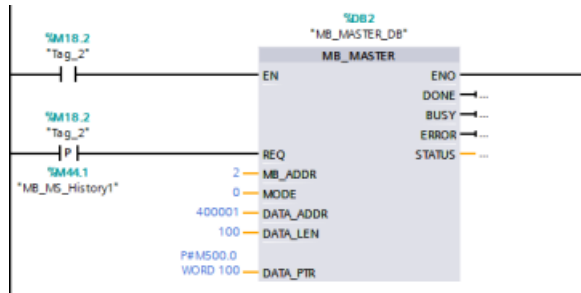
Netzwerk 1: Kommunikationsport des RS485-Moduls nur einmal während des ersten Zyklus konfigurieren/initialisieren.

13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)



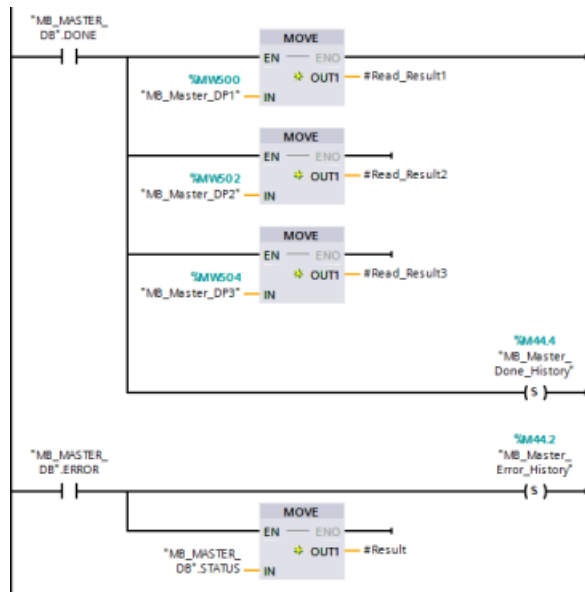
Eine Anweisung MB_MASTER dient im Programmzyklus-OB zur Kommunikation mit einem Slave. Weitere Anweisungen MB_MASTER können im Programmzyklus-OB für die Kommunikation mit anderen Slaves genutzt werden, oder es kann ein MB_MASTER-FB wiederverwendet werden, um mit weiteren Slaves zu kommunizieren.

Netzwerk 2: 100 Wörter an Haltereisterdaten von Adresse 400001 in Slave Nr. 2 in Adresse MW500-MW698 auslesen.

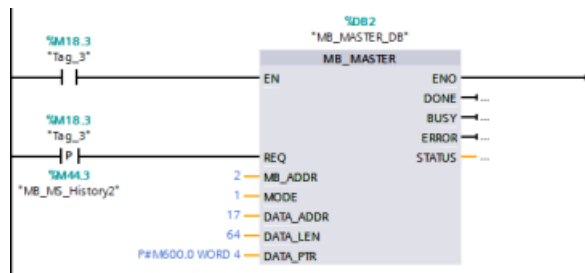


Netzwerk 3: Die ersten drei Wörter der an eine andere Adresse ausgelesenen Haltereisterdaten verschieben und ein Verlaufsbit DONE setzen. Dieses Netzwerk setzt bei Auftreten eines Fehlers auch ein Verlaufsbit ERROR und speichert das Wort STATUS an einer anderen Adresse.

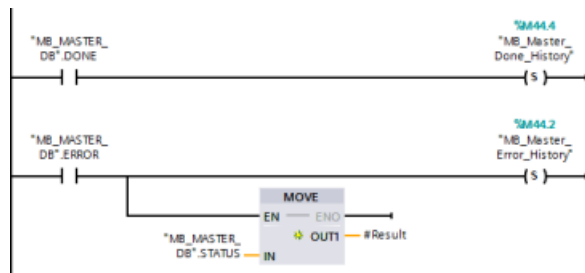
13.9 Modbus RTU-Kommunikation in älteren Systemen (nur CM/CB 1241)



Netzwerk 4: 64 Bit an Daten aus MW600-MW607 in die Ausgangsbitadressen 00017 bis 00081 in Slave Nr. 2 schreiben.



Netzwerk 5: Verlaufsbit DONE setzen, wenn der Schreibvorgang beendet ist. Tritt ein Fehler auf, setzt das Programm das Verlaufsbit ERROR und speichert den STATUS-Code.

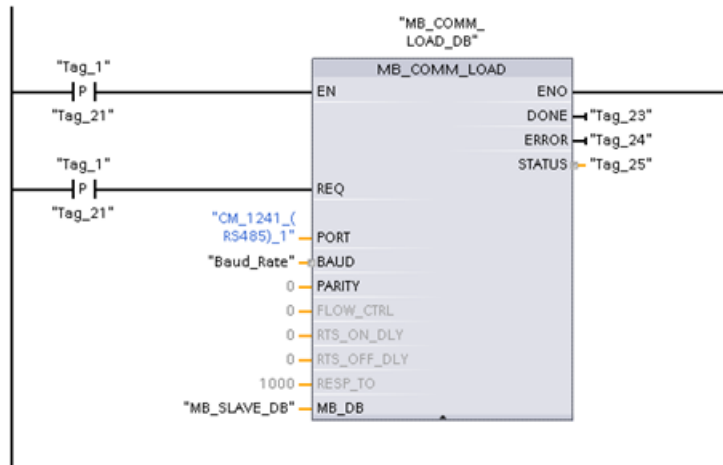


13.9.4.2 Beispiel: Beispielprogramm für einen Modbus RTU-Slave in älteren Systemen

Die im Folgenden gezeigte Anweisung MB_COMM_LOAD wird bei jeder Aktivierung von "Variable_1" initialisiert.

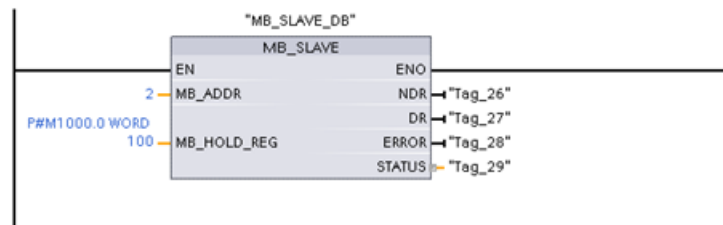
Die Ausführung von MB_COMM_LOAD auf diese Weise darf nur durchgeführt werden, wenn sich die Konfiguration des seriellen Ports als Folge der HMI-Konfiguration während der Laufzeit ändert.

Netzwerk 1: RS485-Modulparameter bei jeder Änderung durch ein HMI-Gerät initialisieren.



Die im Folgenden gezeigte Anweisung MB_SLAVE befindet sich in einem zyklischen OB, der alle 10 ms ausgeführt wird. Dies bietet zwar nicht die absolut schnellste Reaktion durch den Slave, doch es bietet ein gutes Betriebsverhalten bei 9600 Baud für kurze Meldungen (bis zu 20 Byte in der Anforderung).

Netzwerk 2: In jedem Zyklus auf Anforderungen des Modbus-Masters prüfen. Für das Modbus-Haltregister sind 100 Wörter mit Beginn an MW1000 konfiguriert.



13.10 Industrial Remote Communication (IRC)

13.10.1 Übersicht über Telecontrol-CPs

Industrial Remote Communication bietet sicheren und wirtschaftlichen Zugriff auf weit verteilte Maschinen, Anlagen und Anwendungen unterschiedlicher Größen. Industrial Remote Communication umfasst die folgenden Arten der Kommunikation über CP-Module:

- **TeleControl:** Telecontrol ist die Verbindung von Prozessstationen (Remote Terminal Units/ RTUs), die über einen großen geografischen Bereich verteilt sind, mit einem oder mehreren zentralen Prozessleitsystemen zum Zweck des Bedienens und Beobachtens. Mehrere unterschiedliche Übertragungskomponenten in der Produktpalette der Remote-Netzwerke unterstützen die Remote-Kommunikation über einen Bereich an öffentlichen und privaten Netzwerken. Spezielle Telecontrol-Protokolle führen den ereignisgesteuerten oder zyklischen Austausch von Prozessdaten durch, was eine effiziente Steuerung des Gesamtprozesses zulässt.
- **TeleService:** Teleservice umfasst den Datenaustausch mit entfernten technischen Systemen (Maschinen, Anlagen, Computer usw.) zum Zweck der Fehlererkennung, Diagnose, Wartung, Reparatur oder Optimierung.
- **Zusätzliche Anwendungen für die Remote-Kommunikation,** z. B. Überwachung, Smart-Grid-Anwendungen und Maschinenüberwachung.

TeleControl-CPs für die S71200

Für TeleControl-Anwendungen sind die folgenden Kommunikationsprozessoren verfügbar, von denen viele auch Zugriff auf den S7-1200 Webserver (Seite 858) bieten:

- **CP 1243-1:**
 - Artikelnummer: 6GK7 243-1BX30-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 über die öffentliche Infrastruktur (z. B. DSL) an eine Leitstelle mit TeleControl Server Basic (TCSB Version V3).
 - Mit Hilfe von VPN-Technologie und der Firewall ermöglicht der CP geschützten Zugriff auf die S7-1200.
 - Der CP kann als zusätzliche Ethernet-Schnittstelle der CPU für die S7-Kommunikation verwendet werden.
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.
- **CP 1243-1 DNP3:**
 - Artikelnummer: 6GK7 243-1JX30-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 an Leitstellen mittels DNP3-Protokoll
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.

- **CP 1243-1 IEC:**
 - Artikelnummer: 6GK7 243-1PX30-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 an Leitstellen mittels IEC 60870-5-Protokoll
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.
- **CP 1243-1 PCC:**
 - Artikelnummer: 6GK7 243-1HX30-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 an Leitstellen mittels Plant Cloud Communication (PCC)
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.
- **CP 1242-7:**
 - Artikelnummer: 6GK7 242-7KX31-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 über Mobilfunknetze (GPRS) und die öffentliche Infrastruktur (DSL) an eine Leitstelle mit TeleControl Server Basic.
- **CP 1242-7 GPRS V2:**
 - Artikelnummer: 6GK7 242-7KX31-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 über Mobilfunknetze (GPRS) und die öffentliche Infrastruktur (DSL) an eine Leitstelle mit TeleControl Server Basic (TCSB Version V3).
 - Mit Hilfe von VPN-Technologie und der Firewall ermöglicht der CP geschützten Zugriff auf die S7-1200.
 - Der CP kann als zusätzliche Ethernet-Schnittstelle der CPU für die S7-Kommunikation verwendet werden.
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.

- **CP 1243-7 LTE-xx:**
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 über Mobilfunknetze (GPRS) und die öffentliche Infrastruktur (DSL) an eine Leitstelle mit TeleControl Server Basic (TCSB Version V3).
 - Unterstützung der folgenden Mobilfunkspezifikationen: GSM/GPRS, UMTS (G3), LTE
 - Damit der CP auch in Ländern mit unterschiedlichen Mobilfunkspezifikationen eingesetzt werden kann, steht er in zwei Varianten zur Verfügung:
 - CP 1243-7 LTE-US:
 - Nordamerikanischer Standard
 - Artikelnummer: 6GK7 243-7SX30-0XE0
 - CP 1243-7 LTE-EU:
 - Westeuropäischer Standard
 - Artikelnummer: 6GK7 243-7KX30-0XE0
 - Mit Hilfe von VPN-Technologie und der Firewall ermöglicht der CP geschützten Zugriff auf die S7-1200.
 - Der CP kann als zusätzliche Ethernet-Schnittstelle der CPU für die S7-Kommunikation verwendet werden.
 - Die Kommunikation zwischen dem CP und der CPU erfolgt über konfigurierbare Datenpunkte, die auf PLC-Variablen zugreifen.
- **CP 1243-8 IRC:**
 - Artikelnummer: 6GK7 242-8RX30-0XE0
 - Kommunikationsprozessor zum Anschließen der SIMATIC S7-1200 an ein ST7-Netzwerk, eine Datenpunktkonfiguration und ein VPN

Hinweis

Für TeleControl-Anwendungen für andere CPs als den CP 1243-1 benötigen Sie die Software TeleControl Server Basic.

Sichere Kommunikation

Das bewährte SINAUT ST7-Protokoll oder das standardisierte Protokoll DNP3 oder IEC 60870-5 sorgen für Sicherheit in der Industrial Remote Communication (http://w3app.siemens.com/mcms/infocenter/dokumentencenter/sc/ic/InfocenterLanguagePacks/Netzwerksicherheit/6ZB5530-1AP02-0BA4_BR_Network_Security_en_112015.pdf). Die TeleControl-Lösung bietet umfangreiche Maßnahmen, um Datenfälschung und Datenverlust zu verhindern. Jedes Übertragungsmodul hat einen großen Speicher für mehrere tausend Datentelegramme, wodurch Ausfallzeiten in der Übertragungsverbindung überbrückt werden können. Lösungen mit einem dedizierten VPN schützen spezielle IP-basierte Netzwerke.

Der Kommunikationsprozessor CP 1243-1 verbindet die Steuerung SIMATIC S7-1200 sicher mit Ethernet-Netzwerken. Mit seiner integrierten Firewall (Stateful Inspection) und den Sicherheitsfunktionen des VPN-Protokolls (IPsec) unterstützt der Kommunikationsprozessor den Schutz der S7-1200 Stationen und untergeordneten Netzwerke vor unberechtigtem Zugriff und er hilft durch Verschlüsselung, die Datenübertragung vor Manipulation und Spionage zu schützen. Ferner kann der CP auch für die Integration der S7-1200 Station in die

Leitstellensoftware TeleControl Server Basic mittels IP-basierten Remote-Netzwerken verwendet werden.

13.10.2 Anschluss an ein GSM-Netz

IP-basierte WAN-Kommunikation über GPRS

Mit Hilfe des Kommunikationsprozessors CP 1242-7 lässt sich die S7-1200 an GSM-Netze anschließen. Der CP 1242-7 ermöglicht die WAN-Kommunikation von entfernten Stationen mit einer Zentrale und die Kommunikation zwischen Stationen.

Die Kommunikation zwischen Stationen ist nur über das GSM-Netz möglich. Für die Kommunikation einer entfernten Station mit einer zentralen Warte muss in der Zentrale ein PC mit Internet-Anschluss zur Verfügung stehen.

Der CP 1242-7 unterstützt folgende Dienste für die Kommunikation über das GSM-Netz:

- GPRS (General Packet Radio Service)
Der paketorientierte Dienst der Datenübertragung "GPRS" wird über das GSM-Netz abgewickelt.
- Kurzmitteilungsdienst (SMS, Short Message Service)
Der CP 1242-7 kann Meldungen als SMS empfangen und versenden.
Kommunikationspartner kann ein Mobiltelefon oder eine S7-1200 sein.

Der CP 1242-7 ist weltweit für den industrieller Einsatz geeignet und unterstützt folgende Frequenzbänder:

- 850 MHz
- 900 MHz
- 1.800 MHz
- 1.900 MHz

Voraussetzungen

Die Ausrüstung der Stationen oder der Zentrale hängt vom jeweiligen Anwendungsfall ab.

- Für die Kommunikation mit oder über eine zentrale Warte benötigt die Zentrale einen PC mit Internet-Anschluss.
- Für eine entfernte S7-1200-Station mit CP 1242-7, die die Kommunikation über das GSM-Netz nutzen soll, sind neben der Stationsausrüstung folgende Voraussetzungen erforderlich:
 - Ein Vertrag mit einem geeigneten GSM-Netzbetreiber
Wenn GPRS genutzt werden soll, dann muss der Vertrag die Nutzung des GPRS-Dienstes ermöglichen.
Bei direkter Kommunikation zwischen Stationen nur über das GSM-Netz muss der GSM-Netzbetreiber den CPs eine feste IP-Adresse zuweisen. In diesem Fall läuft die Kommunikation zwischen den Stationen nicht über die Zentrale.
 - Die zum Vertrag gehörende SIM-Karte
Die SIM-Karte wird in den CP 1242-7 gesteckt.
 - Lokale Verfügbarkeit eines GSM-Netzes im Bereich der Station

13.10.3 Anwendungen des CP 1242-7

Folgende Anwendungsfälle sind für den CP 1242-7 möglich:

Telecontrol-Anwendungen

- Versenden von Meldungen per SMS
Über den CP 1242-7 empfängt die CPU einer entfernten S7-1200-Station SMS-Nachrichten aus dem GSM-Netz oder verschickt Meldungen per SMS an ein projektiertes Mobiltelefon oder eine S7-1200.
- Kommunikation mit einer Zentrale
Entfernte S7-1200-Stationen kommunizieren über das GSM-Netz und das Internet mit einem Telecontrol-Server in der Zentrale. Für die Datenübertragung per GPRS ist auf dem Telecontrol-Server in der Zentrale die Anwendung "TELECONTROL SERVER BASIC" installiert. Über die integrierte OPC-Server-Funktion kommuniziert der Telecontrol-Server mit einem übergeordneten zentralen Leitsystem.
- Kommunikation zwischen S7-1200-Stationen über ein GSM-Netz
Die Kommunikation zwischen entfernten Stationen und einem CP 1242-7 kann auf zwei unterschiedliche Arten ablaufen:
 - Kommunikation zwischen Stationen über eine Zentrale
In dieser Konfiguration wird eine permanente gesicherte Verbindung zwischen miteinander kommunizierenden S7-1200-Stationen und dem Telecontrol-Server in der Zentrale aufgebaut. Die Kommunikation zwischen den Stationen läuft über den Telecontrol-Server. Der CP 1242-7 arbeitet in Betriebsart "Telecontrol".
 - Direkte Kommunikation zwischen den Stationen
Für die direkte Kommunikation zwischen Stationen ohne den Umweg über eine Zentrale werden SIM-Karten mit fester IP-Adresse eingesetzt, die es ermöglichen, die Stationen direkt anzusprechen. Die möglichen Kommunikationsdienste und Sicherheitsfunktionen (z. B. VPN) hängen dabei vom Angebot des Netzbetreibers ab. Der CP 1242-7 arbeitet in Betriebsart "GPRS direkt".

TeleService über GPRS

Zwischen einer Engineering-Station mit STEP 7 und einer entfernten S7-1200-Station mit CP 1242-7 kann eine TeleService-Verbindung über das GSM-Netz und das Internet aufgebaut werden. Die Verbindung läuft von der Engineering-Station über einen Telecontrol-Server oder über ein TeleService-Gateway, die als Vermittler die Telegramme weiterleiten und die Autorisierung durchführen. Diese PCs nutzen die Funktionen der Anwendung TELECONTROL SERVER BASIC.

Die TeleService-Verbindung können Sie für folgende Zwecke nutzen:

- Laden von Projektierungs- oder Programmdateien aus dem STEP 7-Projekt in die Station
- Abfragen von Diagnosedaten aus der Station

13.10.4 Weitere Eigenschaften des CP 1242-7

Weitere Dienste und Funktionen des CP 1242-7

- Uhrzeitsynchronisation des CP über Internet
Die Uhrzeit des CP können Sie folgendermaßen einstellen:
 - In der Betriebsart "Telecontrol" wird die Uhrzeit vom Telecontrol-Server übertragen. Der CP stellt damit seine Uhrzeit ein.
 - In der Betriebsart "GPRS direkt" kann der CP die Uhrzeit über SNTP anfordern.Zur Synchronisation der CPU-Uhrzeit können Sie die aktuelle Uhrzeit mit Hilfe eines Bausteins aus dem CP auslesen.
- Zwischenspeicherung der zu sendenden Telegramme bei Verbindungsproblemen
- Erhöhte Verfügbarkeit durch die Möglichkeit zum Verbinden mit einem Ersatz-Telecontrol-Server
- Protokollierung des Datenvolumens
Die übertragenen Datenmengen werden protokolliert und können zu weiteren Zwecken ausgewertet werden.

13.10.5 Weitere Informationen

Die CP-Handbücher, zugehörigen Dokumentation und Produktinformationsdokumente bieten ausführliche Informationen:

- CP 1242-7 (<http://support.automation.siemens.com/WW/view/de/45605894>)
- CP 1243-7 LTE (<https://support.industry.siemens.com/cs/ww/de/ps/15924>)
- CP 1243-1 DNP3 (<https://support.industry.siemens.com/cs/ww/de/ps/15938>)
- CP 1243-8 IRC (<https://support.industry.siemens.com/cs/ww/de/ps/21162>)
- CP 1243-1 IEC (<https://support.industry.siemens.com/cs/ww/de/ps/15942>)
- Firmware-Updates nach Verfügbarkeit (<https://support.industry.siemens.com/cs/ww/de/view/109482530>)

13.10.6 Zubehör

GSM/GPRS-Antenne ANT794-4MR

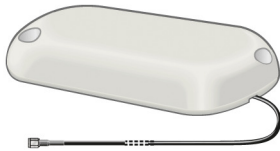
Für den Einsatz in GSM/GPRS-Netzen stehen folgende Antennen zur Montage im Innen- oder Außenbereich zur Verfügung:

- Quadband-Antenne ANT794-4MR (<http://support.automation.siemens.com/WW/view/de/23119005>)



Kurzbezeichnung	Bestell-Nr.	Erläuterung
ANT794-4MR	6NH9 860-1AA00	Quadband-Antenne (900, 1800/1900 MHz, UMTS); witterungsbeständig für Innen- und Außenbereich; 5-m-Anschlusskabel fest mit der Antenne verbunden; SMA-Stecker; inkl. Montagewinkel, Schrauben, Dübel

- Flachantenne ANT794-3M



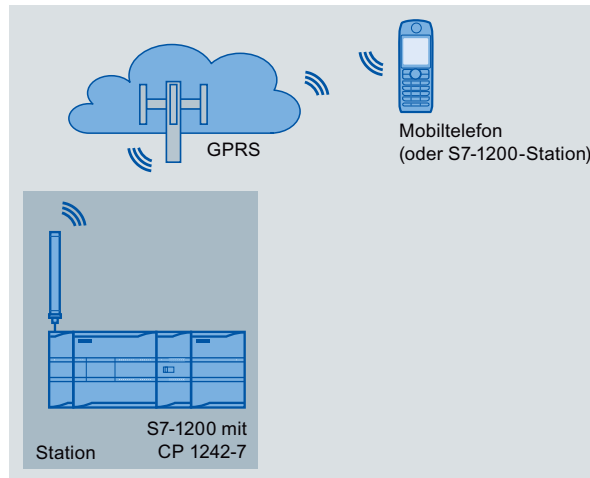
Kurzbezeichnung	Bestell-Nr.	Erläuterung
ANT794-3M	6NH9 870-1AA00	Flachantenne (900, 1800/1900 MHz); witterungsbeständig für Innen- und Außenbereich; 1,2-m-Anschlusskabel fest mit der Antenne verbunden; SMA-Stecker; inkl. Klebepad, Schraubbefestigung möglich

Die Antennen sind separat zu bestellen.

13.10.7 Konfigurationsbeispiele für Telecontrol

Im Folgenden finden Sie einige Konfigurationsbeispiele für Stationen mit CP 1242-7.

Versenden von Meldungen per SMS



Eine SIMATIC S7-1200 mit CP 1242-7 kann Meldungen per SMS an ein Mobiltelefon oder eine projektierte S7-1200-Station versenden.

Telecontrol durch eine Zentrale

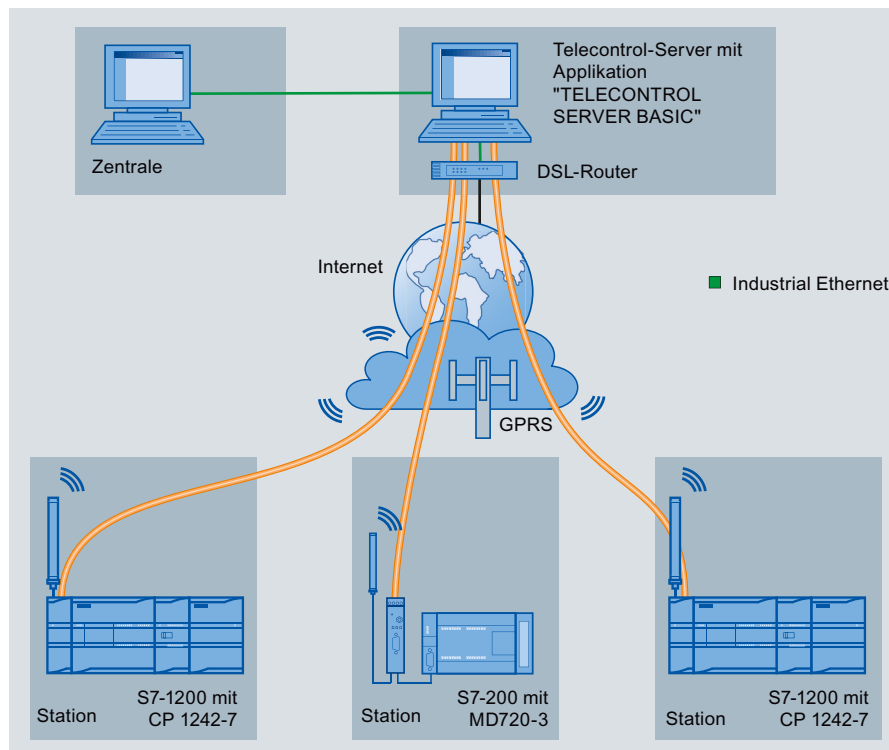


Bild 13-29 Kommunikation zwischen S7-1200-Stationen und einer Zentrale

Bei den Telecontrol-Anwendungen kommunizieren SIMATIC S7-1200-Stationen mit CP 1242-7 über das GSM-Netz und das Internet mit einer Zentrale. Auf dem Telecontrol-Server in der

Zentrale wird die Anwendung "TELECONTROL SERVER BASIC" (TCSB) installiert. Damit ergeben sich folgende Anwendungsfälle:

- Telecontrol-Kommunikation zwischen Station und Zentrale
In diesem Anwendungsfall werden Daten aus dem Feld von den Stationen über das GSM-Netz und das Internet an den Telecontrol-Server in der Zentrale gesendet. Der Telecontrol-Server dient der Überwachung der entfernten Stationen.
- Kommunikation zwischen Station und einer Leitzentrale mit OPC-Client
Wie im ersten Fall kommunizieren die Stationen mit dem Telecontrol-Server. Mithilfe des integrierten OPC-Servers tauscht der Telecontrol-Server die Daten mit dem OPC-Client der Leitzentrale aus.
OPC-Client und Telecontrol-Server können sich auch auf einem einzigen Rechner befinden, beispielsweise wenn TCSB auf einem Leitstellenrechner mit WinCC installiert wird.
- Kommunikation zwischen Stationen über eine Zentrale
Kommunikation ist möglich mit S7-Stationen, die auch mit einem CP 1242-7 ausgerüstet sind.
Für die Kommunikation zwischen Stationen leitet der Telecontrol-Server die Telegramme der Sender-Station an die Empfänger-Station weiter.

Direkte Kommunikation zwischen Stationen

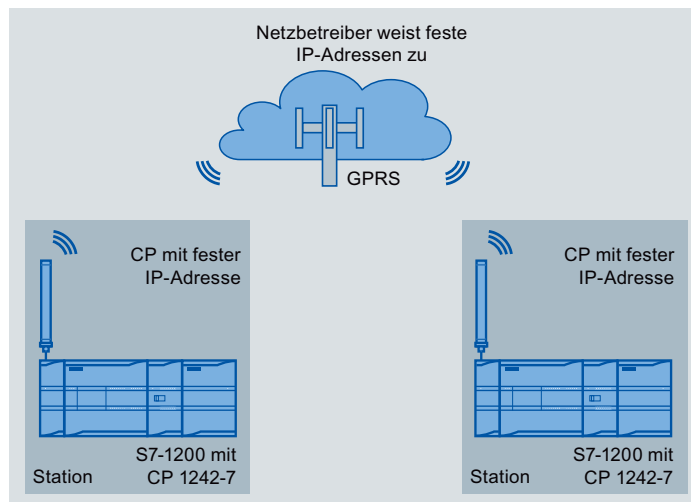


Bild 13-30 Direkte Kommunikation von zwei S7-1200-Stationen

In dieser Konfiguration kommunizieren zwei SIMATIC S7-1200-Stationen mithilfe des CP 1242-7 über das GSM-Netz direkt miteinander. Jeder CP 1242-7 hat eine feste IP-Adresse. Der entsprechende Dienst des GSM-Netzbetreibers muss dies ermöglichen.

TeleService über GPRS

Bei TeleService über GPRS kommuniziert eine Engineering-Station, auf der STEP 7 installiert ist, über das GSM-Netz und das Internet mit dem CP 1242-7 in der S7-1200.

Da eine Firewall in der Regel für Verbindungsanforderungen von außen geschlossen ist, wird eine Vermittlerstation zwischen entfernter Station und Engineering-Station benötigt. Diese

Vermittlerstation kann ein Telecontrol-Server oder, falls in der Konfiguration kein Telecontrol-Server vorhanden ist, ein TeleService-Gateway sein.

TeleService mit Telecontrol-Server

Die Verbindung läuft über den Telecontrol-Server.

- Engineering-Station und Telecontrol-Server sind über Intranet (LAN) oder Internet verbunden.
- Telecontrol-Server und entfernte Station sind über das Internet und das GSM-Netz verbunden.

Engineering-Station und Telecontrol-Server können auch der gleiche Rechner sein, d. h. STEP 7 und TCSB sind auf dem gleichen Rechner installiert.

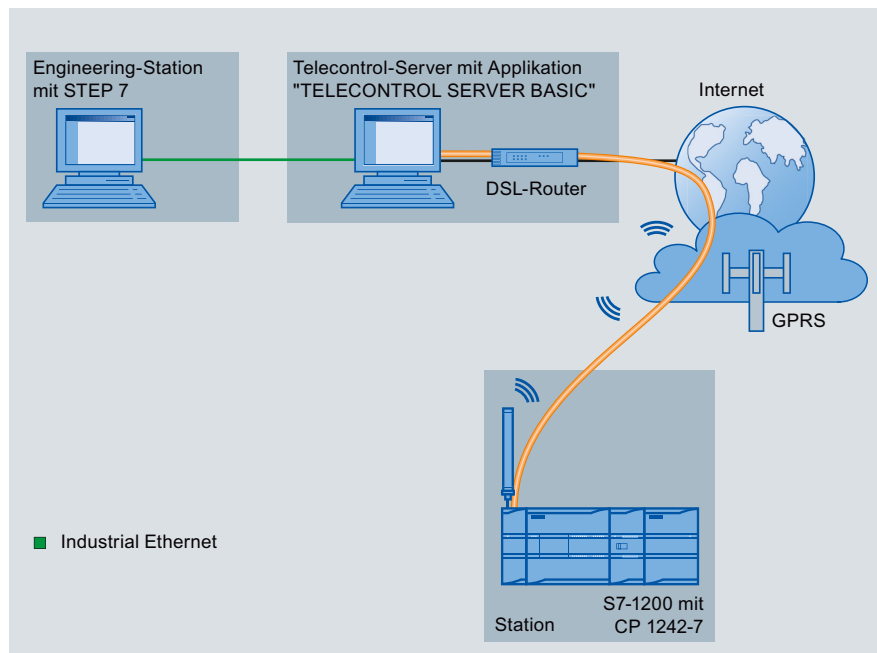


Bild 13-31 TeleService über GPRS in einer Konfiguration mit Telecontrol-Server

TeleService ohne Telecontrol-Server

Die Verbindung läuft über das TeleService-Gateway.

Die Verbindung zwischen Engineering-Station und TeleService-Gateway kann lokal über LAN oder über das Internet laufen.

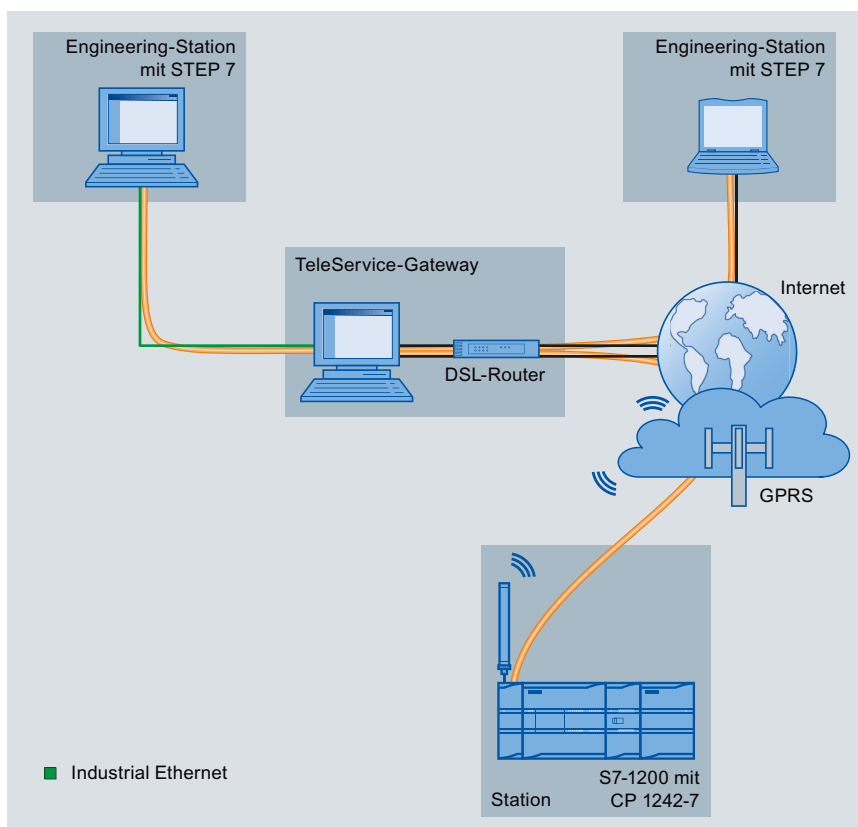


Bild 13-32 TeleService über GPRS in einer Konfiguration mit TeleService-Gateway

TeleService-Kommunikation (SMTP-E-Mail)

14.1 Anweisung TM_Mail (E-Mail senden)

Tabelle 14-1 Anweisung TM_MAIL

KOP/FUP	SCL	Beschreibung																		
<div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">"TM_MAIL_DB"</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center; border-bottom: 1px solid black;">TM_MAIL</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">EN</td> <td style="padding: 2px;">ENO</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">REQ</td> <td style="padding: 2px;">BUSY</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">ID</td> <td style="padding: 2px;">DONE</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">TO_S</td> <td style="padding: 2px;">ERROR</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">CC</td> <td style="padding: 2px;">STATUS</td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">SUBJECT</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">TEXT</td> <td></td> </tr> <tr> <td style="border-right: 1px solid black; padding: 2px;">ATTACHMENT</td> <td></td> </tr> </table> </div>	TM_MAIL		EN	ENO	REQ	BUSY	ID	DONE	TO_S	ERROR	CC	STATUS	SUBJECT		TEXT		ATTACHMENT		<pre>"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_,);</pre>	<p>Die Anweisung TM_MAIL sendet mittels SMTP (Simple Mail Transfer Protocol) eine E-Mail über TCP/IP über die Industrial Ethernet-Verbindung der CPU. Wo keine Ethernet-basierte Internetverbindung verfügbar ist, kann ein optionaler Teleservice-Adapter für die Verbindung mit Telefonfestnetzleitungen verwendet werden. TM_MAIL wird asynchron ausgeführt und der Auftrag erstreckt sich über mehrere Aufrufe von TM_MAIL. Wenn Sie TM_MAIL aufrufen, müssen Sie einen Instanz-DB zuweisen. Der Instanz-DB darf nicht als remanent definiert sein. Dadurch wird sichergestellt, dass der Instanz-DB beim Wechsel der CPU von STOP nach RUN initialisiert wird und dass eine neue Ausführung von TM_MAIL ausgelöst werden kann.</p>
TM_MAIL																				
EN	ENO																			
REQ	BUSY																			
ID	DONE																			
TO_S	ERROR																			
CC	STATUS																			
SUBJECT																				
TEXT																				
ATTACHMENT																				

¹ STEP 7 erstellt automatisch den Instanz-DB, wenn Sie die Anweisung einfügen.

Sie starten den Sendevorgang einer E-Mail mit einer positiven Flanke von 0 nach 1 am Eingangsparameter REQ. Die folgende Tabelle zeigt die Beziehungen zwischen BUSY, DONE und ERROR. Sie können den Fortschritt der Ausführung von TM_MAIL überwachen und die Fertigstellung erkennen, indem Sie diese Parameter bei aufeinander folgenden Aufrufen auswerten.

Die Ausgangsparameter DONE, ERROR, STATUS, und SFC_STATUS gelten nur für jeweils einen Zyklus, wenn der Zustand des Ausgangsparameters BUSY von 1 nach 0 wechselt. Ihre Programmlogik muss temporäre Ausgangszustandswerte speichern, damit Sie in nachfolgenden Programmausführungszyklen Zustandsänderungen erkennen können.

Hinweis

TM_MAIL sendet eine Mail über TCP/IP unter Verwendung der Ethernet-Schnittstelle der CPU. Um eine Mail über eine CP-Schnittstelle (mit oder ohne SSL) zu senden, verwenden Sie die Anweisung Anweisung TMAIL_C (Email über die Ethernet-Schnittstelle der CPU senden) (Seite 688).

14.1 Anweisung TM_Mail (E-Mail senden)

Tabelle 14-2 Interaktion zwischen den Parametern Done, Busy und Error

DONE	BUSY	ERROR	Beschreibung
Irrelevant	1	Irrelevant	Auftrag ist in Bearbeitung.
1	0	0	Der Auftrag wurde erfolgreich ausgeführt.
0	0	1	Auftrag wurde mit Fehler beendet. Die Ursache des Fehlers finden Sie im Parameter STATUS.
0	0	0	Kein Auftrag in Bearbeitung.

Wenn die CPU während der Ausführung von TM_MAIL in STOP versetzt wird, wird die Kommunikationsverbindung zum E-Mailserver beendet. Die Kommunikationsverbindung zum E-Mailserver geht auch verloren, wenn bei der Kommunikation der CPU auf dem Industrial Ethernet-Bus Probleme auftreten. In diesen Fällen wird der Sendevorgang unterbrochen und die E-Mail erreicht den Empfänger nicht.

ACHTUNG

Anwenderprogramme ändern

Das Löschen und Ersetzen von Programmbausteinen, die Aufrufe von TM_MAIL oder Aufrufe des Instanz-DBs von TM_MAIL können die Verknüpfung von Programmbausteinen unterbrechen. Wenn Sie verknüpfte Programmbausteine nicht speichern, können die TPC/IP-Kommunikationsfunktionen einen undefinierten Zustand einnehmen, der möglicherweise zu Sachschaden führt. Nach der Übertragung eines geänderten Programmbausteins müssen Sie einen CPU-Neustart (Warmstart) oder einen Kaltstart durchführen.

Um die Verknüpfung von Programmbausteinen nicht zu unterbrechen, ändern Sie die Teile Ihres Anwenderprogramms, die sich direkt auf die TM_MAIL-Aufrufe auswirken, nur in den folgenden Fällen:

- Sich die CPU in STOP befindet
- Keine E-Mail gesendet wird (REQ und BUSY = 0)

Datenkonsistenz

Der Eingangsparameter ADDR_MAIL_SERVER wird gelesen, wenn der Vorgang gestartet wird. Ein neuer Wert wird erst wirksam, wenn der gegenwärtige Vorgang beendet ist und eine neue Anweisung TM_MAIL initiiert wird.

Im Gegensatz dazu werden die Parameter WATCH_DOG_TIME, TO_S, CC, FROM, SUBJECT, TEXT, ATTACHMENT, USERNAME und PASSWORD während der Ausführung von TM_MAIL gelesen und können erst geändert werden, wenn der Auftrag beendet ist (BUSY = 0)

Wählverbindung: IE-Parameter des TS-Adapters konfigurieren

Sie müssen die IE-Parameter des Teleservice-Adapters für ausgehende Aufrufe konfigurieren, um eine Wählverbindung zum Server Ihres Internet-Diensteanbieters herzustellen. Wenn Sie das Attribut für den Aufruf auf Anforderung einrichten, wird die Verbindung nur hergestellt, wenn eine E-Mail gesendet wird. Bei einer analogen Modemverbindung ist für den Verbindungsvorgang mehr Zeit erforderlich (ca. eine zusätzliche Minute). Sie müssen diese zusätzliche Zeit in den Wert WATCH_DOG_TIME einschließen.

Tabelle 14-3 Datentypen für die Parameter

Parameter und Datentyp	Datentypen	Beschreibung	
REQ	IN	Bool	Die Anweisung wird durch eine steigende Flanke (0 nach 1) gestartet.
ID	IN	Int	Verbindungskenung: Siehe ID-Parameter der Anweisungen TCON, TDISCON, TSEND und TRCV. Eine Zahl, die nicht für weitere Instanzen dieser Anweisung im Anwenderprogramm verwendet wird, muss verwendet werden.
TO_S	IN	String	Empfängeradressen: STRING-Daten mit einer maximalen Länge von 240 Zeichen.
CC	IN	String	Empfängeradressen für CC-Kopie (optional): STRING-Daten mit einer maximalen Länge von 240 Zeichen.
SUBJECT	IN	String	Betreff der E-Mail: STRING-Daten mit einer maximalen Länge von 240 Zeichen.
TEXT	IN	String	Text der E-Mail (optional): STRING-Daten mit einer maximalen Länge von 240 Zeichen. Enthält dieser Parameter eine leere Zeichenkette, wird die E-Mail ohne Textnachricht gesendet.
ATTACHMENT	IN	Variant	Pointer auf E-Mail-Anhang: Byte-, Wort- oder Doppelwortdaten mit einer maximalen Länge von 65.534 Byte. Wird kein Wert zugewiesen, wird die E-Mail ohne Anhang gesendet.
DONE	OUT	Bool	<ul style="list-style-type: none"> 0 - Auftrag noch nicht gestartet oder wird noch ausgeführt. 1 - Auftrag wurde fehlerfrei ausgeführt.
BUSY	OUT	Bool	<ul style="list-style-type: none"> 0 - Kein Vorgang in Bearbeitung 1 - Vorgang in Bearbeitung
ERROR	OUT	Bool	Das Bit ERROR ist einen Zyklus lang = 1, nachdem die letzte Anforderung mit Fehler beendet wurde. Der Fehlercode im Ausgang STATUS ist nur in dem einen Zyklus gültig, in dem ERROR = 1 ist.
STATUS	OUT	Word	Rückgabewert oder Fehlerinformationen der Anweisung TM_MAIL.

Parameter und Datentyp		Datentypen	Beschreibung
ADDR_MAIL_SERVER	¹ Static	DWord	IP-Adresse des Mailserver: Sie müssen jedes Fragment der IP-Adresse als Oktett aus zwei 4-Bit-Hexadezimalzeichen zuweisen. Ist das Fragment der IP-Adresse = Dezimalwert 10, was dem Hexadezimalwert A entspricht, müssen Sie für dieses Oktett "0A" eingeben. Beispiel: IP-Adresse = 192.168.0.10 ADDR_MAIL_SERVER = DW#16#C0A8000A, wobei gilt: <ul style="list-style-type: none"> • 192 = 16#C0, • 168 = 16#A8 • 0 = 16#00 • 10 = 16#0A
WATCH_DOG_TIME	¹ Static	Time	Die maximale Zeitdauer, während der TM_MAIL den gesamten SMTP-Prozess abschließen muss, von der Einleitung der Verbindung zum SMTP bis zum Ende der SMTP-Übertragung. Wird dieser Zeitraum überschritten, wird die Ausführung von TM_MAIL mit einem Fehler beendet. Die tatsächliche Zeitverzögerung, bis TM_MAIL endet und der Fehler ausgegeben wird, kann die WATCH_DOG_TIME überschreiten, weil zum Trennen der Verbindung zusätzliche Zeit erforderlich ist. Sie sollten zunächst einen Zeitraum von 2 Minuten einrichten. Dieser Zeitraum kann bei einer ISDN-Telefonverbindung sehr viel kleiner sein.
USERNAME	¹ Static	String	Benutzername des Mailkontos: STRING-Daten mit einer maximalen Länge von 180 Zeichen.
PASSWORD	¹ Static	String	Passwort für den Mailserver: STRING-Daten mit einer maximalen Länge von 180 Zeichen.
FROM	¹ Static	String	Adresse des Absenders: STRING mit einer maximalen Länge von 240 Zeichen
SFC_STATUS	¹ Static	Word	Bedingungscode der Ausführung der aufgerufenen Kommunikationsbausteine

¹ Die Werte dieser Parameter werden nicht bei jedem Aufruf von TM_MAIL modifiziert. Die Werte werden im Instanz-Datenbaustein TM_MAIL zugewiesen und nur einmal referenziert, im ersten Aufruf von TM_MAIL.

SMTP-Authentifizierung

TM_MAIL unterstützt das SMTP-AUTH-LOGIN-Authentifizierungsverfahren. Informationen zu diesem Authentifizierungsverfahren finden Sie im Handbuch des Mailserver oder auf der Website Ihres Internet-Diensteanbieters.

Das Authentifizierungsverfahren AUTH LOGIN verwendet die Parameter USERNAME und PASSWORD der Anweisung TM_MAIL, um eine Verbindung zum Mailserver herzustellen. Benutzername und Passwort müssen zuvor in einem E-Mail-Konto auf einem E-Mailserver eingerichtet sein.

Wenn dem Parameter USERNAME kein Wert zugewiesen ist, wird das AUTH LOGIN-Authentifizierungsverfahren nicht verwendet und die E-Mail wird ohne Authentifizierung gesendet.

TO_S:, CC: und FROM: Parameter

Die Parameter TO_S:, CC: und FROM: sind Zeichenketten, wie in den folgenden Beispielen gezeigt:

TO: <wenna@mydomain.com>, <ruby@mydomain.com>,

CC: <admin@mydomain.com>, <judy@mydomain.com>,

FROM: <admin@mydomain.com>

Die folgenden Regeln müssen bei der Eingabe dieser Zeichenfolgen beachtet werden:

- Die Zeichen "TO:", "CC:" und "FROM:" müssen eingegeben werden, einschließlich des Doppelpunkts.
- Ein Leerzeichen und eine öffnende spitze Klammer "<" müssen vor jeder Adresse stehen. Beispielsweise muss sich zwischen "TO:" und <E-Mail-Adresse> ein Leerzeichen befinden.
- Eine abschließende spitze Klammer ">" muss nach jeder Adresse eingegeben werden.
- Bei den Adressen TO_S: und CC: muss nach jeder Adresse ein Komma "," eingegeben werden. Beispielsweise ist das Komma nach der einzelnen E-Mail-Adresse in "TO: <email address>," erforderlich.
- Für die Eingabe FROM: darf nur eine E-Mail-Adresse angegeben werden, ohne Komma am Ende.

Wegen des Laufzeitmodus und der Speicherauslastung wird für die Daten TO_S:, CC: und FROM: der Anweisung TM_MAIL keine Syntaxprüfung durchgeführt. Wenn die obigen Formatregeln nicht exakt eingehalten werden, schlägt die Übertragung des SMTP-E-Mailserver fehl.

Parameter STATUS und SFC_STATUS

Die von TM_MAIL zurückgegebenen BedingungsCodes der Ausführung können wie folgt klassifiziert werden:

- W#16#0000: Ausführung von TM_MAIL wurde erfolgreich beendet.
- W#16#7xxx: Zustand der Ausführung von TM_MAIL.
- W#16#8xxx: Fehler in einem internen Aufruf eines Kommunikationsgeräts oder des Mailserver.

Die folgende Tabelle zeigt die BedingungsCodes der Ausführung von TM_MAIL, mit Ausnahme der BedingungsCodes von intern aufgerufenen Kommunikationsmodulen.

Hinweis

Anforderungen an den E-Mailserver

TM_MAIL kann nur mittels SMTP über Port 25 mit einem E-Mailserver kommunizieren. Die zugewiesene Portnummer kann nicht geändert werden.

Die meisten IT-Abteilungen und externen E-Mailserver blockieren mittlerweile Port 25, um zu verhindern, dass ein mit einem Virus infizierter PC zu einem gefährlichen E-Mail-Generator wird.

Sie können über SMTP eine Verbindung mit einem internen E-Mailserver herstellen und den internen Server die aktuellen Sicherheitserweiterungen verwalten lassen, die erforderlich sind, um E-Mails über das Internet an einen externen E-Mailserver weiterzuleiten.

Beispiel: Konfiguration eines internen E-Mailserver

Wenn Sie Microsoft Exchange als internen E-Mailserver nutzen, können Sie den Server so konfigurieren, dass der SMTP-Zugriff von der IP-Adresse, die dem S7-1200 Zielsystem zugewiesen ist, zulässig ist. Exchange-Verwaltungskonsolle konfigurieren: Serverkonfiguration > Hub-Transport > Empfangsconnectors > IP-Relay. Im Register "Netzwerke" gibt es das Feld "E-Mail von Remoteservern mit folgenden IP-Adressen empfangen". Hier geben Sie die IP-Adresse des PLC-Geräts ein, das die Anweisung TM_MAIL ausführt. Für diesen Verbindungstyp ist bei einem internen Microsoft Exchange-Server keine Authentifizierung erforderlich.

Konfiguration des E-Mailserver

TM_MAIL kann nur einen E-Mailserver verwenden, der die Kommunikation über Port 25, SMTP und die AUTH-LOGIN-Authentifizierung (optional) gestattet.

Richten Sie ein kompatibles Konto auf dem E-Mailserver ein, um die Remote-SMTP-Anmeldung zu akzeptieren. Dann bearbeiten Sie den Instanz-DB für TM_MAIL und geben die Zeichenfolgen TM_MAILUSERNAME und PASSWORD ein, über die die Verbindung mit Ihrem E-Mail-Konto authentifiziert wird.

Tabelle 14-4 Bedingungscode

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Beschreibung
0000	-	Die Ausführung von TM_MAIL wurde fehlerfrei beendet. Dieser STATUS-Code 0 gewährleistet nicht, dass tatsächlich eine E-Mail gesendet wurde (siehe Punkt 1 im auf diese Tabelle folgenden Hinweis).
7001	-	TM_MAIL ist aktiv (BUSY = 1).
7002	7002	TM_MAIL ist aktiv (BUSY = 1).
8xxx	xxxx	Die Ausführung von TM_MAIL wurde mit einem Fehler in den internen Aufrufen der Kommunikationsanweisung beendet. Weitere Informationen zum Parameter SFC_STATUS finden Sie in den Beschreibungen des Parameters STATUS der zugrunde liegenden offenen PROFINET-Benutzerkommunikationsanweisungen.
8010	xxxx	Verbindung konnte nicht aufgebaut werden: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TCON.

STATUS (W#16#...):	SFC_STATUS (W#16#...):	Beschreibung
8011	xxxx	Fehler beim Senden der Daten: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TSEND.
8012	xxxx	Fehler beim Empfangen der Daten: Weitere Informationen zum Parameter SFC_STATUS finden Sie in der Beschreibung des Parameters STATUS der Anweisung TRCV.
8013	xxxx	Verbindung konnte nicht aufgebaut werden: Weitere Informationen zum Auswerten des Parameters SFC_STATUS finden Sie in den Beschreibungen des Parameters STATUS der Anweisungen TCON und TDISCON.
8014	-	Verbindung konnte nicht aufgebaut werden: Sie haben möglicherweise eine inkorrekte IP-Adresse des Mailservers (ADDR_MAIL_SERVER) oder zu wenig Zeit (WATCH_DOG_TIME) für die Verbindung angegeben. Es ist auch möglich, dass die CPU keine Verbindung zum Netzwerk hat oder die CPU-Konfiguration inkorrekt ist.
8015	-	Ungültiger Pointer für den Parameter ATTACHMENT: Verwenden Sie einen Variant-Pointer mit Datentyp- und Längenzuweisung. Beispiel: "P#DB.DBX0.0" ist falsch und "P#DB.DBX0.0 Byte 256" ist richtig.
82xx, 84xx, 85xx	-	Die Fehlermeldung kommt vom Mailserver und entspricht der Fehlernummer "8" des SMTP-Protokolls. Siehe Punkt 2 in dem auf diese Tabelle folgenden Hinweis.
8450	-	Vorgang wird nicht ausgeführt: Mailbox ist nicht verfügbar, wiederholen Sie den Vorgang später.
8451	-	Vorgang abgebrochen: Lokaler Fehler bei der Bearbeitung. Wiederholen Sie den Vorgang später.
8500	-	Fehler in der Befehlsyntax: Die Ursache kann sein, dass der E-Mailserver den LOGIN-Authentifizierungsvorgang nicht unterstützt. Prüfen Sie die Parameter von TM_MAIL. Versuchen Sie, eine E-Mail ohne Authentifizierung zu senden. Versuchen Sie, den Parameter USERNAME durch eine leere Zeichenkette zu ersetzen.
8501	-	Syntaxfehler: Inkorrekter Parameter oder inkorrektes Argument; Sie haben möglicherweise eine inkorrekte Adresse in den Parameter TO_S oder CC eingegeben.
8502	-	Befehl ist unbekannt oder nicht implementiert: Prüfen Sie Ihre Eingaben, insbesondere den Parameter FROM. Möglicherweise ist dieser unvollständig und Sie haben eines der Zeichen "@" oder "." weggelassen.
8535	-	SMTP-Authentifizierung ist unvollständig. Sie haben möglicherweise einen inkorrekten Benutzernamen oder ein inkorrektes Passwort eingegeben.
8550	-	Der Mailserver ist nicht erreichbar, oder Sie haben keine Zugriffsrechte. Sie haben möglicherweise einen inkorrekten Benutzernamen oder ein inkorrektes Passwort eingegeben oder Ihr Mailserver unterstützt keinen Zugriff über Anmeldung. Eine weitere Ursache für diesen Fehler kann eine fehlerhafte Eingabe des Domainnamens nach dem Zeichen "@" in den Parametern TO_S oder CC sein.
8552	-	Vorgang abgebrochen: Zugeordnete Speichergröße überschritten; wiederholen Sie den Vorgang später.
8554	-	Übertragung fehlgeschlagen: Wiederholen Sie den Vorgang später.

Hinweis

Mögliche nicht gemeldete E-Mail-Übertragungsfehler

- Die fehlerhafte Angabe einer Empfängeradresse erzeugt bei TM_MAIL keinen STATUS-Fehler. In einem solchen Fall ist nicht gewährleistet, dass die zusätzlichen Empfänger (mit korrekten E-Mail-Adressen) die E-Mail empfangen.
 - Weitere Informationen zu SMTP-Fehlercodes finden Sie im Internet oder in der Fehlerdokumentation für den Mailserver. Sie können auch die letzte vom Mailserver erzeugte Fehlermeldung auslesen. Die Fehlermeldung wird im Parameter buffer1 des Instanz-DB von TM_MAIL gespeichert.
-

Online- und Diagnose-Tools

15.1 Status-LEDs

Die CPU und die E/A-Module nutzen LEDs, um Informationen über den Betriebszustand des Moduls oder der E/A zu liefern.

Status-LEDs an einer CPU

Die CPU bietet die folgenden Statusanzeigen:

- STOP/RUN
 - Gelbes Dauerlicht zeigt den Betriebszustand STOP an
 - Grünes Dauerlicht zeigt den Betriebszustand RUN an
 - Blinken (abwechselnd grün und gelb) zeigt an, dass die CPU in der Betriebsart STARTUP ist
- ERROR
 - Eine blinkende LED zeigt einen Fehler an, z. B. einen internen CPU-Fehler, einen Fehler der Memory Card oder einen Konfigurationsfehler (unpassende Module).
 - Rotes Blinklicht für drei Sekunden zeigt einen Fehler an, der nicht mehr ansteht. Ein Beispiel ist das Rücksetzen der Echtzeituhr (RTC) auf die Standardeinstellung bei einem Spannungsausfall.
 - Fehlerzustand:
 - Rotes Dauerlicht zeigt defekte Hardware an
 - Alle LEDs blinken, wenn die Firmware einen Fehler erkennt
- Wenn Sie eine Memory Card stecken, blinkt die LED MAINT (Wartung). Schalten Sie die CPU aus und wieder ein. Die CPU wechselt dann in den Betriebszustand STOP. Nachdem die CPU in den Betriebszustand STOP gegangen ist, führen Sie eine der folgenden Funktionen durch, um die Auswertung der Memory Card zu starten:
 - Versetzen Sie die CPU in den Betriebszustand RUN
 - Führen Sie ein Umlöschen durch (MRES)
 - Schalten Sie die CPU aus und wieder ein

Den Zustand der LEDs können Sie auch mit der Anweisung LED (Seite 430) ermitteln.

Tabelle 15-1 Status-LEDs für eine CPU

Beschreibung	STOP/RUN Gelb / Grün	ERROR Rot	MAINT Gelb
Netz aus	Aus	Aus	Aus
Anlauf, Selbsttest oder Firmware-Aktualisierung	Blinken (abwechselnd gelb und grün)	-	Aus
Betriebszustand STOP	Ein (gelb)	-	-
Betriebszustand RUN	Ein (grün)	-	-

Beschreibung	STOP/RUN Gelb / Grün	ERROR Rot	MAINT Gelb
Ziehen Sie die Memory Card	Ein (gelb)	-	Blinkt
Fehler	Ein (gelb oder grün)	Blinkt	-
Wartung erforderlich <ul style="list-style-type: none"> • Geforderte E/A • Batteriewechsel erforderlich (bei installiertem Batterieboard) 	Ein (gelb oder grün)	-	Ein
Hardware defekt	Ein (gelb)	Ein	Aus
LED-Test oder CPU-Firmware defekt	Blinken (abwechselnd gelb und grün)	Blinkt	Blinkt
Unbekannte oder inkompatible Version der CPU-Konfiguration	Ein (gelb)	Blinkt	Blinkt

Die CPU bietet auch zwei LEDs, die den Zustand der PROFINET-Kommunikation anzeigen. Öffnen Sie die untere Abdeckklappe der Klemmenleiste, um die PROFINET-LEDs zu sehen.

- Link (grün) wird eingeschaltet, um eine erfolgreiche Verbindung anzuzeigen
- Rx/Tx (gelb) wird eingeschaltet, um Übertragungsaktivität anzuzeigen

Die CPU und jedes digitale Signalmodul (SM) bieten eine I/O Channel -LED für jeden digitalen Eingang und Ausgang. I/O Channel (grün) wird ein- oder ausgeschaltet, um den Zustand des jeweiligen Eingangs oder Ausganges anzuzeigen.

Fehler "Unbekannte oder inkompatible Version der CPU-Konfiguration"

Der Diagnosepuffer kann den Fehler "Unbekannte oder inkompatible Version der CPU-Konfiguration" melden, was in folgenden Situationen vorkommen kann:

- Wenn Sie versuchen, ein S7-1200 V3.0-Programm in eine S7-1200 V4.x zu laden
- Wenn Sie versuchen, ein Projekt zu laden, bei dem der Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161) zwischen CPU und dem Projekt unterschiedlich ist

Wenn Sie diesen Zustand durch Verwendung einer Übertragungskarte (Seite 122) mit einer ungültigen Programmversion hervorgerufen haben, gehen Sie wie folgt vor, um den Fehler zu beheben:

1. Entnehmen Sie die Übertragungskarte.
2. Führen Sie einen Wechsel von STOP nach RUN aus.
3. Führen Sie ein Umröchen (MRES) durch oder schalten Sie das Gerät aus und wieder ein.

Wenn Sie diesen Zustand durch Laden eines ungültigen Programms hervorgerufen haben, setzen Sie die CPU auf die Werkseinstellungen zurück (Seite 1214).

Wenn Sie diesen Zustand durch Nichtübereinstimmung des Schutzes vertraulicher PLC-Konfigurationsdaten in CPU und Projekt hervorgerufen haben, verwenden Sie die Online- und Diagnosetools (Seite 1213), um für das Passwort der Online-CPU zum Schutz vertraulicher PLC-Konfigurationsdaten das Passwort des Projekts festzulegen, oder löschen Sie das Passwort aus der Online-CPU.

Nachdem Sie den Fehlerzustand der CPU behoben haben, können Sie ein gültiges Programm laden.

Verhalten der S7-1200 nach einem schweren Fehler

Wenn die CPU-Firmware einen schweren Fehler erkennt, wird ein Neustart im Fehlermodus versucht und der Fehlermodus durch beständiges Blinken der LEDs STOP/RUN, ERROR und MAINT angezeigt. Das Anwenderprogramm und die Hardwarekonfiguration werden nach dem Neustart im Fehlermodus nicht geladen.

Wenn die CPU den Neustart im Fehlermodus erfolgreich durchführt, führt die CPU folgende Aktionen durch:

- Einstellung der CPU und Signalboard-Ausgänge auf 0
- Einstellung der Ausgänge der Signalmodule im zentralen Baugruppenträger und der dezentralen Peripherie auf die Auswahl für "Reaktion auf CPU STOP" in der Gerätekonfiguration der digitalen Ausgänge des Moduls

Schlägt der Neustart im Fehlermodus fehl (z. B. aufgrund eines Hardwarefehlers), sind die LEDs STOP und ERROR eingeschaltet und die LED MAINT ist ausgeschaltet.



WARNUNG

Betrieb im Fehlerzustand kann nicht gewährleistet werden

Steuerungen können bei unsicheren Betriebszuständen ausfallen und dadurch unkontrolliertes Betriebsverhalten der gesteuerten Geräte verursachen. Derartiges unerwartetes Betriebsverhalten des Automatisierungssystems kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Sorgen Sie daher für eine NOTAUSFunktion, elektromechanische oder andere redundante Sicherheitseinrichtungen, die von Ihrem PLC unabhängig sind.

Status-LEDs an einem Signalmodul (SM)

Außerdem bietet jedes digitale SM eine DIAG-LED, die den Zustand des Moduls anzeigt:

- Grün zeigt an, dass das Modul betriebsbereit ist
- Rot zeigt an, dass das Modul defekt oder nicht betriebsbereit ist

Jedes analoge SM bietet eine I/O Channel -LED für jeden der analogen Eingänge und Ausgänge.

- Grün zeigt an, dass der Kanal konfiguriert wurde und aktiv ist
- Rot zeigt einen Fehlerzustand des jeweiligen analogen Eingangs oder Ausgangs an

Außerdem bietet jedes analoge SM eine DIAG-LED, die den Zustand des Moduls anzeigt:

- Grün zeigt an, dass das Modul betriebsbereit ist
- Rot zeigt an, dass das Modul defekt oder nicht betriebsbereit ist

15.1 Status-LEDs

Das SM erkennt das Vorhandensein bzw. die Abwesenheit von Modulspannung (feldseitige Spannung, sofern erforderlich).

Tabelle 15-2 Status-LEDs für ein Signalmodul (SM)

Beschreibung	DIAG (Gelb/Grün)	I/O Channel (Gelb/Grün)
Feldseitige Spannung ist aus*	Rot blinkend	Rot blinkend
Nicht konfiguriert oder Aktualisierung in Bearbeitung	Grün blinkend	Aus
Modul fehlerfrei konfiguriert	Ein (grün)	Ein (grün)
Fehlerbedingung	Rot blinkend	-
E/A-Fehler (bei aktivierter Diagnose)	-	Rot blinkend
E/A-Fehler (bei deaktivierter Diagnose)	-	Ein (grün)

* Der Status wird nur bei den Analogsignalmodulen unterstützt.

Analogmoduldiagnose

Analogmodule haben abhängig von Modul- und Kanaltyp mehrere Diagnosen. Sie können diese Diagnosen im TIA Portal in der Gerätekonfiguration des Projekts bzw. in den allgemeinen Eigenschaften des Moduls für jedes Modul und jeden Kanal einzeln aktivieren bzw. deaktivieren.

Modulfehler

Stromversorgungsfehler werden wie folgt gemeldet:

Stromversorgungsfehler	Gemeldeter Fehler
Analogmodul mit einem Stromversorgungsdiagnosefehler meldet:	Überlauf: 32767 für alle Eingangskanäle Fehlende Spannungsversorgungsdiagnose, wenn für Ausgabemodule aktiviert

Kanaltypfehler

Sie können diese Diagnose für jeden Kanal und jeden Kanaltyp einzeln aktivieren (siehe nachfolgende Tabelle).

Kanaltyp	Gemeldeter Fehler
Spannungseingang	Überlauf: 32767
	Unterlauf: -32768
Stromeingang (0 bis 20 mA)	Überlauf: 32767
	Unterlauf: -32768
Stromeingang (4 bis 20 mA) (bei Eingang < 1,185 mA)	Drahtbruch: 32767
	Überlauf: 32767
Spannungsausgang (bei Ausgang > 0,5 V)	Kurzschlussdiagnose, wenn aktiviert
Stromausgang (bei Ausgang > 1,0 mA)	Stromkreisunterbrechungsdiagnose, wenn aktiviert
RTD-Eingang	Drahtbruch: 32767
	Überlauf: 32767
	Unterlauf: -32768

Kanaltyp	Gemeldeter Fehler
Widerstandseingang	Drahtbruch: 32767
	Überlauf: 32767
Thermoelementeingang	Drahtbruch: 32767
	Überlauf: 32767
	Unterlauf: -32768

Ein Analogeingabemodul mit einem Diagnosefehler an einem beliebigen Kanal meldet 32767 oder -32768 an dem Kanal, selbst wenn die Diagnose nicht aktiviert ist. Analogeingangskanäle melden 32767, wenn deaktiviert.

Analogeingabemodule können Diagnosefehler an mehreren Kanälen gleichzeitig haben (mehrere Fehler). Wenn das geschieht, wird nur der erste Fehler der CPU gemeldet. Nachdem der erste Fehler gemeldet wurde, werden keine weiteren Fehler gemeldet, bis die Ursache des ersten Fehlers im Modul behoben ist. Nachdem der erste Fehler behoben ist, wird der zweite Fehler gemeldet, sofern der Fehlerzustand noch vorhanden ist.

Status-LEDs an einem Signalboard (SB)

Jedes analoge SB bietet für jeden der Analogeingänge und -ausgänge eine I/O Channel -LED.

Tabelle 15-3 Status-LEDs für ein Signalboard (SB)

Beschreibung	I/O Channel (Rot/Grün)
Nicht konfiguriert oder Aktualisierung in Bearbeitung	Aus
Board fehlerfrei konfiguriert	Ein (grün)
E/A-Fehler (bei aktivierter Diagnose)	Rot blinkend
E/A-Fehler (bei deaktivierter Diagnose)	Ein (grün)

15.2 Online-Verbindung mit einer CPU herstellen

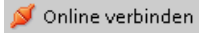
Zum Laden von Programmen und Projektdaten sowie für die folgenden Tätigkeiten ist eine Online-Verbindung zwischen dem Programmiergerät und der CPU erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU (Seite 1218)
- Anzeigen und Einstellen von Datum und Uhrzeit der CPU (Seite 1212)
- Anzeigen der Modulinformationen
- Vergleichen und Synchronisieren (Seite 1221) von Offline- und Online-Programmbausteinen
- Laden von Programmbausteinen in die und aus der CPU
- Anzeigen von Diagnose und Diagnosepuffer (Seite 1219)
- Mit einer Beobachtungstabelle (Seite 1226) das Anwenderprogramm durch Beobachten und Steuern von Werten testen

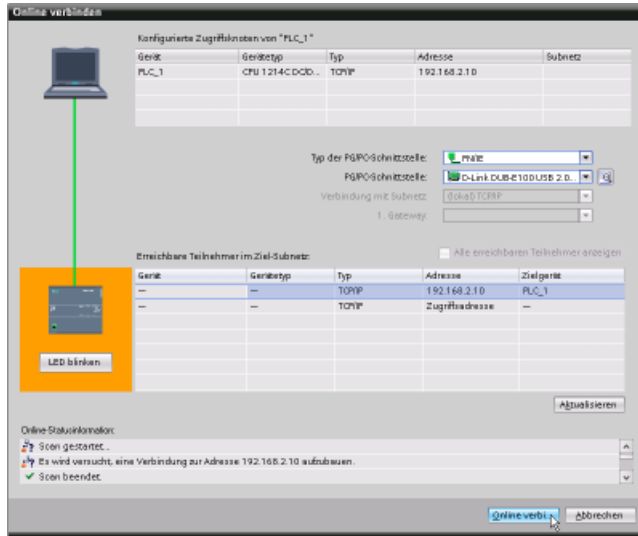
15.3 Einem PROFINET IO-Gerät online einen Namen zuweisen

- Mit einer Forcetable Werte in der CPU forcen (Seite 1229)
- CPU auf Werkseinstellungen zurücksetzen (Seite 1214)

Um eine Online-Verbindung zu einer konfigurierten CPU herzustellen, klicken Sie in der Projektnavigation auf die CPU und in der Projektansicht auf die Schaltfläche "Online verbinden":



Wenn Sie zum ersten Mal mit dieser CPU online gehen, müssen Sie den Typ der PG/PC-Schnittstelle und die spezifische PG/PC-Schnittstelle im Dialog "Online verbinden" auswählen, bevor Sie eine Online-Verbindung zu einer CPU an dieser Schnittstelle herstellen.



Wenn die CPU über Schutz vertraulicher PLC-Konfigurationsdaten verfügt, werden Sie möglicherweise aufgefordert, der CPU zu vertrauen. Sie können das Zertifikat der CPU anzeigen und verifizieren und entscheiden, ob Sie der CPU-Onlineverbindung vertrauen oder die Verbindung abbrechen möchten.

Nach der Verbindung weisen orangefarbene Rahmen auf eine Online-Verbindung hin. Nun können Sie die Tools unter "Online & Diagnose" in der Projektnavigation und die Taskcard "Online-Tools" verwenden.

Siehe auch

Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161)

15.3 Einem PROFINET IO-Gerät online einen Namen zuweisen

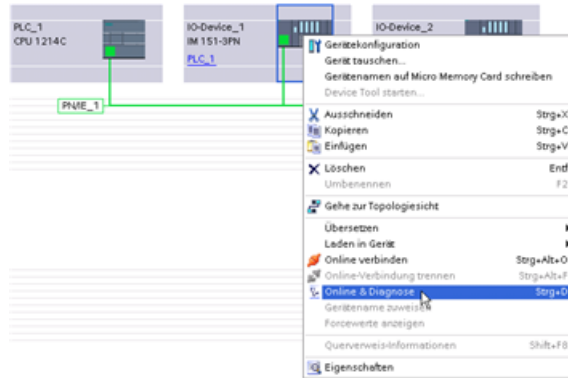
Den Geräten in Ihrem PROFINET-Netzwerk muss ein Name zugewiesen sein, damit Sie eine Verbindung zur CPU herstellen können. Sie weisen Ihren PROFINET-Geräten im Editor "Geräte & Netze" Namen zu, wenn die Geräte noch keinen Namen haben oder wenn der Name eines Geräts geändert werden soll.

Den Namen eines PROFINET IO-Device müssen Sie sowohl im STEP 7-Projekt als auch über das Werkzeug "Online & Diagnose" im Konfigurationsspeicher des PROFINET IO-Device zuweisen (z. B. im Konfigurationsspeicher eines ET200 S-Schnittstellenmoduls). Fehlt ein Name oder

15.3 Einem PROFINET IO-Gerät online einen Namen zuweisen

entsprechen sich die Namen an den beiden Speicherorten nicht, funktioniert der Modus für den PROFINET IO-Datenaustausch nicht.

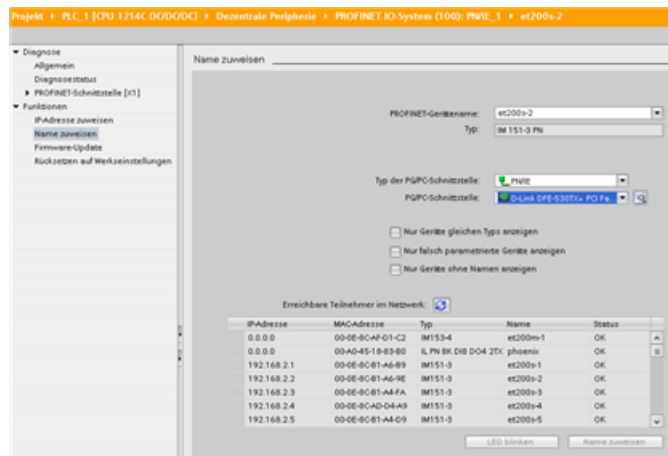
1. Klicken Sie im Editor "Geräte & Netze" mit der rechten Maustaste auf das gewünschte PROFINET IO-Device und wählen Sie "Online & Diagnose".



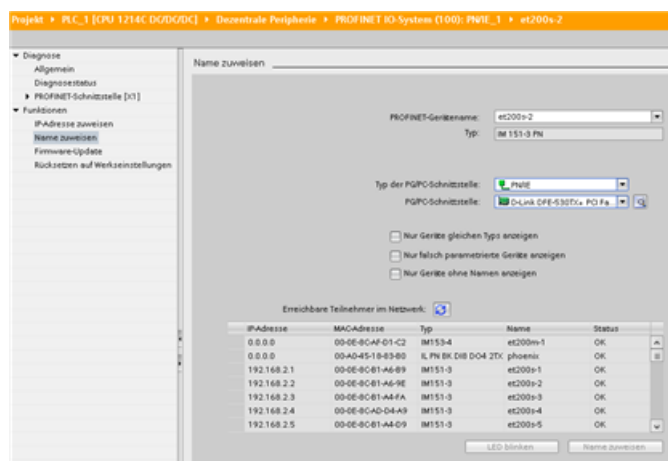
2. Wählen Sie im Dialog "Online & Diagnose" die folgenden Menübefehle:

- "Funktionen"
- "PROFINET-Gerätename zuweisen"

Klicken Sie auf die Schaltfläche "Liste aktualisieren", um alle PROFINET IO-Geräte im Netzwerk anzuzeigen.



3. Klicken Sie in der daraufhin angezeigten Liste auf das gewünschte PROFINET IO-Device und dann auf die Schaltfläche "Name zuweisen", um den Namen in den Konfigurationsspeicher des PROFINET IO-Device zu schreiben.



15.4 Einstellen der IP-Adresse und der Uhrzeit

Sie können die IP-Adresse (Seite 606) und die Uhrzeit der Online-CPU einstellen. Nach dem Zugriff auf "Online & Diagnose" in der Projektnavigation einer Online-CPU können Sie die IP-Adresse anzeigen oder ändern. Ebenso können Sie Datum und Uhrzeit der CPU online aufrufen und ändern.



Hinweis

Diese Funktion ist nur für eine CPU verfügbar, die entweder nur eine MAC-Adresse hat (der noch keine IP-Adresse zugewiesen wurde) oder die auf die Werkseinstellungen zurückgesetzt wurde.

15.5 Firmware aktualisieren

Sie können die Firmware einer verbundenen CPU über die Online- und Diagnosetools in STEP 7 auf eine der folgenden Arten aktualisieren:

- Aktualisierung über die CPU im Projekt
- Aktualisierung über die erreichbaren Teilnehmer in der Projektnavigation

Aktualisierung der Firmware einer CPU in Ihrem Projekt

Für ein Firmware-Update gehen Sie folgendermaßen vor:

1. Öffnen Sie die CPU in der Projektnavigation, die der verbundenen CPU entspricht.
2. Öffnen Sie die Online- und Diagnoseansicht der angeschlossenen CPU.
3. Wählen Sie im Ordner "Funktionen" den Eintrag "Firmware-Update".
4. Klicken Sie im Bereich "Firmware-Lader" auf die Schaltfläche "Durchsuchen" und navigieren Sie zum Speicherort der Firmware-Updatedatei. Hierbei kann es sich um einen Speicherort auf Ihrer Festplatte handeln, wo Sie das Firmware-Update S7-1200 (<http://support.automation.siemens.com/WW/view/de/34612486/133100>) von der Website Siemens Industry Online-Support Web (<https://support.industry.siemens.com/cs/ww/de>) heruntergeladen und gespeichert haben.

15.6 Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten festlegen oder löschen

5. Wählen Sie eine Datei, die für Ihr Modul geeignet ist. In der Tabelle werden die kompatiblen Module für die ausgewählte Datei angezeigt.
6. Klicken Sie auf die Schaltfläche "Aktualisierung ausführen". Folgen Sie ggf. den Anweisungen in den Dialogen, um den Betriebszustand Ihrer CPU zu ändern.

Während das Firmware-Update geladen wird, wird der Fortschritt in STEP 7 angezeigt. Nach beendetem Update werden Sie aufgefordert, das Modul mit der neuen Firmware zu starten.

Hinweis

Wollen Sie das Modul mit der neuen Firmware nicht starten, bleibt die alte Firmware aktiv, bis das Modul, zum Beispiel durch Aus- und Einschalten, zurückgesetzt wird. Die neue Firmware wird erst nach dem Rücksetzen des Moduls aktiv.

Aktualisieren der Firmware über die erreichbaren Teilnehmer

Um ein Firmware-Update für einen oder mehrere erreichbare Teilnehmer durchzuführen, gehen Sie wie folgt vor:

1. Öffnen Sie "Onlinezugriff" in der Projektnavigation.
2. Öffnen Sie die Kommunikationsschnittstelle, mit der Ihre CPU verbunden ist.
3. Doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren" und warten Sie, bis STEP 7 die Online-Geräte anzeigt.
4. Erweitern Sie die CPU, die Sie aktualisieren möchten, und doppelklicken Sie auf "Online & Diagnose".
5. Erweitern Sie im Ordner "Funktionen" den Eintrag "Firmware-Update". Sie sehen den PLC sowie die lokalen Module für den PLC. Über den Bereich "PLC" oder den Bereich "Lokale Baugruppen" können Sie die Aktualisierung der Firmware wie oben beschrieben im Bereich "Firmware-Lader" fortsetzen.

Ein Firmware-Update können Sie auch auf eine der folgenden Arten durchführen:

- Mit einer SIMATIC Memory Card (Seite 131)
- Über die Standard-Webseite "Modulinformationen" des Webservers (Seite 872)
- Mit dem SIMATIC Automation Tool (<https://support.industry.siemens.com/cs/ww/de/view/98161300>)

15.6 Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten festlegen oder löschen

Mit der Funktion "Schutz vertraulicher PLC-Konfigurationsdaten" können Sie jede CPU in Ihrem Projekt einzeln schützen. In der Gerätekonfiguration (Seite 161) können Sie diesen Schutz aktivieren und ein Passwort für den "Schutz vertraulicher PLC-Konfigurationsdaten" festlegen. Clients wie das TIA Portal und das SIMATIC Automation Tool können nur nach Eingabe des Passworts auf die vertraulichen Daten in der PLC zugreifen.

Sie können diese Funktion auch im Security-Assistenten (Seite 160) aktivieren und das Passwort für den "Schutz vertraulicher PLC-Konfigurationsdaten" festlegen.

Verwenden der Online- und Diagnosetools

Wenn das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten in der CPU nicht mit dem im Projekt übereinstimmt, kann die CPU nicht in den Betriebszustand RUN wechseln. Sie müssen das richtige Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten festlegen oder es löschen, damit die CPU in den Betriebszustand RUN wechseln kann.

Wenn die CPU online (Seite 1209) ist, können Sie das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten in der Online-CPU wie folgt festlegen oder löschen:

1. Versetzen Sie die CPU in den Betriebszustand STOP.
2. Öffnen Sie die Online- und Diagnosetools für Ihre CPU.
3. Wählen Sie im Menü "Funktionen" die Option "Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten festlegen".
4. Klicken Sie auf "Festlegen", um ein Passwort festzulegen, oder klicken Sie auf "Löschen", um das bestehende Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten in der Online-CPU zu löschen. Wenn "Löschen" nicht verfügbar ist, hat die Online-CPU kein Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten.

15.7 Rücksetzen auf Werkseinstellungen

Unter den folgenden Bedingungen können Sie eine S7-1200 auf die ursprünglichen Werkseinstellungen zurücksetzen:

- Die CPU hat eine Online-Verbindung.
- Die CPU befindet sich im Betriebszustand STOP.

Hinweis

Wenn sich die CPU im Betriebszustand RUN befindet und Sie den Rücksetzvorgang starten, können Sie die CPU nach Bestätigung einer Eingabeaufforderung in den Betriebszustand STOP versetzen.

Vorgehen

Um eine CPU auf die Werkseinstellungen zurückzusetzen, gehen Sie wie folgt vor:

1. Öffnen Sie die Ansicht "Online & Diagnose" der CPU.
2. Wählen Sie im Ordner "Funktionen" den Eintrag "Auf Werkseinstellungen zurücksetzen".
3. Aktivieren Sie das Kontrollkästchen "IP-Adresse löschen", wenn Sie die IP-Adresse löschen möchten.
4. Aktivieren Sie das Kontrollkästchen "Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten löschen", wenn Sie das Passwort löschen möchten. Sie müssen beispielsweise das Passwort löschen, wenn Sie ein neues Projekt in die CPU laden möchten oder die CPU durch ein neues Gerät ersetzen (Seite 1455).
5. Aktivieren Sie das Kontrollkästchen "Memory Card formatieren", wenn Sie die derzeit in der Online-CPU gesteckte Memory Card formatieren möchten. Wenn Sie Ihr CPU-Programm von der Memory Card aus ausführen und das Programm formatieren möchten, aktivieren Sie dieses Kontrollkästchen.

6. Klicken Sie auf die Schaltfläche "PLC zurücksetzen".
7. Quittieren Sie die Bestätigungsmeldung mit "Ja", um zu bestätigen, dass Sie das Modul mit Ihren Einstellungen zurücksetzen möchten.

Ergebnis

Das Modul wird gegebenenfalls in den Betriebszustand STOP versetzt und auf die Werkseinstellungen zurückgesetzt: Die Online-CPU führt die folgenden Vorgänge durch:

- Der Arbeitsspeicher und die remanenten Datenbereiche werden gelöscht
- Der Ladespeicher wird gelöscht, wenn es sich um internen Ladespeicher handelt; der Ladespeicher auf der SIMATIC Memory Card wird NUR gelöscht, wenn Sie auch "Memory Card formatieren" ausgewählt haben
- Alle Parameter und Operandenbereiche werden auf ihre konfigurierten Werte gesetzt
- Der Diagnosepuffer wird gelöscht
- Die Uhrzeit wird zurückgesetzt
- Identifikations- und Wartungsdaten (I&M) außer I&M0 werden gelöscht
- Die Betriebsstundenzähler werden zurückgesetzt
- Die IP-Adresse wird abhängig von Ihrer Auswahl beibehalten oder gelöscht. (Die MAC-Adresse ist fest zugewiesen und wird nie geändert.)
Wenn Sie "IP-Adresse löschen" nicht ausgewählt haben, behält die CPU die IP-Adresse, Subnetzmaske und Routeradresse (falls verwendet) aus den Einstellungen in Ihrer Hardwarekonfiguration bei, sofern Sie diese Werte nicht über das Anwenderprogramm oder ein anderes Tool geändert haben. In einem solchen Fall stellt die CPU die geänderten Werte wieder her.
- Sofern vorhanden wird der Steuerdatensatz (Seite 141) gelöscht
- Das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten wird abhängig von Ihrer Einstellung gelöscht oder beibehalten
- Die Memory Card wird formatiert, wenn eine Memory Card in der Online-CPU gesteckt ist und Sie die Option zum Formatieren der Memory Card ausgewählt haben

15.8 Prüfen eines Moduls auf Defekte (Servicedaten speichern)

Speichern von Servicedaten

Bei den S7-1200 CPUs V4.5 können Sie die Servicedaten eines Moduls speichern.

Im Wartungsfall benötigt der SIEMENS-Kundendienst Informationen über den Zustand eines Moduls Ihres Systems zu Diagnosezwecken. Tritt ein solcher Fall in Ihrem System ein, werden Sie vom Kundendienst möglicherweise gebeten, die Servicedaten des Moduls zu speichern und die Ergebnisdatei an den Kundendienst zu senden.

So speichern Sie die Servicedaten eines Moduls

Sie können die Servicedaten eines Moduls in der Online- oder Diagnoseansicht an den folgenden Stellen speichern:

Im Ordner "Funktionen" der Gruppe "Servicedaten speichern" besteht die Gruppe "Servicedaten speichern" aus den folgenden Bereichen:

- Online-Daten
- Speichern von Servicedaten

"Online-Daten"

Im Bereich "Online-Daten" werden die folgenden Daten des Moduls angezeigt:

- Artikelnummer (Bestellnummer)
- Firmwareversion
- Modulname (den haben Sie beim Konfigurieren der Hardware festgelegt)
- Baugruppenträger
- Steckplatz



"Servicedaten speichern"

Gehen Sie wie folgt vor, um eine Datei mit speziellen Servicedaten zu erzeugen und zu speichern:

1. Wählen Sie die Stelle im Dateisystem, an der Sie die Datei speichern möchten:
 - Verwenden Sie den voreingestellten Pfad im Feld "Pfad".
 - Klicken Sie auf die Schaltfläche mit den drei Punkten (Durchsuchen). Geben Sie im daraufhin angezeigten Dialog den gewünschten Pfad und den Dateinamen an.
2. Klicken Sie auf die Schaltfläche "Daten speichern".
3. Es wird eine Statusanzeige zum Lesen der Servicedaten angezeigt. Wenn die Servicedaten erfolgreich extrahiert wurden, wird die Meldung "Speichern der Servicedaten abgeschlossen" angezeigt.

Hinweise:

- Servicedaten können von der S7-1200 im Betriebszustand RUN oder STOP extrahiert werden. Im Zustand Defekt/Fatal können keine Servicedaten gelesen werden.
- Wenn Sie in der CPU eine Passwortschutzstufe programmiert haben, müssen Sie das Passwort eingeben, bevor Sie die Servicedaten extrahieren können. Die Authentifizierung mittels Passwort ist bei allen Schutzstufen erforderlich, weil beim Extrahieren der Servicedaten ein Datensatz geschrieben wird.
- Das TIA Portal unterstützt das Speichern einer Servicedatendatei.
- Die S7-1200 Servicedatenelemente in der Servicedatendatei sind verschlüsselt.

15.9 Formatierung einer SIMATIC Memory Card über STEP 7

Sie können die Memory Card in einer verbundenen CPU über die Online- und Diagnosetools in STEP 7 formatieren. Hierfür gehen Sie wie folgt vor:

1. Stellen Sie sicher, dass sich die CPU in STOP befindet. Wenn sich die CPU im Betriebszustand RUN befindet und Sie einen Formatierungsvorgang starten, werden Sie von STEP 7 aufgefordert, zu bestätigen, dass STEP 7 die CPU in STOP versetzen darf.
2. Stecken Sie eine Memory Card in die angeschlossene CPU.
3. Öffnen Sie für die verbundene CPU die Option Online & Diagnose entweder über die CPU im Projekt oder über die erreichbaren Teilnehmer unter Onlinezugriff in der Projektnavigation.
4. Wenn die CPU nicht online ist, wählen Sie für die verbundene CPU den Befehl "Online verbinden".
5. Wählen Sie im Menü "Funktionen" die Option "Memory Card formatieren".
6. Klicken Sie auf "Formatieren".
7. Bestätigen Sie die Meldung mit "Ja".

STEP 7 formatiert dann die Memory Card und zeigt bei Abschluss eine Meldung im Infofenster an. Die CPU befindet sich bei Abschluss des Formatierungsvorgangs im Betriebszustand STOP und die Leuchten STOP und MAINT blinken. Zu diesem Zeitpunkt können Sie nicht in den Betriebszustand RUN wechseln. Sie müssen eine der folgenden Maßnahmen ergreifen:

- Entnehmen Sie die Memory Card und starten Sie die CPU neu: Wenn der interne Ladespeicher der CPU ein Programm enthält, startet die CPU mit diesem Programm.
- Starten Sie die CPU neu, ohne die Memory Card zu entnehmen: Wenn der interne Ladespeicher der CPU ein Programm enthält, kopiert die CPU dieses Programm in die Memory Card und startet mit diesem Programm. Wenn der interne Ladespeicher kein Programm enthält, ändert die CPU die Memory Card in eine Programmkarte (Seite 124) und wartet auf einen Download.

Risiken im Zusammenhang mit der Außerbetriebsetzung

S7-1200 CPUs unterstützen nicht das sichere Löschen der Memory Card und des internen Flash-Speichers. Deshalb müssen Sie bei der Außerbetriebsetzung die CPU und die Memory Card sicher entsorgen, um den Verlust proprietärer und vertraulicher Informationen zu verhindern.

Hinweis

Das Formatieren einer Memory Card wirkt sich nicht auf den Inhalt des internen Ladespeichers aus.

Wenn die CPU zu dem Zeitpunkt, zu dem Sie die Memory Card gesteckt hatten, den internen Ladespeicher verwendete und Sie die CPU zwischen dem Stecken der Memory Card und dem Ausführen des Formatierungsvorgangs nicht neu gestartet haben, speichert die CPU weiterhin den Inhalt des internen Ladespeichers.

15.10 Bedienpanel für die Online-CPU



Im CPU-Bedienpanel wird der Betriebszustand (STOP oder RUN) der Online-CPU angezeigt. Hier wird auch angezeigt, ob ein Fehler in der CPU aufgetreten ist oder ob Werte gefordert wurden.

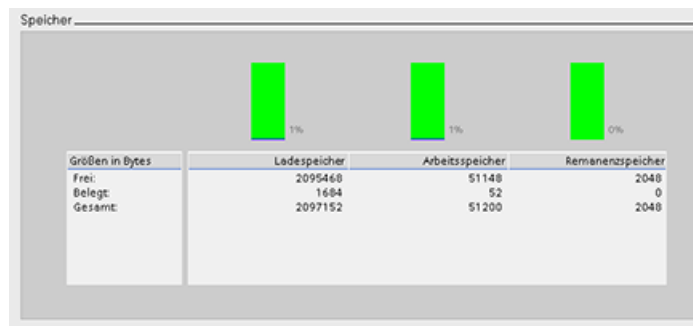
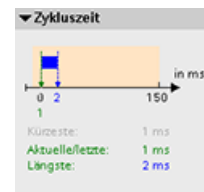
Im CPU-Bedienpanel der Taskcard "Online-Tools" können Sie den Betriebszustand einer Online-CPU ändern. Die Taskcard "Online-Tools" ist verfügbar, wenn die CPU online ist.

15.11 Überwachung von Zykluszeit und Speicherauslastung

Sie können die Zykluszeit und die Speicherauslastung einer Online-CPU überwachen.

Nach dem Herstellen der Verbindung zur Online-CPU öffnen Sie die Taskcard "Online-Tools", um die folgenden Messwerte anzuzeigen:

- Zykluszeit
- Speicherauslastung

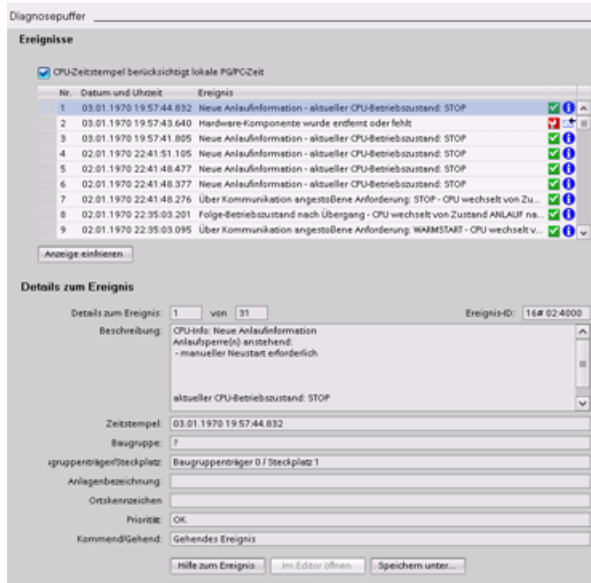


15.12 Diagnoseereignisse in der CPU anzeigen

Im Diagnosepuffer können Sie die letzten Ereignisse in der CPU betrachten. Der Diagnosepuffer ist in der Projektnavigation über "Online & Diagnose" für eine Online-CPU verfügbar. Er enthält die folgenden Einträge:

- Diagnoseereignisse
- Änderungen im Betriebszustand der CPU (Wechsel zwischen STOP und RUN)

15.12 Diagnoseereignisse in der CPU anzeigen



Der erste Eintrag entspricht dem jüngsten Ereignis. Jeder Eintrag im Diagnosepuffer enthält das Datum und die Uhrzeit, zu denen das Ereignis erfasst wurde, und eine Beschreibung.

Wie viele Einträge maximal möglich sind, hängt von der CPU ab. Es werden bis zu 50 Einträge unterstützt.

Nur die letzten 10 Ereignisse im Diagnosepuffer werden nullspannungsfest gespeichert. Wird die CPU auf die Werkseinstellungen zurückgesetzt, so werden dadurch die Einträge im Diagnosepuffer gelöscht.

Die Diagnoseinformationen können Sie auch mit der Anweisung GET_DIAG (Seite 459) erfassen.

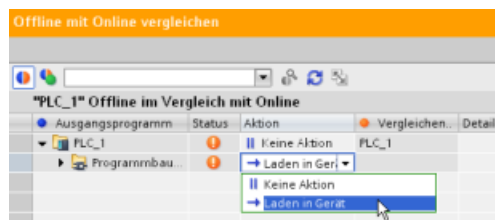
15.13 Vergleichen von Offline- und Online-CPU

Sie können die Codebausteine in einer Online-CPU mit den Codebausteinen Ihres Projekts vergleichen. Wenn die Codebausteine Ihres Projekts nicht den Codebausteinen der Online-CPU entsprechen, haben Sie im Editor "Vergleichen" die Möglichkeit, Ihr Projekt mit der Online-CPU abzugleichen. Laden Sie dazu entweder die Codebausteine Ihres Projekts in die CPU oder löschen Sie die Bausteine aus dem Projekt, die in der Online-CPU nicht vorhanden sind.



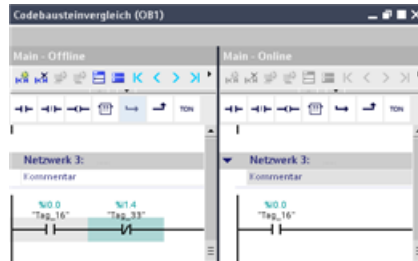
Wählen Sie die CPU in Ihrem Projekt aus.

Wählen Sie im Editor "Vergleichen" den Befehl "Offline/online vergleichen". (Rufen Sie den Befehl entweder über das Menü "Werkzeuge" oder durch Rechtsklick auf die CPU in Ihrem Projekt auf.)



Klicken Sie in die Spalte "Aktion" eines Objekts, um auszuwählen, ob das Objekt gelöscht, keine Maßnahme durchgeführt oder das Objekt in das Gerät geladen werden soll.

Durch Klicken auf die Schaltfläche "Synchronisieren" laden Sie die Codebausteine.



Klicken Sie in der Spalte "Vergleichen mit" mit der rechten Maustaste auf ein Objekt und wählen Sie "Detaillierten Vergleich starten", um die Codebausteine nebeneinander anzuzeigen.

Bei diesem Detailvergleich werden die Unterschiede zwischen den Codebausteinen der Online-CPU und den Codebausteinen der CPU in Ihrem Projekt hervorgehoben.

Hinweis

Lesezugriff erforderlich auf geschützte CPU für den Offline-/Online-Vergleich

Für STEP 7 V14 oder neuer ist die Sicherheitsstufe "HMI-Zugriff" nicht ausreichend für den Offline-/Online-Vergleich. Für Offline-/Online-Vergleiche benötigen Sie den "Lesezugriff" oder "Vollen Zugriff".


Siehe auch Zugriffsschutz für die CPU (Seite 163)

15.14 Durchführen eines Online/Offline-Topologievergleichs

In der Topologieübersicht in STEP 7 können Sie die konfigurierte Offline-Topologie mit der tatsächlichen Online-Topologie vergleichen.






Vorgehensweise

Um die Unterschiede zwischen der konfigurierten und der tatsächlichen Topologie zu ermitteln, gehen Sie wie folgt vor:



1. Rufen Sie die Topologieübersichtstabelle der Topologieansicht auf.
2. Klicken Sie in der Symbolleiste der Topologieübersicht auf die Schaltfläche „Offline/Online-Vergleich“: 


Ergebnis

STEP 7 entfernt die Spalten "Partnerstation", „Partnerschnittstelle“ und „Kabeldaten" in der Topologieübersichtstabelle und fügt Vergleichsspalten für "Status" und „Aktion“ ein. Für jedes Gerät oder jeden Port in der Topologieübersicht zeigt die Statusspalte den Vergleichsstatus wie folgt an:

Symbol	Bedeutung
	Unterschiedliche Topologie in mindestens einer untergeordneten Komponente
	Identische Topologie
	Topologieinformationen sind nur offline verfügbar oder das Gerät ist deaktiviert
	Topologieinformationen sind nur online verfügbar
	Unterschiedliche Topologie
	Gerät unterstützt keine Topologiefunktionen

Die Aktionsspalte bietet für jeden verglichenen Port oder jedes verglichene Gerät die folgenden möglichen Optionen:

Symbol	Bedeutung
	Keine Aktion möglich
	Online-Verschaltung übernehmen

Um den Vergleich zu wiederholen, klicken Sie in der Symbolleiste der Topologieübersicht auf die Schaltfläche .

Weitere Informationen zur Topologieansicht, zur Topologieübersicht und dem Online/Offline-Topologievergleich finden Sie im STEP 7 Informationssystem. Zusätzlich finden Sie weitere Informationen im Handbuch PROFINET mit STEP 7 V13 (<https://support.industry.siemens.com/cs/ww/de/view/49948856>).

15.15 Werte in der CPU beobachten und steuern

STEP 7 bietet Online-Funktionen zum Beobachten der CPU:

- Sie können die aktuellen Werte der Variablen anzeigen oder beobachten. Die Beobachtungsfunktion ändert den Programmablauf nicht. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU.
- Sie können den Ablauf und die Daten des Anwenderprogramms auch mit anderen Funktionen steuern:
 - Sie können den Wert der Variablen in der Online-CPU steuern, um zu ermitteln, wie das Anwenderprogramm reagiert.
 - Sie können einen Peripherieausgang (wie A0.1:P oder "Start":P) auf einen bestimmten Wert forcen.
 - Sie können Ausgänge im Betriebszustand STOP freischalten.

Hinweis

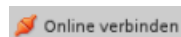
Verwenden Sie die Steuerfunktionen stets mit Vorsicht. Diese Funktionen können die Ausführung des Anwender-/Systemprogramms schwerwiegend beeinflussen.

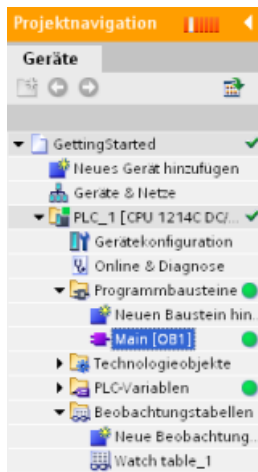
Tabelle 15-4 Online-Funktionen der STEP 7-Editoren

Editor	Beobachten	Steuern	Forcen
Beobachtungstabelle	Ja	Ja	Nein
Forcetabelle	Ja	Nein	Ja
Programmiereditor	Ja	Ja	Nein
Variablentabelle	Ja	Nein	Nein
DB-Editor	Ja	Nein	Nein

15.15.1 Online gehen, um die Werte in der CPU zu beobachten

Die Beobachtung der Variablen setzt eine Online-Verbindung zur CPU voraus. Klicken Sie in der Funktionsleiste einfach auf die Schaltfläche "Online verbinden".





Wenn Sie eine Verbindung zur CPU hergestellt haben, stellt STEP 7 die Überschriften der Arbeitsbereiche orangefarben dar.

In der Projektnavigation wird ein Vergleich des Offline-Projekts mit der Online-CPU angezeigt. Ein grüner Kreis bedeutet, dass die CPU und das Projekt synchronisiert sind, d. h. beide haben dieselbe Konfiguration und dasselbe Anwenderprogramm.

Variablentabellen zeigen die Variablen. Beobachtungstabellen können auch die Variablen anzeigen, ebenso wie direkte Adressen.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	*On	%E0.0	Bool		
2	*Off	%E0.1	Bool		
3	*Run	%A0.0	Bool		



Um die Ausführung des Anwenderprogramms zu beobachten und die Werte der Variablen anzuzeigen, klicken Sie in der Funktionsleiste auf die Schaltfläche "Alle beobachten".

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	*On	%E0.0	Bool	<input type="checkbox"/> FALSE	
2	*Off	%E0.1	Bool	<input type="checkbox"/> FALSE	
3	*Run	%A0.0	Bool	<input type="checkbox"/> FALSE	

Im Feld "Beobachtungswert" wird für jede Variable der Wert angezeigt.

15.15.2 Zustand im Programmiereditor anzeigen

Sie können den Zustand von bis zu 50 Variablen im KOP- und FUP-Editor beobachten. Rufen Sie den KOP-Editor über die Editorleiste auf. Über die Editorleiste können Sie zwischen den geöffneten Editoren umschalten, ohne die Editoren öffnen oder schließen zu müssen.

Klicken Sie im Programmiereditor in der Funktionsleiste auf die Schaltfläche "Beobachten ein/aus", um den Zustand Ihres Anwenderprogramms anzuzeigen.





Im Netzwerk im Programmiereditor wird der Signalfluss grün dargestellt.

Sie können auch mit der rechten Mastaste auf die Anweisung oder den Parameter klicken, um den Wert der Anweisung zu ändern.

15.15.3 Erfassen einer Momentaufnahme der Online-Werte eines DBs zum Wiederherstellen von Werten

Sie können von einer Online-CPU eine Momentaufnahme der Istwerte von Datenbausteinvariablen erfassen, um die Werte später zu verwenden.

Es gelten folgende Voraussetzungen:

- Sie benötigen eine Online-Verbindung zur CPU.
- Sie müssen den DB in STEP 7 geöffnet haben.

Erfassen einer Momentaufnahme

Um eine Momentaufnahme zu erfassen, gehen Sie wie folgt vor:

1. Klicken Sie im DB-Editor auf die Schaltfläche "Alle Variablen beobachten": Die Spalte "Beobachtungswert" zeigt die Istdatenwerte an.
2. Klicken Sie auf die Schaltfläche , um eine Momentaufnahme der Istwerte zu erfassen und in der Spalte "Momentaufnahme" anzuzeigen.

Diese Momentaufnahme können Sie später verwenden, um die Istwerte der CPU zu aktualisieren oder um die Startwerte zu ersetzen.

Kopieren der Werte der Momentaufnahme in die CPU


Um die Werte der Momentaufnahme in die Istwerte der Datenbausteinvariablen in der CPU zu kopieren, klicken Sie auf die folgende Schaltfläche:

Die Online-CPU lädt die Werte der Momentaufnahme in die Istwerte. Die Spalte "Beobachtungswert" zeigt die Istwerte in der CPU. Der Zyklus ändert die Werte in der CPU möglicherweise später, doch zum Zeitpunkt des Kopierens lädt die CPU die Werte der Momentaufnahme in einem konsistenten Ladevorgang.

Hinweis

Wenn Ihre Momentaufnahme Zustandsinformationen, Zeitwerte oder berechnete Informationen enthält, stellt die CPU diese Werte so wieder her, wie sie zu dem Zeitpunkt der Momentaufnahme waren.

Kopieren der Werte der Momentaufnahme in die Startwerte

Um die Werte der Momentaufnahme in die Startwerte der Datenbausteinvariablen in der CPU zu kopieren, klicken Sie auf die folgende Schaltfläche: 

Nachdem Sie den DB übersetzt und in die CPU geladen haben, verwendet der DB die neuen Startwerte, wenn die CPU in den Betriebszustand RUN geht.

Kopieren einzelner Momentaufnahme- oder Beobachtungswerte in die Startwerte

Im Datenbausteineditor können Sie auch einzelne Werte kopieren und in die Startwerte einfügen. Klicken Sie einfach mit der rechten Maustaste auf einen Wert in einer beliebigen Spalte und wählen Sie "Kopieren", um den Wert in der Zwischenablage abzulegen. Dann können Sie mit der rechten Maustaste auf einen beliebigen Startwert klicken und "Einfügen" wählen, um den Wert durch den Wert aus der Zwischenablage zu ersetzen.

Nachdem Sie den DB übersetzt und in die CPU geladen haben, verwendet der DB die neuen Startwerte, wenn die CPU in den Betriebszustand RUN geht.

15.15.4 Werte in der CPU über die Beobachtungstabelle beobachten und steuern

Mit Hilfe einer Beobachtungstabelle können Sie die Datenpunkte beobachten und steuern, während die CPU Ihr Programm ausführt. Bei diesen Datenpunkten kann es sich je nach Beobachtungs- oder Steuerfunktion um das Prozessabbild (E oder A), um M, DB oder physische Eingänge (E_:P) handeln. Sie können die physischen Ausgänge (A_:P) nicht genau beobachten, weil die Beobachtungsfunktion nur den letzten geschriebenen Wert aus dem Speicherbereich A anzeigen kann und nicht den tatsächlichen Wert aus den physischen Ausgängen liest.

Die Beobachtungsfunktion ändert den Programmablauf nicht. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU.

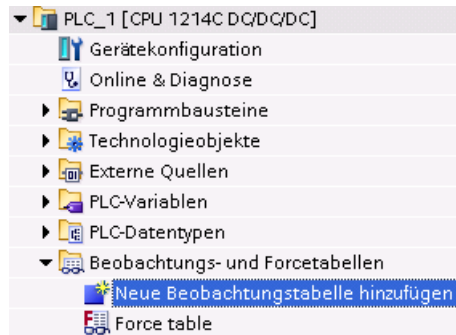
Die Steuerfunktionen ermöglichen es Ihnen, den Ablauf und die Daten des Programms zu steuern. Verwenden Sie die Steuerfunktionen stets mit Vorsicht. Diese Funktionen können die Ausführung des Anwender-/Systemprogramms schwerwiegend beeinflussen. Bei den drei Steuerfunktionen handelt es sich um Steuern, Forcen und Ausgänge in STOP freischalten.

Sie können mit der Beobachtungstabelle die folgenden Online-Funktionen ausführen:

- Status der Variablen beobachten
- Werte der einzelnen Variablen bearbeiten

Sie können auswählen, wann eine Variable beobachtet oder geändert werden soll:

- Zyklusbeginn: Der Wert wird bei Zyklusbeginn gelesen oder geschrieben
- Zyklusende: Der Wert wird bei Zyklusende gelesen oder geschrieben
- Umschalten nach Stop



So erstellen Sie eine Beobachtungstabelle:

1. Öffnen Sie mit Doppelklick auf "Neue Beobachtungstabelle hinzufügen" eine neue Beobachtungstabelle.
2. Geben Sie den Namen einer Variablen ein, die in der Beobachtungstabelle hinzugefügt werden soll.

Für die Beobachtung der Variablen stehen die folgenden Möglichkeiten zur Verfügung:

- Alle beobachten: Mit diesem Befehl wird die Beobachtung der sichtbaren Variablen in der aktiven Beobachtungstabelle gestartet.
- Jetzt beobachten: Mit diesem Befehl wird die Beobachtung der sichtbaren Variablen in der aktiven Beobachtungstabelle gestartet. Die Beobachtungstabelle beobachtet die Variablen sofort und nur einmal.

Zum Ändern der Variablen stehen die folgenden Möglichkeiten zur Verfügung:

- "Steuern auf 0" setzt den Wert der ausgewählten Adresse auf "0".
- "Steuern auf 1" setzt den Wert der ausgewählten Adresse auf "1".
- "Steuern jetzt" ändert den Wert der ausgewählten Adresse sofort und nur für einen Zyklus.
- "Steuern mit Trigger" ändert die Werte für die ausgewählten Adressen. Diese Funktion erzeugt keine Rückmeldung, mit der die Änderung der ausgewählten Adressen bestätigt wird. Wird eine Bestätigung der Änderung benötigt, so ist die Funktion "Steuern jetzt" zu verwenden.
- "PA freischalten" deaktiviert den Befehl zur Sperrung von Ausgängen und steht nur zur Verfügung, wenn die CPU im Betriebszustand STOP ist.

Diese Beobachtung der Variablen setzt jedoch eine Online-Verbindung zur CPU voraus.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Beobachten mit Trigger	Steuern mit Trigger	Steuernwert
1	"Start"	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	"Stop"	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	"Running"	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

Mit den Schaltflächen am oberen Rand der Beobachtungstabelle können Sie die verschiedenen Funktionen auswählen.

Geben Sie den Namen der zu beobachtenden Variablen ein und wählen Sie in der Klappliste ein Anzeigeformat. Besteht eine Online-Verbindung zur CPU, wird durch Anklicken der Schaltfläche "Beobachten" der Istwert des Datenpunkts im Feld "Beobachtungswert" angezeigt.

15.15.4.1 Variablen mit Trigger beobachten oder steuern

Der Trigger legt fest, an welchem Punkt im Zyklus die ausgewählte Adresse beobachtet oder gesteuert wird.

Tabelle 15-5 Trigger-Arten

Trigger	Beschreibung
Permanent	Die Daten werden ständig erfasst
Zu Beginn des Zyklus	Permanent: Die Daten werden zu Beginn des Zyklus, nachdem die CPU die Eingänge gelesen hat, ständig erfasst
	Einmalig: Die Daten werden zu Beginn des Zyklus, nachdem die CPU die Eingänge gelesen hat, erfasst
Am Ende des Zyklus	Permanent: Die Daten werden am Ende des Zyklus, bevor die CPU in die Ausgänge schreibt, ständig erfasst
	Einmalig: Die Daten werden am Ende des Zyklus, bevor die CPU in die Ausgänge schreibt, erfasst
Beim Wechsel in STOP	Permanent: Die Daten werden ständig erfasst, wenn die CPU in STOP wechselt
	Einmalig: Die Daten werden einmal erfasst, nachdem die CPU in STOP wechselt


Um eine PLC-Variable mit einem bestimmten Trigger zu steuern, wählen Sie entweder den Anfang oder das Ende des Zyklus aus.

- **Steuern eines Ausgangs:** Das beste Triggerereignis zum Steuern eines Ausgangs ist am Ende des Zyklus, unmittelbar bevor die CPU in die Ausgänge schreibt. Beobachten Sie den Wert der Ausgänge am Anfang des Zyklus, um zu ermitteln, welcher Wert in die physischen Ausgänge geschrieben wird. Beobachten Sie auch die Ausgänge, bevor die CPU die Werte in die physischen Ausgänge schreibt, um die Programmlogik zu prüfen und mit dem tatsächlichen E/A-Verhalten zu vergleichen.
- **Steuern eines Eingangs:** Das beste Triggerereignis zum Steuern eines Eingangs ist am Anfang des Zyklus, unmittelbar nachdem die CPU die Eingänge gelesen hat und bevor das Anwenderprogramm die Eingangswerte nutzt. Wenn Sie den Verdacht haben, dass sich Werte während des Zyklus ändern, kann es sinnvoll sein, den Wert der Eingänge am Zyklusende zu überwachen, um sicherzugehen, dass sich die Werte im Zyklusverlauf nicht geändert haben. Unterscheiden sich die Werte, schreibt Ihr Anwenderprogramm möglicherweise in einen falschen Eingang.

Um zu ermitteln, warum die CPU in STOP gegangen ist, erfassen Sie mit dem Trigger "Wechsel in STOP" die letzten Prozesswerte.

15.15.4.2 Ausgänge im Betriebszustand STOP freischalten

Mit der Beobachtungstabelle können Sie in die Ausgänge schreiben, wenn sich die CPU im Betriebszustand STOP befindet. Mithilfe dieser Funktionalität können Sie die Verdrahtung der Ausgänge prüfen und sicherstellen, dass der an eine Ausgangsklemme angeschlossene Draht das Signal 1 oder 0 an die Klemme des angeschlossenen Prozessgeräts weitergibt.

 WARNUNG
Gefahren beim Schreiben in physische Ausgänge im Betriebszustand STOP
Auch wenn sich die CPU im Betriebszustand STOP befindet, kann die Freischaltung eines physischen Ausgangs den daran angeschlossenen Prozesspunkt aktivieren und zu unerwünschtem Verhalten der Geräte führen. Dies kann tödliche oder schwere Verletzungen zur Folge haben.
Vor dem Schreiben in einen Ausgang in der Beobachtungstabelle ist daher sicherzustellen, dass eine Änderung des physischen Ausgangswerts kein unerwünschtes Verhalten von Geräten hervorrufen kann. Beachten Sie immer die Sicherheitsvorkehrungen für Ihre Prozessanlagen.

Sie können den Zustand der Ausgänge im Betriebszustand STOP ändern, wenn die Ausgänge aktiviert sind. Sind die Ausgänge deaktiviert, können Sie die Ausgänge im Betriebszustand STOP nicht steuern. Gehen Sie wie folgt vor, um die Änderung von Ausgangswerten aus der Beobachtungstabelle im Betriebszustand STOP zu ermöglichen:

1. Wählen Sie im Menü "Online" den Menübefehl "Erweiterter Modus".
2. Wählen Sie im Menü "Online" den Befehl "Steuern" und dann die Option "PA freischalten" oder klicken Sie im Kontextmenü mit der rechten Maustaste auf die entsprechende Zeile der Beobachtungstabelle.

Wenn Sie dezentrale Peripherie konfiguriert haben, können Sie in der Betriebsart STOP keine Ausgänge aktivieren. Versuchen Sie es trotzdem, wird ein Fehler gemeldet.

Wenn die CPU in den Betriebszustand RUN wechselt, wird die Option "PA freischalten" deaktiviert.

Wenn Eingänge oder Ausgänge geforct sind, kann die CPU im Betriebszustand STOP keine Ausgänge aktivieren. Die Force-Funktion muss zunächst beendet werden.

15.15.5 Werte in der CPU forcen

15.15.5.1 Arbeiten mit der Forcetabelle

Eine Forcetabelle bietet die Funktion "Forcen", die den Wert eines Eingangs oder Ausgangs zwangsweise auf einen vorgegebenen Wert für die Adresse des Peripherieeingangs bzw. -ausgangs setzt. Das Forcen wird im Prozessabbild der Eingänge vor der Ausführung des Anwenderprogramms und im Prozessabbild der Ausgänge vor dem Schreiben der Ausgänge in die Module durchgeführt.

Hinweis

Die Forcewerte werden in der CPU und nicht in der Forcetabelle gespeichert.

Sie können keinen Eingang (Adresse "E") oder Ausgang (Adresse "A") forcen. Sie können jedoch einen Peripherieingang oder einen Peripherieausgang forcen. Die Forcetabelle hängt automatisch ein ":P" an die Adresse an (Beispiel: "On":P oder "Run":P).

	Name	Adresse	Anzeigeformat	Beobachtungswert	Forcewert	F
1	"On":P	%I0.0:P	Bool		TRUE	<input checked="" type="checkbox"/>
2	"Off":P	%I0.1:P	Bool			<input type="checkbox"/>
3	"Run":P	%Q0.1:P	Bool			<input type="checkbox"/>

Geben Sie den Wert für den zu forcenden Eingang oder Ausgang in die Zelle "Forcewert" ein. Sie können dann das Kontrollkästchen in der Spalte "Forcen" aktivieren, um das Forcen des Eingangs oder Ausganges zu aktivieren.



Klicken Sie auf die Schaltfläche "Forcen starten oder ersetzen", um die Werte der Variablen in der Forcetabelle zu forcen. Klicken Sie auf die Schaltfläche "Forcen beenden", um die Werte der Variablen zurückzusetzen.

In der Forcetabelle können Sie den Status des geforcten Werts eines Eingangs beobachten. Den geforcten Wert eines Ausganges können Sie jedoch nicht beobachten.

Sie können den Zustand der geforcten Werte auch im Programmiereditor anzeigen.



Hinweis

Wenn ein Eingang oder Ausgang in einer Forcetabelle geforct wird, werden die Forceaktionen Teil der Projektkonfiguration. Beim Schließen von STEP 7 bleiben die geforcten Elemente im CPU-Programm so lange aktiv, bis sie gelöscht werden. Um diese geforcten Elemente zu löschen, müssen Sie über STEP 7 eine Verbindung zur Online-CPU herstellen und dann mithilfe der Forcetabelle die Forcefunktion für diese Elemente deaktivieren oder stoppen.

15.15.5.2 Funktionsweise der Forcefunktion

Die CPU gestattet Ihnen das Forcen von Eingängen und Ausgängen, indem Sie in der Forcetabelle die Adresse der physischen Eingänge und Ausgänge (E_:P oder A_:P) angeben und dann die Forcefunktion starten.

Im Programm werden die gelesenen Werte der physischen Eingänge durch den Forcewert überschrieben. Das Programm nutzt den geforcten Wert während der Bearbeitung. Wenn das Programm in einen physischen Ausgang schreibt, wird der Ausgangswert durch den Forcewert

überschrieben. Der geforcte Wert erscheint am physischen Ausgang und wird im Prozess verwendet.

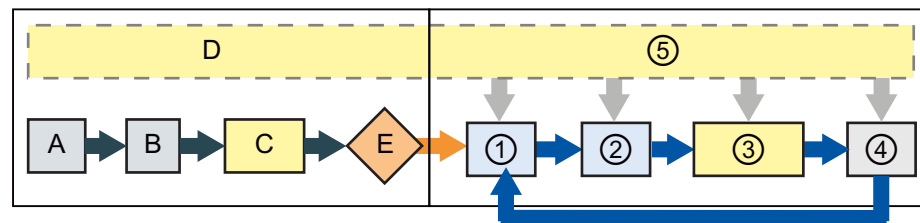
Wenn ein Eingang oder Ausgang in der Forcetabelle geforct wird, werden die Forceaktionen Teil des Anwenderprogramms. Auch wenn Sie die Programmiersoftware schließen, bleiben die geforcten Werte im ausgeführten CPU-Programm solange aktiv, bis Sie wieder in der Programmiersoftware online gehen und die Forcefunktion beenden. Programme mit geforcten Ein- und Ausgängen, die von einer Memory Card in eine andere CPU geladen werden, forcen auch weiterhin die im Programm ausgewählten Ein- und Ausgänge.

Wenn die CPU das Anwenderprogramm auf einer schreibgeschützten Memory Card ausführt, können Sie das Forcen von E/A nicht über eine Beobachtungstabelle auslösen oder ändern, weil Sie die Werte in dem schreibgeschützten Anwenderprogramm nicht überschreiben können. Jeder Versuch, die schreibgeschützten Werte zu forcen, führt zu einem Fehler. Bei Verwendung einer Memory Card zum Übertragen eines Anwenderprogramms werden auf dieser Memory Card gespeicherte geforcte Elemente mit an die CPU übertragen.

Hinweis

Zu HSC, PWM und PTO zugewiesene digitale E/A können nicht geforct werden

Die vom schnellen Zähler (HSC), von der Impulsdauermodulation (PWM) und von der Impulsfolge (PTO) verwendeten E/A werden während der Konfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Forcefunktion der Forcetabelle geändert werden.



Anlauf

- A Das Löschen des Speicherbereichs E wird von der Forcefunktion nicht beeinflusst.
- B Die Initialisierung der Ausgangswerte wird von der Forcefunktion nicht beeinflusst.
- C Während der Ausführung der Anlauf-OBs schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift.
- D Das Speichern von Alarmereignissen in der Warteschlange wird nicht beeinflusst.
- E Die Freigabe des Schreibens in die Ausgänge wird nicht beeinflusst.

RUN

- ① Beim Schreiben von A-Speicher in die physischen Ausgänge schaltet die CPU den Forcewert bei der Aktualisierung der Ausgänge auf.
- ② Beim Lesen der physischen Eingänge wendet die CPU die Forcewerte an, kurz bevor die Eingänge in den Speicherbereich E kopiert werden.
- ③ Während der Ausführung der Anwenderprogramms (Programmzyklus-OBs) schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift oder in den physischen Ausgang schreibt.
- ④ Die Behandlung von Kommunikationsanforderungen und die Selbsttestdiagnose werden von der Forcefunktion nicht beeinflusst.
- ⑤ Die Verarbeitung von Alarmen während eines beliebigen Teils des Zyklus wird nicht beeinflusst.

15.16 Laden im Betriebszustand RUN

Die CPU unterstützt das "Laden im Betriebszustand RUN". Diese Funktion soll Ihnen ermöglichen, kleinere Änderungen am Anwenderprogramm vorzunehmen, ohne den vom Programm gesteuerten Prozess zu stören. Diese Funktion ermöglicht jedoch auch größere Programmänderungen, die den Prozess beeinträchtigen oder sogar gefährlich werden können.



WARNUNG

Risiken beim Laden im Betriebszustand RUN

Wenn Sie im Betriebszustand RUN Änderungen in die CPU laden, wirken sich die Änderungen sofort auf den Prozess aus. Wenn Sie das Programm im Betriebszustand RUN ändern, kann dies zu unerwartetem Verhalten im Prozess führen und Tod, schwere Körperverletzungen und/oder Sachschaden können die Folge sein.

Nur dazu befugtes Personal mit Kenntnis der Auswirkungen einer Programmbearbeitung in RUN auf das Prozessverhalten darf einen Ladevorgang im Betriebszustand RUN durchführen.

Die Funktion zum "Laden im Betriebszustand RUN" ermöglicht Ihnen, Änderungen an einem Programm vorzunehmen und sie in die CPU zu laden, ohne nach STOP wechseln zu müssen.

- Sie können kleinere Änderungen am aktuellen Prozess vornehmen (z. B. eine Parameterwertänderung), ohne den Prozess herunterfahren zu müssen.
- Außerdem können Sie mit dieser Funktion Programmfehler schneller beheben (z. B. die Logik für einen Schließerkontakt oder Öffnerkontakt invertieren).

Sie können die folgenden Änderungen an Programmbausteinen und Variablen vornehmen und sie im Betriebszustand RUN laden:

- Funktionen (FCs), Funktionsbausteine (FBs) und Variablen tabellen erstellen, überschreiben und löschen
- Datenbausteine (DBs) und Instanz-Datenbausteine für Funktionsbausteine (FBs) erstellen, löschen und überschreiben. Sie können DB-Strukturen hinzufügen und im Betriebszustand RUN laden. Je nach Ihren Konfigurationseinstellungen (Seite 1237) kann die CPU die Werte vorhandener Bausteinvariablen speichern und die neuen Datenbausteinvariablen mit den Anfangswerten initialisieren oder alle Datenbausteinvariablen auf die Anfangswerte zurücksetzen. Im Betriebszustand RUN können Sie keinen Webserver-DB (Steuerung oder Fragment) laden.
- Organisationsbausteine (OBs) überschreiben; es können jedoch keine OBs erstellt oder gelöscht werden.

Sie können maximal zwanzig Bausteine gleichzeitig im Betriebszustand RUN laden. Müssen mehr als zwanzig Bausteine geladen werden, ist die CPU in STOP zu setzen.

Wenn Sie Änderungen in einen realen Prozess laden (im Unterschied zu einem simulierten Prozess wie z. B. bei der Fehlerbehebung in einem Programm), sollten Sie vor dem Laden

unbedingt in Gedanken die möglichen Folgen für die Sicherheit der Maschinen und Maschinenbediener durchspielen.

Hinweis

Befindet sich die CPU im Betriebszustand RUN und wurden Programmänderungen vorgenommen, versucht STEP 7 zunächst stets, die Änderungen im Betriebszustand RUN zu laden. Möchten Sie dies verhindern, müssen Sie die CPU in STOP versetzen.

Werden die vorgenommenen Änderungen nicht von der Funktion "Laden im Betriebszustand RUN" unterstützt, werden Sie von STEP 7 aufgefordert, die CPU in den Betriebszustand STOP zu versetzen.

15.16.1 Voraussetzungen für "Laden im Betriebszustand RUN"

Sie können Ihre Programmänderungen nur dann im Betriebszustand RUN in die CPU laden, wenn die folgenden Voraussetzungen erfüllt sind:

- Ihre CPU-Version weist mindestens V3.0 auf

Hinweis

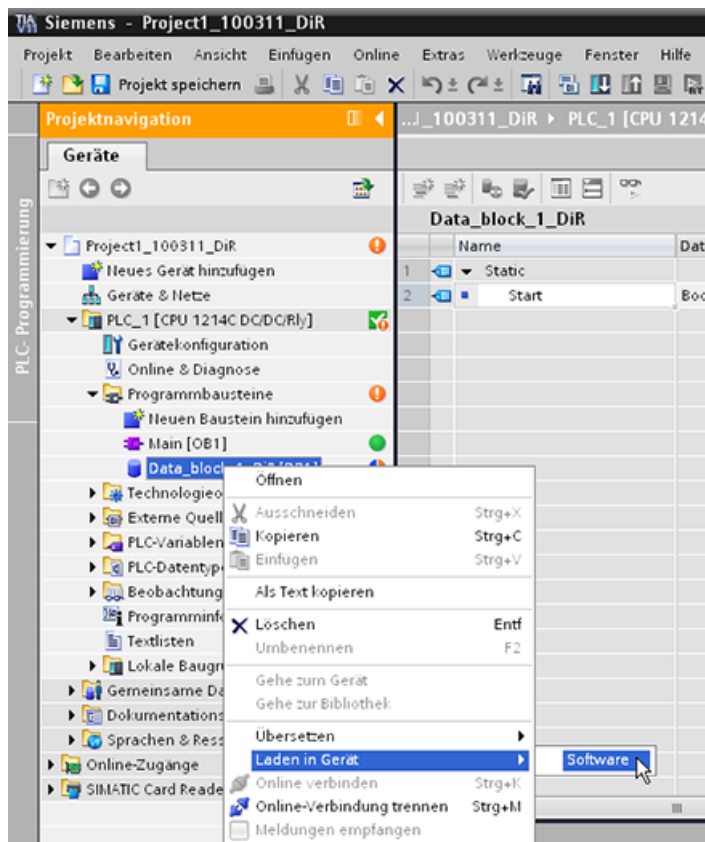
Sie benötigen eine CPU ab Version V4.0, um die erweiterte Bausteinschnittstelle im Betriebszustand RUN zu ändern. (Seite 1237)

- Ihr Programm muss sich erfolgreich übersetzen lassen.
- Sie müssen die Kommunikation zwischen dem Programmiergerät, auf dem STEP 7 läuft, und der CPU erfolgreich aufgebaut haben.

15.16.2 Ändern des Programms im Betriebszustand RUN

Um das Programm in RUN zu ändern, muss zuerst sichergestellt werden, dass die CPU und das Programm die Voraussetzungen (Seite 1233) erfüllen; dann ist wie folgt vorzugehen:

1. Um Ihr Programm im Betriebszustand RUN zu laden, gehen Sie auf eine der folgenden Arten vor:
 - Wählen Sie im Menü "Online" den Befehl "Laden in Gerät".
 - Klicken Sie in der Symbolleiste auf die Schaltfläche "Laden in Gerät".
 - Klicken Sie in der "Projektnavigation" mit der rechten Maustaste auf "Programmbausteine" und wählen Sie den Befehl "Laden in Gerät > Software".



Wenn das Programm erfolgreich übersetzt wird, lädt STEP 7 das Programm in die CPU.

2. Wenn STEP 7 Sie auffordert, das Programm zu laden oder den Vorgang abzubrechen, klicken Sie auf "Laden", um das Programm in die CPU zu laden.

15.16.3 Ausgewählte Bausteine laden

Im Ordner "Programmbausteine" können Sie einen einzelnen oder mehrere Bausteine zum Laden auswählen.

Wenn Sie einen einzelnen Baustein zum Laden auswählen, wird in der Spalte "Aktion" als einzige Option "Konsistent laden" angezeigt.

Sie können die Kategoriezeile erweitern, um zu prüfen, welche Bausteine geladen werden müssen. In diesem Beispiel wurde eine geringfügige Änderung am Offline-Baustein vorgenommen und es müssen keine anderen Bausteine geladen werden.

In diesem Beispiel müssen mehrere Bausteine geladen werden.



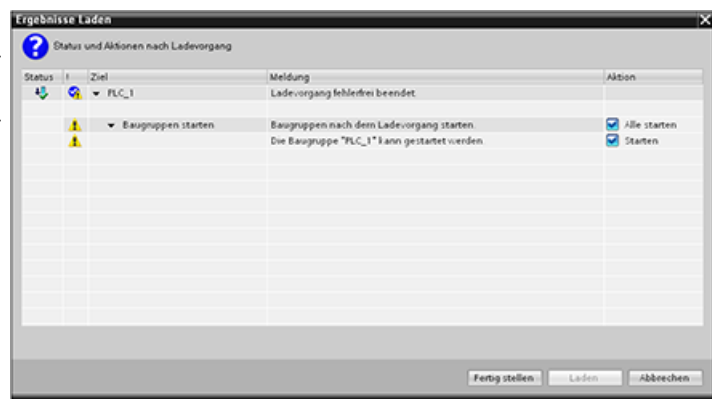
Hinweis

Sie können maximal zwanzig Bausteine gleichzeitig im Betriebszustand RUN laden. Müssen mehr als zwanzig Bausteine geladen werden, ist die CPU in STOP zu setzen.

Wenn Sie versuchen, die Bausteine in RUN zu laden, doch das System erkennt vor dem tatsächlichen Ladevorgang, dass dies nicht möglich ist, erscheint im Dialog eine Zeile zum Stoppen der Module.



Klicken Sie auf die Schaltfläche "Laden". Daraufhin wird der Dialog "Ladeergebnisse" angezeigt. Klicken Sie auf die Schaltfläche "Fertig stellen", um den Ladevorgang abzuschließen.



15.16.4 Einen einzelnen ausgewählten Baustein mit einem Übersetzungsfehler in einem anderen Baustein laden

Wenn Sie versuchen, einen konsistenten Ladevorgang mit einem Übersetzungsfehler in einem anderen Baustein durchzuführen, zeigt der Dialog einen Fehler an und die Schaltfläche zum Laden wird deaktiviert.



Sie müssen zunächst den Übersetzungsfehler in dem anderen Baustein korrigieren. Danach wird die Schaltfläche "Laden" wieder aktiviert.

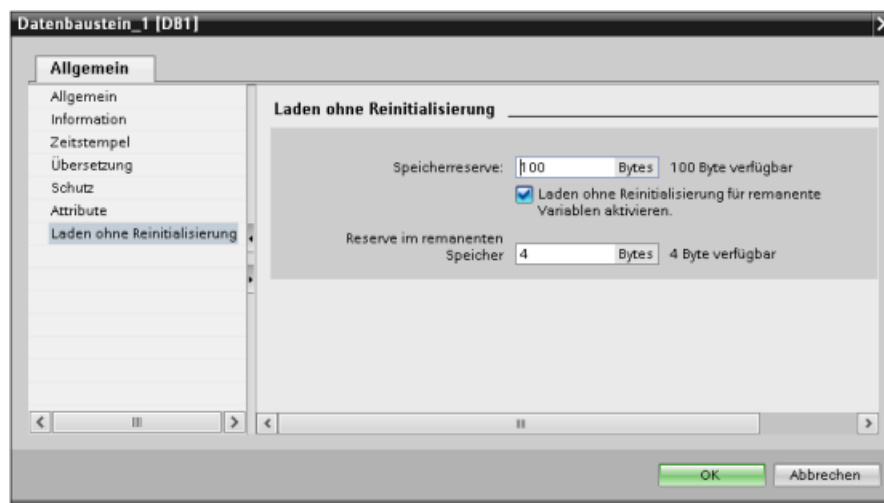


15.16.5 Bestehende Bausteine im Betriebszustand RUN ändern und ins Zielsystem laden

Mit der Funktion zum Laden im Betriebszustand RUN können Sie Variablen in Datenbausteinen und Funktionsbausteinen hinzufügen und ändern und dann den geänderten Baustein im Betriebszustand RUN in die CPU laden.

Laden ohne Reinitialisierung


Jeder DB und FB hat eine Speicherreserve, die Sie nutzen können, um dem Baustein Variablen hinzuzufügen und ihn nachfolgend im Betriebszustand RUN in die CPU zu laden. Standardmäßig beträgt die Anfangsgröße der Speicherreserve 100 Byte. Sie können im Rahmen der Größe der Speicherreserve zusätzliche Variablen zu Ihren Daten hinzufügen und den erweiterten Baustein im Betriebszustand RUN in die CPU laden. Sie können die Speicherreserve auch vergrößern, wenn Sie mehr Speicher für zusätzliche Variablen in Ihrem Baustein benötigen. Wenn Sie mehr Variablen hinzufügen als der Speicher aufnehmen kann, können Sie den erweiterten Baustein nicht im Betriebszustand RUN in die CPU laden.




Mit der Funktion "Laden ohne Reinitialisierung" können Sie einen Datenbaustein erweitern, indem Sie mehr Datenbausteinvariablen hinzufügen und den erweiterten Datenbaustein im

Betriebszustand RUN in die CPU laden. Auf diese Weise können Sie einem Datenbaustein Variablen hinzufügen und den DB ohne Reinitialisierung in Ihr Programm laden. Die CPU speichert die Werte der bestehenden Datenbausteinvariablen und initialisiert die neu hinzugefügten Variablen mit ihren Startwerten.

Um diese Funktion für ein Online-Projekt mit der CPU im Betriebszustand RUN zu aktivieren, gehen Sie wie folgt vor:

1. Öffnen Sie im Ordner der Programmbausteine in der STEP 7-Projektnavigation den Baustein.
2. Klicken Sie im Bausteineditor auf die Schaltfläche "Laden ohne Reinitialisierung", um die Funktion zu aktivieren. (Das Symbol ist im aktivierten Zustand umrahmt: )
3. Bestätigen Sie Ihre Auswahl mit "OK".
4. Fügen Sie der Bausteinschnittstelle Variablen hinzu und laden Sie den Baustein im Betriebszustand RUN. Sie können so viele neue Variablen hinzufügen und laden, wie Ihre Speicherreserve zulässt.

Wenn Sie Ihrem Baustein mehr Bytes hinzugefügt haben, als Sie für die Speicherreserve konfiguriert haben, zeigt STEP 7 einen Fehler an, sobald Sie versuchen, den Baustein im Betriebszustand RUN zu laden. Sie müssen die Bausteineigenschaften bearbeiten und die Speicherreserve vergrößern. Solange die Funktion "Laden ohne Reinitialisierung" aktiviert ist, können Sie keine bestehenden Einträge löschen und die Speicherreserve des Bausteins nicht ändern. Um die Funktion "Laden ohne Reinitialisierung" zu deaktivieren, gehen Sie folgendermaßen vor:

1. Klicken Sie im Bausteineditor auf die Schaltfläche "Laden ohne Reinitialisierung", um die Funktion zu deaktivieren. (Das Symbol ist im deaktivierten Zustand nicht umrahmt: )
2. Bestätigen Sie Ihre Auswahl mit "OK".
3. Laden Sie den Baustein in die CPU. Im Dialogfeld zum Laden müssen Sie "Reinitialisieren" auswählen, um den erweiterten Baustein zu laden.

Beim Laden werden dann alle bestehenden und neuen Bausteinvariablen mit ihren Startwerten reinitialisiert.

Remanente Bausteinvariablen laden

Um remanente Bausteinvariablen im Betriebszustand RUN laden zu können, ist die Zuordnung einer remanenten Speicherreserve erforderlich. Zum Konfigurieren dieser remanenten Speicherreserve gehen Sie folgendermaßen vor:

1. Klicken Sie im Ordner der Programmbausteine in der STEP 7-Projektnavigation mit der rechten Maustaste auf den Baustein und wählen Sie im Kontextmenü "Eigenschaften" aus.
2. Wählen Sie die Eigenschaft "Laden ohne Reinitialisierung" aus.
3. Aktivieren Sie das Kontrollkästchen für "Laden ohne Reinitialisierung für remanente Variablen aktivieren".
4. Konfigurieren Sie die für die remanente Speicherreserve verfügbare Anzahl von Bytes.
5. Bestätigen Sie Ihre Änderungen mit "OK".
6. Fügen Sie dem Datenbaustein remanente Datenbausteinvariablen hinzu und laden Sie den Datenbaustein im Betriebszustand RUN. Sie können so viele neue remanente Datenbausteinvariablen hinzufügen und laden, wie Ihre remanente Speicherreserve zulässt.

Wenn Sie Ihrem Datenbaustein mehr remanente Bytes hinzugefügt haben, als Sie für die Speicherreserve konfiguriert haben, zeigt STEP 7 einen Fehler an, sobald Sie versuchen, den Baustein im Betriebszustand RUN zu laden. Sie können remanente Bausteinvariablen nur im Rahmen der Größe der remanenten Speicherreserve hinzufügen, damit Sie die Variablen im Betriebszustand RUN laden können.

Wenn Sie die erweiterten remanenten Bausteinvariablen laden, enthalten die Variablen ihre aktuellen Werte.

Größe der Speicherreserve für neue Bausteine konfigurieren

Die Standardgröße der Speicherreserve für neue Datenbausteine beträgt 100 Byte. Wenn Sie einen neuen Baustein erstellen, umfasst dieser eine Reserve von 100 Byte. Wenn Sie neue Bausteine mit einer Speicherreserve in einer anderen Größe erstellen möchten, können Sie die Einstellung in den PLC-Programmireinstellungen ändern:

1. Wählen Sie in STEP 7 den Menübefehl **Optionen > Einstellungen** aus.
2. Erweitern Sie im Dialogfeld "Einstellungen" den Bereich "PLC-Programmierung" und wählen Sie "Allgemein".
3. Geben Sie im Abschnitt "Laden ohne Reinitialisierung" die Anzahl von Bytes für die Speicherreserve ein.

Beim Erstellen neuer Bausteine verwendet STEP 7 die von Ihnen für neue Bausteine konfigurierte Speicherreserve.

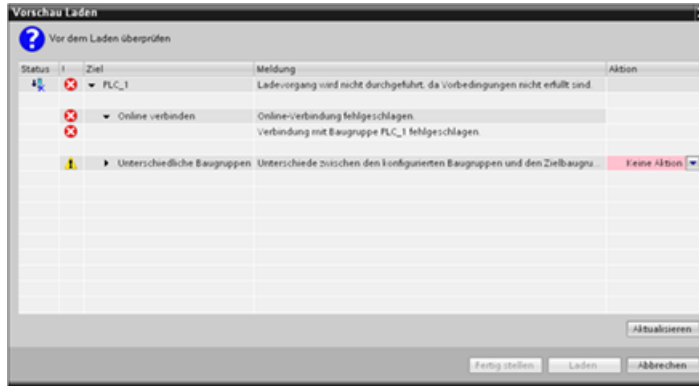
Einschränkungen

Die folgenden Einschränkungen gelten für das Bearbeiten und Laden von Bausteinen im Betriebszustand RUN:

- Das Erweitern der Bausteinschnittstelle und das Hinzufügen neuer Variablen und das Laden im Betriebszustand RUN sind nur bei optimierten Bausteinen (Seite 182) möglich.
- Ohne Reinitialisierung können Sie nicht die Struktur eines Bausteins ändern und den geänderten Baustein im Betriebszustand RUN laden. Für das Hinzufügen neuer Elemente zu einer Struct (Seite 113)-Variable, für das Ändern von Variablennamen, Array-Größen, Datentypen oder des remanenten Zustands ist es erforderlich, den Baustein beim Laden im Betriebszustand RUN zu reinitialisieren. Die einzigen Änderungen an bestehenden Bausteinvariablen, die Sie durchführen und trotzdem den Baustein ohne Reinitialisierung in RUN laden können, sind Änderungen an Startwerten (Datenbausteine), Standardwerten (Funktionsbausteine) oder Kommentaren.
- Sie können maximal so viele neue Bausteinvariablen im Betriebszustand RUN laden, wie die Speicherreserve aufnehmen kann.
- Sie können maximal so viele neue remanente Bausteinvariablen im Betriebszustand RUN laden, wie die remanente Speicherreserve aufnehmen kann.

15.16.6 Systemreaktion bei fehlgeschlagenem Ladevorgang

Tritt während des ersten Ladevorgangs in RUN ein Netzwerkverbindungsfehler auf, zeigt STEP 7 den im Folgenden abgebildeten Dialog "Vorschau laden" an:



15.16.7 Sicherheitsaspekte beim Laden im Betriebszustand RUN

Bevor Sie das Programm im Betriebszustand RUN laden, bedenken Sie die Auswirkungen Ihrer im Betriebszustand RUN vorgenommenen Änderungen auf den Betrieb der CPU in folgenden Fällen:

- Wenn Sie die Steuerungslogik für einen Ausgang gelöscht haben, behält die CPU den letzten Zustand des Ausganges bei, bis die CPU ausgeschaltet oder in den Betriebszustand STOP versetzt wird.
- Wenn Sie einen schnellen Zähler oder eine Impulsausgabe gelöscht haben und eine der beiden Funktionen in Betrieb ist, läuft der schnelle Zähler bzw. die Impulsausgabe bis zum nächsten Ausschalten bzw. bis zum nächsten Übergang in STOP weiter.

- Logik, die durch den Merker des ersten Zyklus aktiviert wird, wird erst nach dem nächsten Einschalten bzw. nach dem nächsten Wechsel von STOP nach RUN ausgeführt. Der Merker des ersten Zyklus wird nur durch den Wechsel in den Betriebszustand RUN gesetzt und wird nicht durch das Laden im Betriebszustand RUN beeinflusst.
- Die aktuellen Werte von Datenbausteinen (DBs) und/oder Variablen können überschrieben werden.

Hinweis

Damit Sie Ihr Programm im Betriebszustand RUN laden können, muss die CPU das Laden im Betriebszustand RUN unterstützen. Das Programm muss fehlerfrei übersetzt werden und die Kommunikation zwischen STEP 7 und der CPU muss fehlerfrei sein.

Sie können die folgenden Änderungen an Programmbausteinen und Variablen vornehmen und sie im Betriebszustand RUN laden:

- Funktionen (FCs), Funktionsbausteine (FBs) und Variablen tabellen erstellen, überschreiben und löschen
- Datenbausteine (DBs) erstellen und löschen; DB-Strukturänderungen können jedoch nicht überschrieben werden. Ausgangswerte des DBs können überschrieben werden. Im Betriebszustand RUN können Sie keinen Webserver-DB (Steuerung oder Fragment) laden.
- Organisationsbausteine (OBs) überschreiben; es können jedoch keine OBs erstellt oder gelöscht werden.

Sie können maximal zwanzig Bausteine gleichzeitig im Betriebszustand RUN laden. Müssen mehr als zwanzig Bausteine geladen werden, ist die CPU in STOP zu setzen.

Sobald Sie einen Ladevorgang gestartet haben, können Sie erst dann weitere Vorgänge in STEP 7 durchführen, wenn der Ladevorgang beendet ist.

Anweisungen, die beim "Laden im Betriebszustand RUN" fehlschlagen können

Die folgenden Anweisungen melden möglicherweise kurzzeitige Fehler, wenn Änderungen aus dem Laden in RUN in der CPU aktiviert werden. Der Fehler tritt auf, wenn die Anweisung eingeleitet wird, während die CPU die Aktivierung der geladenen Änderungen vorbereitet. Während dieses Zeitraums unterbricht die CPU den weiteren Zugriff des Anwenderprogramms auf den Ladespeicher, solange sie noch in Bearbeitung befindliche Zugriffe des Anwenderprogramms auf den Ladespeicher abarbeitet. Dies geschieht, damit geladene Änderungen konsistent aktiviert werden können.

Anweisung	Reaktion, solange die Aktivierung ansteht
DataLogCreate	STATUS = W#16#80C0, ERROR = WAHR
DataLogOpen	STATUS = W#16#80C0, ERROR = WAHR
DataLogWrite	STATUS = W#16#80C0, ERROR = WAHR
DataLogClose	STATUS = W#16#80C0, ERROR = WAHR
DataLogNewFile	STATUS = W#16#80C0, ERROR = WAHR
DataLogClear	STATUS = W#16#80C0, ERROR = WAHR
DataLogDelete	STATUS = W#16#80C0, ERROR = WAHR
READ_DBL	RET_VAL = W#16#82C0
WRIT_DBL	RET_VAL = W#16#82C0
Create_DB	RET_VAL = W#16#80C0

15.17 CPU-Daten bei Auslösebedingungen verfolgen und aufzeichnen

Anweisung	Reaktion, solange die Aktivierung ansteht
Delete_DB	RET_VAL = W#16#80C0
RTM	RET_VAL = 0x80C0

In allen Fällen ist das VKE der Anweisung falsch, wenn der Fehler auftritt. Der Fehler ist temporär. Wenn er auftritt, führen Sie die Anweisung zu einem späteren Zeitpunkt aus.

Hinweis

Versuchen Sie nicht, die Anweisung während der aktuellen Bearbeitung des OBs auszuführen.

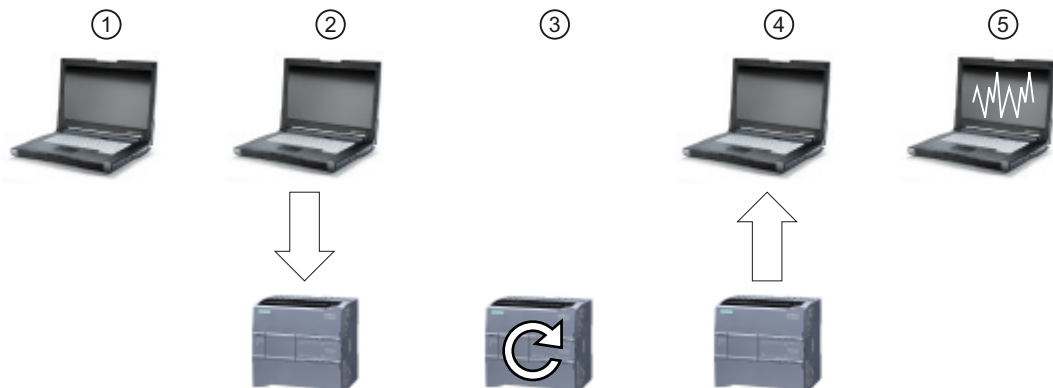
15.17 CPU-Daten bei Auslösebedingungen verfolgen und aufzeichnen

STEP 7 bietet Trace- und Logic-Analyzer-Funktionen, mit denen Sie Variablen für die PLC konfigurieren können, die Sie verfolgen und aufzeichnen wollen. Die aufgezeichneten Daten der Trace-Messungen können Sie dann in Ihr Programmiergerät laden und mit den STEP 7-Tools analysieren, verwalten und grafisch aufbereiten. In Ordner "Traces" in der STEP 7-Projektnavigation können Sie Traces anlegen und verwalten.

Hinweis

Die Daten der Trace-Messungen sind nur im STEP 7-Projekt verfügbar und stehen nicht für die Verarbeitung durch andere Werkzeuge zur Verfügung.

Die folgende Abbildung zeigt den Ablauf der Trace-Funktion:



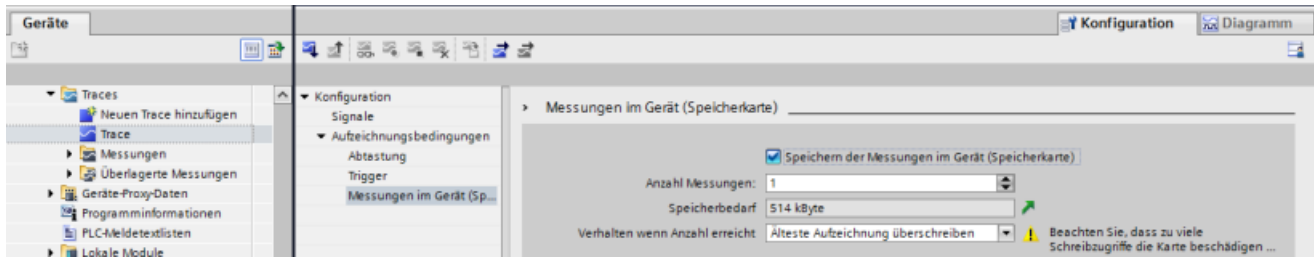
- ① Richten Sie die Trace-Funktion im Trace-Editor von STEP 7 ein. Sie können die folgenden Optionen festlegen:
 - Aufzuzeichnende Datenwerte
 - Erfassungsdauer
 - Erfassungsfrequenz
 - Auslösebedingung
- ② Übertragen Sie die Trace-Konfiguration von STEP 7 in die PLC.
- ③ Die PLC führt das Programm aus und startet die Aufzeichnung der Trace-Daten, sobald die Auslösebedingung eintritt.
- ④ Übertragen Sie die aufgezeichneten Werte von der PLC in STEP 7.
- ⑤ Verwenden Sie die Werkzeuge in STEP 7, um die Daten auszuwerten, grafisch darzustellen und zu sichern.

Die S7-1200 unterstützt zwei Trace-Aufträge mit pro Trigger-Ereignis maximal 16 erfassten Variablen. Jeder Trace-Auftrag stellt 524.288 Byte an RAM-Speicher für die Aufzeichnung der Trace-Werte und die mit diesen Werten in Zusammenhang stehenden Verwaltungswerte bereit, zum Beispiel Variablenadressen und Zeitstempel.

Speichern von Trace-Messungen auf der Memory Card

Die S7-1200 CPU kann Trace-Messungen nur auf der SIMATIC Memory Card speichern. Wenn in Ihrer CPU keine Memory Card vorhanden ist, erzeugt die CPU einen Eintrag im Diagnosepuffer, wenn das Programm versucht, Trace-Messungen zu speichern. Die CPU begrenzt den Speicherplatz für Trace-Messungen, so dass immer 1 MB des externen Ladespeichers verfügbar ist. Würde eine Trace-Messung mehr Speicher als maximal zulässig erfordern, speichert die CPU die Messung nicht und erzeugt einen Eintrag im Diagnosepuffer.

Zusätzlich gilt, dass sich bei Auswahl von "Älteste Aufzeichnung überschreiben" in STEP 7 die Lebensdauer des Ladespeichers durch das ständige Beschreiben des Datenträgers verringern kann. Bei Auswahl von "Älteste Aufzeichnung überschreiben" ersetzt die CPU die älteste Messung durch die neueste Messung, nachdem sie die konfigurierte Anzahl von Trace-Messungen gespeichert hat. Anschließend werden weitere Trace-Messungen erfasst und gespeichert. Überschreiben der ältesten Messungen ist hilfreich bei der Diagnose sporadischer Probleme.



Die CPU unterstützt maximal 999 Trace-Messergebnisse. Während der Zeit, in der die CPU die Trace-Messungen im externen Ladespeicher speichert, prüft die CPU die Triggerbedingung für den Trace-Auftrag nicht. Sobald die CPU die Trace-Messungen gespeichert hat, prüft Sie wieder auf Triggerbedingungen.

Zugriff auf Beispiele

Einzelheiten zur Trace-Programmierung, zum Laden der Konfiguration, Hochladen der Trace-Daten und Darstellung der Daten im Logic Analyzer finden Sie im Informationssystem von STEP 7. Kapitel "Online- und Diagnosefunktionen nutzen" > "Trace- und Logikanalysatorfunktion nutzen" enthält ausführliche Beispiele.

Außerdem ist das Online-Handbuch "Industry Automation SINAMICS/SIMATIC Trace- und Logikanalysatorfunktion nutzen" (<https://support.industry.siemens.com/cs/ww/de/view/64897128>) eine ausgezeichnete Hilfe.

15.18 Ermitteln der Art eines Drahtbruchs über ein Modul SM 1231

Wie unter Messbereiche der analogen Eingänge für Spannung und Strom (SB und SM) (Seite 1358) beschrieben, gibt das Modul SM 1231 den Analogeingangswert 32767 (16#7FFF) sowohl bei einem Drahtbruch als auch bei einem Überlauf aus. Wenn Sie ermitteln möchten, welcher dieser beiden Zustände aufgetreten ist, können Sie zur Ermittlung Logik in Ihr STEP 7-Programm aufnehmen. Die Methode zum Ermitteln des Zustands besteht aus den folgenden Aufgaben:

- Erstellen Sie einen Diagnosefehler-OB, der dann aufgerufen wird, wenn ein kommendes oder gehendes Diagnoseereignis vorhanden ist.
- Nehmen Sie einen Aufruf der Anweisung RALRM auf.
- Richten Sie ein Array aus Bytes für den Parameter AINFO ein, der die Informationen über den Zustand enthält.
- Werten Sie die Bytes 32 und 33 der AINFO-Struktur von RALRM_DB aus, wenn die CPU den Diagnosealarm-OB auslöst.

Erstellen eines Diagnosefehler-OBs

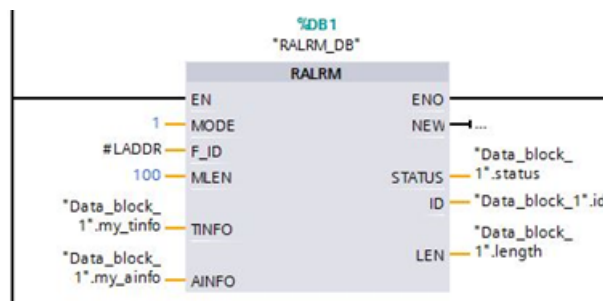
Um ermitteln zu können, wann ein Drahtbruch auftritt, erstellen Sie einen Diagnosefehler-OB. Die CPU ruft diesen OB immer dann auf, wenn ein kommendes oder gehendes Diagnoseereignis auftritt.

Wenn die CPU den Diagnosefehler-OB aufruft, enthält der Eingangsparameter LADDR die Hardwarekennung des Moduls mit dem Fehler. Sie finden die Hardwarekennung des Moduls SM 1231 in der STEP 7-Gerätekonfiguration des Moduls SM 1231.

Aufrufen der Anweisung RALRM

Um den Aufruf der Anweisung RALRM zu programmieren, gehen Sie wie folgt vor:

1. Fügen Sie einen Aufruf von RALRM in Ihr STEP 7-Programm ein.
2. Richten Sie für den Eingangsparameter F_ID die Hardwarekennung des Parameters LADDR des Diagnosefehler-OBs ein.
3. Verwenden Sie ein Array aus Bytes für die Eingangsparameter TINFO und AINFO. Verwenden Sie eine Array-Größe von mindestens 34 Bytes.



Auswerten von AINFO nach Auftreten eines Diagnosealarms

Nach Ausführung des Fehleralarm-OBs enthält das Byte-Array AINFO die Informationen über die Moduldiagnose.

Bytes 0 bis 25 sind Kopfinformationen. Die folgenden Bytes gehören zur Moduldiagnose:

Byte	Beschreibung
26 und 27	Wortwert 16#8000 - weist darauf hin, dass es sich um eine Diagnose im PROFINET-Stil handelt
28 und 29	Wort, das die für diese Diagnose verantwortliche Kanalnummer enthält
30	Bitmuster aaabb000 gibt die Art des Kanals (aaa) und die Art des Fehlers (bb) an
	aaa
	bb
	000: Reserviert
	001: Eingangskanal
	010: Ausgangskanal
	011: Eingangs-/Ausgangskanal
	00: Reserviert
	01: kommender Fehler
	10: gehender Fehler
	11: gehender Fehler, weitere Fehler sind vorhanden

15.18 Ermitteln der Art eines Drahtbruchs über ein Modul SM 1231

Byte	Beschreibung
31	Anzeige des Datenformats 0: Freies Datenformat 1: Bit 2: Zwei Bits 3: Vier Bits 4: Byte 5: Wort (zwei Bytes) 6: Doppelwort (vier Bytes) 7: Zwei Doppelwörter (acht Bytes)
32 und 33	Wort, das die Art des Fehlers definiert: 16#0000 Reserviert 16#0001: Kurzschluss 16#0002: Unterspannung 16#0003: Überspannung 16#0004: Überlast 16#0005: Übertemperatur 16#0006: Drahtbruch 16#0007: oberer Grenzwert überschritten 16#0008: unterer Grenzwert überschritten 16#0009: Fehler

Betrachten Sie zum Beispiel die Bytes 26 bis 33 dieser AINFO-Struktur:

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

- Das Wort in den Bytes 26 und 27 ist 16#8000, was darauf hinweist, dass es sich um eine Diagnose im PROFINET-Stil handelt.
- Das Wort in den Bytes 28 und 29 zeigt an, dass diese Diagnose für Kanal 0 bzw. das Modul gilt.
- Byte 30 ist 16#28, was bei Auswertung als Bitmuster aaa bb 00 = 001 01 000 ist. Dieser Wert deutet an, dass diese Diagnose für einen Eingangskanal gilt und es sich um einen kommenden Fehler handelt.
- Byte 31 ist 5, was auf einen Wortwert hinweist.
- Der Wortwert in den Bytes 32 und 33 ist 16#0007, was auf ein Überschreiten des oberen Grenzwerts deutet.

Durch Erfassen der AINFO-Informationen eines Diagnosefehleralarmereignisses können Sie auf diese Weise die Ursache des Diagnoseereignisses ermitteln.

15.19 Sichern und Wiederherstellen einer CPU

15.19.1 Optionen zum Sichern und Wiederherstellen

Im Lauf der Zeit nehmen Sie an Ihrem Automatisierungssystem eine Vielzahl von Änderungen vor, so fügen Sie beispielsweise neue Geräte hinzu, ersetzen vorhandene Geräte oder passen das Anwenderprogramm an. Sollten diese Änderungen zu unerwünschtem Verhalten führen, können Sie für die Automatisierungsanlage eine frühere Version wiederherstellen, sofern Sie eine Sicherungskopie angelegt haben. STEP 7 und die S7-1200 CPU bieten verschiedene Optionen zum Sichern und Wiederherstellen der Hardwarekonfiguration und der Software.

Optionen zum Sichern

Die folgende Tabelle bietet einen Überblick über die Optionen zum Sichern und Wiederherstellen von S7-CPU:

	Momentaufnahme der Beobachtungswerte	Aus Gerät laden (Software)	Gerät als neue Station laden (Hardware und Software)	Sicherungskopie aus Online-Gerät laden
Anwendungsfall	Einen spezifischen Zustand eines Datenbausteins wiederherstellen. Die Istwerte von Datenbausteinen einschließlich Zeitstempel werden im Projekt angenommen.	Bausteine aus einer CPU ins Projekt laden.	Hardwarekonfiguration und Software von einem Gerät ins Projekt laden.	Eine vollständige Sicherungskopie einer CPU als Wiederherstellungspunkt erstellen. Die Sicherungskopie ist konsistent und kann nicht geändert oder geöffnet werden.
Voraussetzung	Die CPU ist in einem Projekt vorhanden. Die Datenbausteine müssen online und offline identisch sein.	Die CPU ist im Projekt vorhanden.	Das Gerät ist im Hardwarekatalog des TIA Portals vorhanden. Alle erforderlichen HSPs oder GSD-Dateien sind installiert.	-
Möglich im Betriebszustand	RUN, STOP	RUN, STOP	RUN, STOP	STOP
Möglich für F-CPU	Ja	Ja	Nein	Ja
Sicherungskopie kann bearbeitet werden	Ja	Ja	Ja	Nein

Inhalte der Sicherungskopie

Die folgende Tabelle zeigt, welche Daten Sie mit welchen Optionen laden und sichern können:

	Momentaufnahme der Beobachtungswerte	Aus Gerät laden (Software)	Gerät als neue Station laden (Hardware und Software)	Sicherungskopie aus Online-Gerät laden
Istwerte der Datenbausteine	Momentaufnahme ist möglich	Laden ist möglich	Laden ist möglich	Sichern ist möglich
Softwarebausteine	-	Laden ist möglich	Laden ist möglich	Sichern ist möglich
PLC-Variablen (Namen von Variablen und Konstanten)	-	Laden ist möglich	Laden ist möglich	Sichern ist möglich
Technologieobjekte	-	Laden ist möglich	Laden ist möglich	Sichern ist möglich
Hardwarekonfiguration	-	-	Laden ist möglich	Sichern ist möglich
Beobachtungstabellen (Webserver)	-	-	Laden ist nicht möglich	Sichern ist möglich
Lokale Daten, Merker, Zeiten, Zähler und Prozessabbild	Momentaufnahme ist nicht möglich	Laden ist nicht möglich	Laden ist nicht möglich	Sichern ist möglich
Archive und Rezepte (PLC)	-	-	-	Sichern ist möglich
Allgemeine Daten auf der SIMATIC Memory Card, zum Beispiel Hilfe für Programmbausteine oder GSD-Dateien	-	-	-	Sichern ist möglich

Besondere Hinweise zum Sichern von Istwerten

Die Sicherung vom Typ "Sicherung von Online-Gerät laden" sichert die Istwerte der Variablen, die als remanent definiert sind. Um die Konsistenz der remanenten Daten zu gewährleisten, deaktivieren Sie während des Sicherungsvorgangs alle Schreibzugriffe auf remanente Daten.

Ein Betriebszustandswechsel von STOP in RUN setzt die Istwerte von nichtremanenten Daten auf ihre Startwerte. Eine CPU-Sicherungskopie enthält nur die Startwerte von nichtremanenten Daten.

15.19.2 Sichern einer Online-CPU

Eine Sicherungskopie Ihrer Konfiguration anzufertigen, kann sinnvoll sein, wenn Sie zu einem späteren Zeitpunkt möglicherweise zu einer bestimmten Konfiguration zurückkehren möchten. Sie können die aktuelle Konfiguration dann wiederherstellen.

Voraussetzung

Sie können so viele Sicherungskopien anlegen, wie Sie möchten, und eine Vielzahl von Konfigurationen für eine CPU speichern. Um eine Sicherungskopie anzulegen, müssen die folgenden Voraussetzungen erfüllt sein:

- Sie haben die CPU im STEP 7-Projekt bereits angelegt.
- Sie haben die CPU über die PROFINET-Schnittstelle der CPU direkt mit dem Programmiergerät/ PC verbunden. Die Sicherungs- und Wiederherstellungsvorgänge unterstützen die PROFIBUS-Schnittstellen der CMs nicht.
- Die CPU ist online. (Wenn keine Online-Verbindung besteht, wird vom Sicherungsvorgang eine Online-Verbindung hergestellt.)
- Die CPU befindet sich im Betriebszustand STOP. (Wenn sich die CPU nicht im Betriebszustand STOP befindet, werden Sie vom Sicherungsvorgang aufgefordert, zu bestätigen, dass die CPU in den Betriebszustand STOP versetzt werden darf.)

Vorgehen

Um von der aktuellen Konfiguration einer CPU eine Sicherungskopie anzulegen, gehen Sie folgendermaßen vor:

1. Wählen Sie die CPU in der Projektnavigation aus.
2. Wählen Sie im Menü "Online" den Befehl "Sicherung von Online-Gerät laden" aus. Sie müssen gegebenenfalls das Passwort für den Lesezugriff auf die CPU eingeben und bestätigen, dass die CPU in den Betriebszustand STOP gehen soll.

Ergebnis

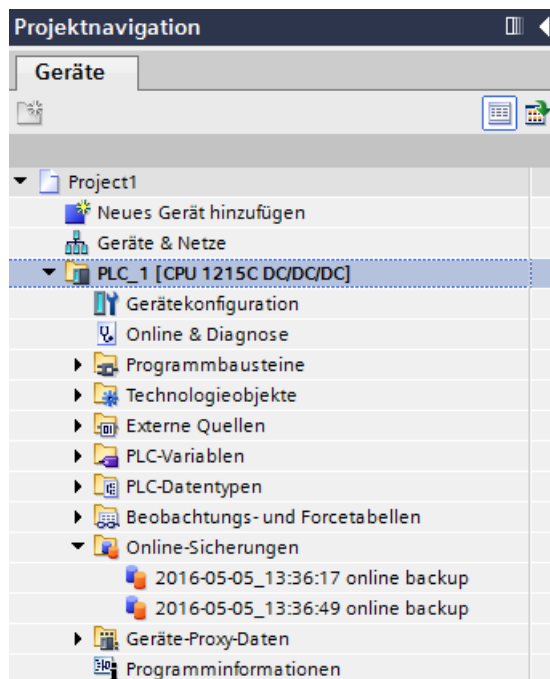
Die Sicherungskopien werden mit dem Namen der CPU sowie der Uhrzeit und dem Datum der Sicherungskopie benannt. Die Sicherungskopie umfasst alle Daten, die erforderlich sind, um eine bestimmte Konfiguration einer CPU wiederherzustellen. Die CPU sichert die folgenden Daten:

- Inhalte der Memory Card, sofern vorhanden
- Remanente Speicherbereiche von Datenbausteinen, Zählern und Merkern
- Andere remanente Speicherinhalte wie IP-Adressparameter

Die Sicherung enthält die Istwerte der CPU, jedoch nicht den Diagnosepuffer.

Die Sicherung enthält nicht das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten.

Sie finden die Sicherungskopie in der Projektnavigation unter der CPU im Ordner "Online-Sicherungen". Die folgende Abbildung zeigt eine S7-1200 CPU, für die zwei Sicherungskopien erstellt wurden:



Hinweis

Beachten Sie, dass Sie die Online-CPU auch über das SIMATIC Automation Tool (SAT) oder die Standard-Webseite "Online-Sicherung" des Webservers (Seite 883) sichern können.

Wenn Sie Dateien über STEP 7 sichern, speichert STEP 7 die Dateien im STEP 7-Projekt. Wenn Sie Dateien über den Webserver sichern, speichert Ihr PC bzw. Gerät die Sicherungsdateien im Standardordner für Downloads. Sie können STEP 7-Sicherungsdateien nicht über den Webserver wiederherstellen, und Sie können Webserver-Sicherungsdateien nicht über STEP 7 wiederherstellen. Sie können jedoch STEP 7-Sicherungsdateien direkt im Download-Ordner Ihres PCs oder Geräts speichern. Wenn Sie auf diese Weise vorgehen, können Sie diese Dateien über den Webserver wiederherstellen.

Speichern von Sicherungsdateien auf Ihrem PC oder Gerät


Um eine Sicherungsdatei auf Ihrem PC oder Gerät zu speichern, gehen Sie folgendermaßen vor:

1. Klicken Sie in der Projektnavigation mit der rechten Maustaste auf eine Datei im Ordner Online-Sicherungen.
2. Wählen Sie im Kontextmenü "Speichern unter".
3. Navigieren Sie in den Ordner, in dem Sie die Datei speichern möchten, z. B. den Standardordner für Downloads auf Ihrem PC oder Gerät.
4. Klicken Sie auf "Speichern".

15.19.3 Wiederherstellen einer CPU

Wenn Sie die Konfiguration einer CPU zu einem früheren Zeitpunkt gesichert haben, können Sie die Sicherungskopie in die CPU übertragen. Die CPU geht beim Wiederherstellen einer Sicherung in den Betriebszustand STOP. Wenn für die CPU eine Zugriffsstufe konfiguriert ist, müssen Sie das Passwort für Lesezugriff auf die CPU eingeben.

Die Sicherung enthält nicht das Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten.

 WARNUNG
Wiederherstellen von Sicherungskopien mit unbekanntem Inhalt
Wenn Sie eine Sicherungskopie mit unbekanntem Inhalt wiederherstellen, kann dies bei Fehlfunktionen oder Programmfehlern zu schweren Sachschäden oder Verletzungen führen. Achten Sie darauf, dass die Sicherungskopie eine Konfiguration mit bekanntem Inhalt enthält.

Wiederherstellen einer Sicherung auf einer CPU mit Schutz vertraulicher PLC-Konfigurationsdaten

Wenn Ihre CPU über den Schutz vertraulicher PLC-Konfigurationsdaten (Seite 161) verfügt, müssen Sie sicherstellen, dass das konfigurierte Passwort in der Sicherungsdatei für den Schutz vertraulicher PLC-Konfigurationsdaten mit dem in der CPU übereinstimmt.

Wenn die Passwörter nicht übereinstimmen, kann die CPU nicht in den Betriebszustand RUN wechseln.

Wenn Sie versuchen, eine Sicherungsdatei wiederherzustellen, die ein anderes Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten hat als die CPU, schlägt die Wiederherstellung fehl. Die CPU wird jedoch im Fehlerzustand neu gestartet, weil der Schutz vertraulicher PLC-Konfigurationsdaten in der CPU nicht mit dem des Projekts übereinstimmte, das Sie in der CPU wiederhergestellt haben.

In diesem Fall müssen Sie den Schutz vertraulicher PLC-Konfigurationsdaten in der CPU so einstellen, dass er mit dem im wiederhergestellten Projekt übereinstimmt. Sie können das Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten in der CPU hier festlegen oder löschen:

- SIMATIC Automation Tool ab V4.0 SP3
- TIA Portal ab V17, Online & Diagnose (Seite 1213)
- SIMATIC Memory Card (Seite 129)

Voraussetzung

Um eine Sicherungskopie wiederherzustellen, müssen die folgenden Voraussetzungen erfüllt sein:

- Das STEP 7-Projekt umfasst eine Konfiguration für die CPU und eine zuvor angelegte Sicherungskopie.
- Die CPU ist über die PROFINET-Schnittstelle der CPU direkt mit dem Programmiergerät verbunden.

- Die CPU befindet sich im Betriebszustand STOP.
- Sie kennen das Passwort für vollständigen Zugriff auf die CPU, sofern eine Zugriffsstufe (Seite 163) konfiguriert wurde.

Vorgehen

Um eine Sicherungskopie wiederherzustellen, gehen Sie wie folgt vor:

1. Öffnen Sie die CPU in der Projektnavigation, um die untergeordneten Objekte anzuzeigen.
2. Wählen Sie im Ordner "Online-Sicherungen" die Sicherungskopie aus, die Sie wiederherstellen möchten.
3. Wählen Sie im Menü "Online" den Befehl "Laden in Gerät" aus.
 - Wenn Sie bereits vorher eine Online-Verbindung hergestellt haben (Seite 1209), wird das Dialogfeld "Vorschau Laden" angezeigt. Dieses Dialogfeld zeigt Alarme an und empfiehlt Maßnahmen für den Ladevorgang.
 - Wenn Sie zuvor noch keine Online-Verbindung aufgebaut haben, wird das Dialogfeld "Erweitertes Laden in Gerät" angezeigt, und Sie müssen zunächst die Schnittstelle auswählen, über die Sie die Online-Verbindung mit der CPU herstellen möchten.
4. Prüfen Sie die Alarme im Dialogfeld "Vorschau Laden" und wählen Sie gegebenenfalls die Maßnahmen in der Spalte "Aktion" aus.
5. Klicken Sie auf die Schaltfläche "Laden". (Die Schaltfläche "Laden" kann ausgewählt werden, sobald das Laden möglich ist.)
6. STEP 7 stellt die Sicherung in der CPU wieder her. Im Dialogfeld "Ergebnisse des Ladevorgangs" können Sie prüfen, ob der Ladevorgang erfolgreich war und gegebenenfalls weitere Maßnahmen ergreifen.
7. Nach Überprüfung im Dialogfeld "Ergebnisse des Ladevorgangs" klicken Sie auf die Schaltfläche "Fertigstellen".
Geben Sie, wenn Sie dazu aufgefordert werden, das Passwort für vollständigen Zugriff auf die CPU ein und bestätigen Sie, dass die CPU in den Betriebszustand STOP wechseln soll.
STEP 7 stellt den Inhalt der Sicherungskopie in der CPU wieder her und startet die CPU neu.

Hinweis

Beachten Sie, dass Sie eine CPU-Sicherungskopie auch über die Standard-Webseite "Online-Sicherung" des Webservers (Seite 883) wiederherstellen können.

Technische Daten

A.1 Siemens-Website für Online-Support

Technische Informationen zu diesen Produkten finden Sie auf der Website Siemens Industry Online-Support (<https://support.industry.siemens.com/cs/ww/de/>).

A.2 Allgemeine technische Daten

Erfüllte Normen

Der Aufbau des Automatisierungssystems S7-1200 erfüllt die folgenden Normen und Prüfvorschriften. Die Prüfkriterien für S7-1200 beruhen auf diesen Normen und Prüfvorschriften.

Beachten Sie, dass möglicherweise nicht alle S7-1200 Varianten nach diesen Normen zertifiziert sind und dass sich der Zertifizierungszustand ohne Ankündigung ändern kann. Es liegt in der Verantwortung des Anwenders, geltende Zertifizierungen anhand der auf dem Produkt angebrachten Zulassungen zu ermitteln. Wenden Sie sich an Ihre Siemens-Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

CE-Zulassung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen und sicherheitsrelevanten Ziele der folgenden EU-Richtlinien und entspricht den harmonisierten europäischen Normen (EN) für speicherprogrammierbare Steuerungen, die in den Amtsblättern der EU aufgeführt sind.

- EU-Richtlinie 2006/95/EG (Niederspannungs-Richtlinie) "Elektrische Betriebsmittel für die Verwendung innerhalb bestimmter Spannungsgrenzen"
 - EN 61131-2 Automatisierungssysteme - Anforderungen an die Geräte und Prüfungen
- EU-Richtlinie 2004/108/EG (EMV-Richtlinie) "Elektromagnetische Verträglichkeit"
 - Störaussendung
EN 61000-6:+A1: Industriebereich
 - Funkentstörung
EN 61000-6-2: Industriebereich
- EG-Richtlinie 94/9/EG (ATEX) "Geräte und Schutzsysteme zur bestimmungsgemäßen Verwendung in explosionsgefährdeten Bereichen"
 - EN 60079-0:+A11
 - EN 60079-15: Schutzart 'n':

Die CE-Konformitätserklärung steht allen zuständigen Behörden zur Verfügung bei der:

Siemens AG
Digital Industries
Factory Automation
DI FA AS SYS
Postfach 1963
D-92209 Amberg
Deutschland

cULus-Zulassung



Underwriters Laboratories, Inc. erfüllt:

- Underwriters Laboratories, Inc.: UL 508 Listed (Industriesteuerungsgeräte)
- Canadian Standards Association: CSA C22.2 Nummer 142 (Prozesssteuerungsgeräte)

Hinweis

Die Produktreihe SIMATIC S7-1200 entspricht der CSA-Norm.

Das cULus-Zeichen zeigt an, dass die S7-1200 von Underwriters Laboratories (UL) nach den Normen UL 508 und CSA 22.2 Nr. 142 geprüft und zugelassen wurde.

FM-Zertifizierung



FM-Zulassungen

Zertifizierungsnorm Klasse Nummer 3600, 3611 (ANSI/UL 121201), 3810 (ANSI/UL 61010-1), CSA-Norm C22.2 No. 0-10, C22.2 No. 213, C22.2 No. 61010-1

Zugelassen für den Einsatz in:

Class I, Division 2, Gas Group A, B, C, D, Temperature Class T3C Ta = 60 °C [CA, US]

Class I, Zone 2, Group IIC, Temperature Class T3 Ta = 60 °C [US]

Canadian Class I, Zone 2 Installation nach CEC 18-150 [CA]

WICHTIGE AUSNAHME: Die Anzahl von Eingängen und Ausgängen, die gleichzeitig eingeschaltet sein dürfen, finden Sie in den technischen Daten. Einige Modelle sind auf Ta = 60 °C herabgesetzt.



WARNUNG

Der Austausch von Komponenten kann die Eignung für Class I, Division 2 und Zone 2 beeinträchtigen.

Reparatur von Geräten darf nur von einem autorisierten Siemens Service Center durchgeführt werden.

IECEx-Zulassung

IEC 60079-0: Explosionsfähige Atmosphäre – Allgemeine Anforderungen

IEC 60079-15: Elektrische Betriebsmittel für gasexplosionsgefährdete Bereiche
 Schutzart 'nA'
 IEC FMG 14.0012X
 Ex nA IIC T3 Gc

Die IECEX-Zulassung ist möglicherweise mit der FM-Zulassung für Gefahrenbereiche auf dem Produkt angegeben.

Nur Produkte mit IECEX-Kennzeichnung sind zugelassen. Wenden Sie sich an Ihre Siemens-Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

Relaismodelle haben keine IECEX-Zulassungen.

Die Temperatúrauslegung finden Sie bei der spezifischen Produktbezeichnung.

Bauen Sie die Module in einem geeigneten Gehäuse mit einer Schutzart von mindestens IP54 nach IEC 60079-15 ein.

ATEX-Zulassung



Die ATEX-Zulassung gilt nur für DC-Varianten. Die ATEX-Zulassung gilt nicht für AC- und Relaisvarianten.

EN 60079-0: Explosionsfähige Atmosphäre - Allgemeine Anforderungen

EN 60079-15: Elektrische Betriebsmittel für gasexplosionsgefährdete Bereiche;
 Schutzart 'nA'
 II 3 G Ex nA IIC T4 oder T3 Gc

Besondere Bedingungen für die sichere Verwendung:

Bauen Sie die Module in einem geeigneten Gehäuse mit einer Schutzart von mindestens IP54 nach EN 60529 oder an einem Standort mit ähnlicher Schutzart ein.

Die angeschlossenen Kabel und Leitungen müssen unter den Nennbedingungen für die tatsächlich gemessene Temperatur ausgelegt sein.

Es sind Vorkehrungen dagegen zu treffen, dass die Bemessungsspannung an den Netzklemmen durch transiente Störgrößen von mehr als 119 V überschritten wird.

Zulassung CCCEX



Nach GB 3836.8 (explosionsfähige Atmosphäre – Teil 8: Geräteschutz durch Zündschutzart "n")

GB 3836.1 (explosionsfähige Atmosphäre – Teil 1: Geräte – Allgemeine Anforderungen)

Ex nA IIC T3 Gc

Spezifische Bedingungen für sichere Verwendung:

- Das Gerät darf nur in einem Bereich mit einem Verschmutzungsgrad von max. 2 nach GB/T 17935.1-2008 verwendet werden.
- Das Gerät muss in ein Gehäuse mit mindestens Schutzart IP54 nach GB 3836.8-2014 eingebaut werden.
- Schutz vor Spannungsspitzen muss mit einem Pegel von maximal 140 % der Spitzenbemessungsspannung an den Einspeiseklemmen des Geräts bereitgestellt werden.

- Mit einem Sternchen (*) gekennzeichnete Module werden bei horizontaler Montage auf 55 °C und bei vertikaler Montage auf 45 °C herabgesetzt, sofern sie für maximale Last konfiguriert sind, und benachbarte Kanäle dürfen nicht eingeschaltet sein. Informationen zum Derating finden Sie im Benutzerhandbuch.
- Weitere Informationen finden Sie in der Anleitung.

Australien und Neuseeland - RCM Mark (Regulatory Compliance Mark)



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Normen nach AS/NZS 61000.6.4 und IEC 610000-6-4 (Klasse A).

Koreanische Zertifizierung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Koreanischen Zertifizierung (KC-Kennzeichen). Es wurde als Gerät der Klasse A eingestuft und ist für industrielle Anwendungen und nicht für die private Nutzung gedacht.

Zulassung für die Eurasische Zollunion (Belarus, Kasachstan, Russische Föderation)



EAC (Eurasische Konformität): Deklaration der Konformität nach den technischen Vorschriften der Zollunion (Technical Regulation of Customs Union, TR CU)

Zulassung für das Seewesen

Die S71200 Produkte werden regelmäßig für die Zulassungen hinsichtlich bestimmter Märkte und Anwendungen bei bestimmten Behörden eingereicht. Wenden Sie sich an Ihre Siemens-Vertretung, wenn Sie eine Liste mit den aktuellen Zulassungen für die einzelnen Bestellnummern benötigen.

Klassifizierungsgesellschaften:

- American Bureau of Shipping (ABS): USA
- Bureau Veritas (BV): Frankreich
- Det Norske Veritas (DNV): Norwegen
- Germanischer Lloyd (GL): Deutsch
- Lloyds Register of Shipping (LRS): England
- Nippon Kaiji Kyokai (ClassNK): Japan
- Korean Register of Shipping: Korea
- China Classification Society (CSS): China

Industrienumgebungen

Das Automatisierungssystem S7-1200 wurde für den Einsatz in Industrienumgebungen entwickelt.

Tabelle A-1 Industrienumgebungen

Anwendungsgebiet	Anforderungen an die Störaussendung	Anforderungen an die Entstörung
Industrie	EN 61000-6-4:2007+A1	EN 61000-6-2:2005

Elektromagnetische Verträglichkeit (EMV)

Die elektromagnetische Verträglichkeit (EMV) eines elektrischen Geräts ist dessen Fähigkeit, in einer elektromagnetischen Umgebung bestimmungsgemäß zu funktionieren und keine elektromagnetischen Störungen auszusenden, die den Betrieb anderer elektrischer Geräte in der Umgebung beeinträchtigen könnten.

Tabelle A-2 Störfestigkeit EN 61000-6-2

Elektromagnetische Verträglichkeit - Entstörung nach EN 61000-6-2	
EN 61000-4-2 Elektrostatische Entladung	8 kV Entladung durch die Luft an allen Oberflächen 6 kV Entladung durch Kontakt mit freiliegenden leitenden Oberflächen
EN 61000-4-3 Prüfung der Störfestigkeit gegen hochfrequente elektromagnetische Felder	80 bis 1000 MHz, 10 V/m, 80 % AM bei 1 kHz 1,4 bis 2,0 GHz, 3 V/m, 80 % AM bei 1 kHz 2,0 bis 2,7 GHz, 1 V/m, 80 % AM bei 1 kHz
EN 61000-4-4 Schnelle transiente Störgröße	2 kV, 5 kHz bei Kopplungsnetz zu AC und DC-Systemspannung 2 kV, 5 kHz bei Kopplungsklemme zu Ein/Ausgängen
EN 61000-4-5 Stoßwellenfestigkeit	AC-Systeme - 2 kV Gleichtakt, 1 kV Gegentakt DC-Systeme - 2 kV Gleichtakt, 1 kV Gegentakt Zu DC-Systemen siehe "Stoßwellenfestigkeit" unten.
EN 61000-4-6 Leitungsgeführte Störungen	150 kHz bis 80 MHz, 10 V effektiv, 80% AM bei 1 kHz
EN 61000-4-11 Spannungseinbrüche	AC-Systeme 0 % für 1 Zyklus, 40 % für 12 Zyklen und 70 % für 30 Zyklen bei 60 Hz

Tabelle A-3 Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4

Elektromagnetische Verträglichkeit - Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4		
Leitungsgeführte Störaussendungen EN 55016, Klasse A, Gruppe 1	0,15 MHz bis 0,5 MHz	<79 dB (µV) Quasi-Spitze; <66 dB (µV) Mittelwert
	0,5 MHz bis 5 MHz	<73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert
	5 MHz bis 30 MHz	<73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert

Elektromagnetische Verträglichkeit - Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4		
Abgestrahlte Störaussendungen EN 55016, Klasse A, Gruppe 1	30 MHz bis 230 MHz	<40 dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m
	230 MHz bis 1 GHz	<47 dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m
	1 GHz bis 3 GHz	< 76dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m

Stoßwellenfestigkeit

Verdrahtungssysteme, die Einkopplungen durch Blitzschlag ausgesetzt sind, müssen mit einem externen Schutz versehen sein. Eine Spezifikation zur Bewertung des Schutzes vor Blitzstoßspannungen ist in der Norm EN 61000-4-5 zu finden. Die Betriebsgrenzen sind in EN 61000-6-2 definiert. Wenn S7-1200-Gleichstrom-CPU's und -Signalmodule Stoßspannungen gemäß dieser Norm ausgesetzt sind, benötigen sie zur Aufrechterhaltung des sicheren Betriebs einen externen Schutz.

Nachstehend sind einige Geräte aufgeführt, die die erforderliche Stoßwellenfestigkeit gewährleisten. Diese Geräte bieten nur dann Schutz, wenn sie gemäß den Herstellerempfehlungen ordnungsgemäß installiert sind. Geräte anderer Hersteller mit gleichen oder besseren technischen Daten können ebenfalls verwendet werden:

Tabelle A-4 Geräte, die Stoßwellenfestigkeit unterstützen

Subsystem	Schutzgerät
Stromversorgung +24 V DC	BLITZDUCTOR VT, BVT AVD 24, Teilenummer 918 422
Industrial Ethernet	DEHNpatch DPA M CLE RJ45B 48, Teilenummer 929 121
RS-485	BLITZDUCTOR XT, Basiseinheit BXT BAS, Teilenummer 920 300
	BLITZDUCTOR XT, Modul BXT ML2 BD HFS 5, Teilenummer 920 271
RS-232	BLITZDUCTOR XT, Basiseinheit BXT BAS, Teilenummer 920 300
	BLITZDUCTOR XT, Modul BXT ML2 BE S 12, Teilenummer 920 222
Digitaleingänge +24 V DC	DEHN, Inc., Type DCO SD2 E 24, Teilenummer 917 988
Digitalausgänge und Geberversorgung +24 V DC	DEHN, Inc., Type DCO SD2 E 24, Teilenummer 917 988
Analoge E/A	DEHN, Inc., Type DCO SD2 E 12, Teilenummer 917 987
Relaisausgänge	Kein Schutz erforderlich

Umgebungsbedingungen

Tabelle A-5 Transport und Lagerung

Umgebungsbedingungen - Transport und Lagerung	
EN 6006822, Test Bb, trockene Wärme und EN 6006821 Test Ab, Kälte	-40 °C bis +70 °C
EN 60068230, Test Db, feuchte Wärme	25 °C bis 55 °C, 95 % Luftfeuchtigkeit
EN 60068-2-14, Test Na, Temperaturschock	-40 °C bis +70 °C, Haltezeit 3 Stunden, 2 Zyklen

Umgebungsbedingungen - Transport und Lagerung	
EN 60068-2-32 Freier Fall	0,3 m, 5 Mal, in Versandverpackung
Atmosphärischer Druck	1140 bis 660 hPa (entspricht einer Höhe von -1000 bis 3500 m)

Tabelle A-6 Klimatische Umgebungsbedingungen

Umgebungsbedingungen - klimatische Umgebungsbedingungen	
Das Automatisierungssystem S7-1200 ist für den Einsatz an witterungsbeständigen festen Standorten geeignet. Die Betriebsbedingungen beruhen auf Anforderungen nach DIN IEC 60721-3-3:	
<ul style="list-style-type: none"> • Klasse 3M3 (mechanische Anforderungen) • Klasse 3K3 (klimatische Anforderungen) 	
Umgebungstemperaturen (Lufteinlass 25 mm unterhalb des Geräts)	-20 °C bis 60 °C C horizontale Montage -20 °C bis 50 °C vertikale Montage 95 % Luftfeuchtigkeit, nicht kondensierend Sofern nicht anders angegeben
Atmosphärischer Druck	1140 bis 795 hPa (entspricht einer Höhe von -1000 bis 2000 m)
Konzentration von Schmutzstoffen	SO ₂ : < 0,5 ppm; H ₂ S: < 0,1 ppm; rel. LF < 60% nicht kondensierend ISA-S71.04 Stärkegrad G1, G2, G3
EN 60068-2-14, Test Nb, Temperaturveränderung	0 °C bis 60 °C
EN 60068227 Mechanische Stoßbeanspruchung	15 G, Impuls 11 ms, 6 Stöße auf jeder der 3 Achsen
EN 6006826 Sinusschwingung	Hutschienenmontage: 3,5 mm von 5-9 Hz, 1 G von 8,4 - 150 Hz Schalttafeleinbau: 7,0 mm von 5 bis 8,4 Hz; 2 G von 8,4 bis 150 Hz 10 Ablenkungen je Achse, 1 Oktave/Minute

Verschmutzungsgrad und Überspannungskategorie nach IEC 61131-2

- Verschmutzungsgrad 2
- Überspannungskategorie: II

Schutzart

- Schutzklasse II nach EN 61131-2 (Schutzleiter nicht erforderlich)

Schutzgrad

- IP20 Mechanischer Schutz, EN 60529
- Schutz gegen direkte Berührung von Hochspannung wie mit genormter Sonde ermittelt. Externer Schutz erforderlich gegen Staub, Schmutz, Wasser und Fremdkörper mit einem Durchmesser von < 12,5 mm.

Bemessungsspannungen

Tabelle A-7 Bemessungsspannungen

Nennspannung	Toleranz
24 V DC	20,4 V DC bis 28,8 V DC
120/230 V AC	85 V AC bis 264 V AC, 47 bis 63 Hz


Hinweis

Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-1200 CPU oder einem digitalen Erweiterungsmodul einschaltet, wird ca. 50 Mikrosekunden lang das Signal "1" an die Digitalausgänge gesendet. Dies kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu lebensgefährlichen Verletzungen und/oder Sachschäden führen kann. Das müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

Verpolschutz

Verpolschutz ist vorhanden bei allen Klemmenpaaren mit +24-V-DC-Spannungsversorgung oder anwenderseitiger Eingangsspannung für CPUs, Signalmodule (SMs) und Signalboards (SBs). Trotzdem sind Beschädigungen des System weiterhin dadurch möglich, dass unterschiedliche Klemmenpaare mit entgegengesetzter Polarität verdrahtet werden.

Einige der 24-V-DC-Eingangsports des S7-1200 Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Beispielsweise sind die folgenden Stromkreise miteinander verbunden, sofern sie in den Datenblättern als "nicht potentialgetrennt" angegeben sind: die 24-V-DC-Versorgung der CPU, die Sensorleistung der CPU, der Leistungseingang für die Relaispule eines SM und die Versorgung eines nicht potentialgetrennten Analogeingangs. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

 WARNUNG
Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betriebsverhalten des Zielsystems und angeschlossener Geräte führen kann.
Die Nichteinhaltung dieser Richtlinien kann Schäden oder unvorhersehbaren Betrieb verursachen, was zu schweren oder lebensgefährlichen Verletzungen und/oder Sachschäden führen kann.
Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem S7-1200 System an dasselbe Bezugspotential an.

DC-Ausgänge

Es stehen keine kurzschlussfesten Schaltungen für die Gleichspannungsausgänge an CPUs, Signalmodulen (SMs) und Signalboards (SBs) zur Verfügung.

Lebensdauer eines Relais

Die typischen Leistungsdaten, die anhand von Beispieltests geschätzt wurden, sind nachstehend aufgeführt. Die tatsächliche Leistungsfähigkeit richtet sich nach der jeweiligen Verwendung. Ein externer Schutzkreis, der der Last angepasst ist, verlängert die Lebensdauer der Kontakte. Öffnerkontakte haben unter induktiver Last oder Lampenlast eine typische Lebensdauer von ungefähr einem Drittel der Lebensdauer eines Schließerkontakts.

Ein externer Schutzkreis erhöht die Lebensdauer der Kontakte.

Tabelle A-8 Typische Leistungsdaten

Daten für die Auswahl eines Aktors				
Thermischer Dauerstrom		max. 2 A		
Schaltvermögen und Lebensdauer der Kontakte				
Bei ohmscher Last	Spannung	Strom	Anzahl der Schaltzyklen (typ.)	
	24 V DC	2,0 A	0,1 Millionen	
	24 V DC	1,0 A	0,2 Millionen	
	24 V DC	0,5 A	1,0 Millionen	
	48 V AC	1,5 A	1,5 Millionen	
	60 V AC	1,5 A	1,5 Millionen	
	120 V AC	2,0 A	1,0 Millionen	
	120 V AC	1,0 A	1,5 Millionen	
	120 V AC	0,5 A	2,0 Millionen	
	230 V AC	2,0 A	1,0 Millionen	
	230 V AC	1,0 A	1,5 Millionen	
	230 V AC	0,5 A	2,0 Millionen	
	Bei induktiver Last (nach IEC 947-5-1 DC13/AC15)	Spannung	Strom	Anzahl der Schaltzyklen (typ.)
24 V DC		2,0 A	0,05 Millionen	
24 V DC		1,0 A	0,1 Millionen	
24 V DC		0,5 A	0,5 Millionen	
24 V AC		1,5 A	1,0 Millionen	
48 V AC		1,5 A	1,0 Millionen	
60 V AC		1,5 A	1,0 Millionen	
120 V AC		2,0 A	0,7 Millionen	
120 V AC		1,0 A	1,0 Millionen	
120 V AC		0,5 A	1,5 Millionen	
230 V AC		2,0 A	0,7 Millionen	
230 V AC		1,0 A	1,0 Millionen	
230 V AC		0,5 A	1,5 Millionen	
Aktivieren eines Digitaleingangs		Möglich		
Schaltfrequenz				

Daten für die Auswahl eines Aktors		
	Mechanisch	Max. 10 Hz
	Bei ohmscher Last	Max. 1 Hz
	Bei induktiver Last (nach IEC 947-5-1 DC13/AC15)	Max. 0,5 Hz
	Bei Lampenlast	Max. 1 Hz

Speicherung im internen CPU-Speicher

- Lebensdauer von remanenten Daten und Datenprotokolldaten: 10 Jahre
- Remanente Daten bei Spannungsausfall, Schreibzyklusbeständigkeit: 2 Millionen Zyklen
- Datenprotokolldaten: Schreibzyklusbeständigkeit: 500 Millionen Datenprotokolleinträge

Hinweis

Auswirkung von Datenprotokollen auf den internen CPU-Speicher

Jeder Schreibvorgang eines Datenprotokolls verbraucht mindestens 2 KB Speicher. Wenn Ihr Programm häufig kleinere Mengen von Daten schreibt, werden bei jedem Schreibvorgang mindestens 2 KB Speicher verbraucht. Eine bessere Umsetzung wäre die Ansammlung kleiner Datenelemente in einem Datenbaustein (DB), der dann weniger häufig ins Datenprotokoll geschrieben würde.


Wenn Ihr Programm sehr häufig viele Datenprotokolleinträge schreibt, ziehen Sie die Verwendung einer austauschbaren SD Memory Card in Betracht.

A.3 Anschlussbelegung PROFINET-Schnittstellenport X1

Die S7-1200 CPU wird an das PROFINET-Netzwerk mit einer Standard-RJ45-Buchse angeschlossen. Die Anschlussbelegung der Buchse hängt vom CPU-Typ ab.

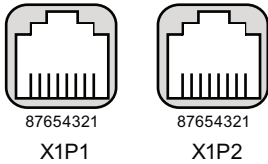
Single-Port-CPU

Single-Port-CPU (CPU 1211C, CPU 1212C und CPU 1214C) besitzen die folgende Standard-Ethernet-MDI-Anschlussbelegung:

Pin	Signalname	Beschreibung	Anschlussbelegung RJ45-Buchse
1	TD+	Daten senden	
2	TD-		
3	RD+	Daten empfangen	
4	GND		
5	GND	Masse	
6	RD-		
7	GND	Masse	
8	GND		

Dual-Port-CPU

Dual-Port-CPU (CPU 1215C und CPU1217C) besitzen die folgende Standard-Ethernet-MDI-X-Anschlussbelegung:

Pin	Signalname	Beschreibung	Anschlussbelegung RJ45-Buchse
1	RD+	Daten empfangen	
2	RD-		
3	TD+	Daten senden	
4	GND	Masse	
5	GND		
6	TD-	Daten senden	
7	GND	Masse	
8	GND		

Hinweis

Dual-Port-CPU

Die unter dem Pin angezeigte Zahl deutet darauf hin, dass es bei Dual-Port-CPU kein Crossover zwischen Pins gibt. Die Geräte verfügen über einen internen Ethernet-Switch: bei den Paaren TD+/- und RD+/- gibt es intern kein Crossover.

Autonegotiation

Wenn die Anschlusskonfiguration Autonegotiation ermöglicht, erkennt die S7-1200 CPU automatisch den Kabeltyp und vertauscht bei Bedarf die Sende-/Empfangsleitungen. Ist Autonegotiation in der Anschlusskonfiguration nicht aktiviert, wird von der CPU auch das automatische Vertauschen deaktiviert. Die Autonegotiation-Einstellung eines Ports wird im TIA Portal im Dialog für die Anschlussoptionen konfiguriert. Dies ist eine portspezifische erweiterte Option für die PROFINET-Schnittstelle (X1) in den CPU-Eigenschaften. Siehe "Konfigurieren des PROFINET-Ports" in Abschnitt 11.2.3.4: "IP-Adresse für eine CPU in Ihrem Projekt konfigurieren" (Seite 608) mit weiteren Informationen.

A.4 CPU 1211C

A.4.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A-9 Allgemeine technische Daten

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Artikelnummer	6ES7211-1BE40-0XB0	6ES7211-1HE40-0XB0	6ES7211-1AE40-0XB0
Abmessungen B x H x T (mm)	90 x 100 x 75		
Versandgewicht	420 Gramm	380 Gramm	370 Gramm

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Leistungsverlust	10 W	8 W	
Verfügbarer elektrischer Strom (CM-Bus)	max. 750 mA (5 V DC)		
Verfügbarer elektrischer Strom (24 V DC)	max. 300 mA (Geberversorgung)		
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang		

Tabelle A-10 CPU-Merkmale

Technische Daten		Beschreibung
Anwenderspeicher (Siehe "Allgemeine technische Daten" (Seite 1253), "Speicherung im internen CPU-Speicher".)	Arbeitsspeicher	50 KB
	Ladespeicher	1 MB, intern, erweiterbar bis auf SD-Kartengröße
	Remanent	14 kByte
Integrierte digitale E/A		6 Eingänge/4 Ausgänge
Integrierte analoge E/A		2 Eingänge
Größe des Prozessabbilds		1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)
Merker (M)		4096 Byte
Temporärer (lokaler) Speicher		<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs) • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs)
Zusätzliche Signalmodule		Keine
Erweiterung SB, CB, BB		max. 1
Zusätzliche Kommunikationsmodule		max. 3 CM
Schnelle Zähler		Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge. Siehe "Anschlussbelegung des Hardwareeingangs" (Seite 568) für CPU 1211C: Zuweisungen der HSC-Standardadressen. 100/180 kHz (Ea.0 bis Ea.5)
Impulsausgänge ²		Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge 100 kHz (Aa.0 bis Aa.3)
Eingänge für Impulsabgriff		6
Verzögerungsalarme		4 mit Auflösung von 1 ms
Weckalarme		4 mit Auflösung von 1 ms
Flankenalarme		6 steigend und 6 fallend (10 und 10 mit optionalem Signalboard)
Memory Card		SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr		+/- 60 Sekunden/Monat
Pufferung Echtzeituhr		Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

¹ Die langsamere Geschwindigkeit gilt, wenn der HSC als A/B-Zähler konfiguriert ist.

² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A-11 Leistung

Art der Anweisung		Ausführungsgeschwindigkeit	
		Direkte Adressierung (E, A und M)	DB-Zugriff
Boolescher Wert		0,08 µs/Anweisung	
Verschieben	Move_Bool	0,3 µs/Anweisung	1,17 µs/Anweisung
	Move_Word	0,137 µs/Anweisung	1,0 µs/Anweisung
	Move_Real	0,72 µs/Anweisung	1,0 µs/Anweisung
Realzahlarithmetik	Add Real	1,48 µs/Anweisung	1,78 µs/Anweisung

Hinweis

Viele Variablen wirken sich auf die gemessenen Zeiten aus. Die oben angegebenen Leistungszeiten gelten für die schnellsten Anweisungen in dieser Kategorie und fehlerfreie Programme.

A.4.2 Von der CPU 1211C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A-12 Von der CPU 1211C unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung
Bausteine	Typ	OB, FB, FC, DB
	Größe	Bis zur Größe des Arbeitsspeichers
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	FB und FC: 1 bis 65535 (z. B. FB 1 bis FB 65535) DB: 1 bis 59999
	Schachteltiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ¹
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.

Element		Beschreibung
OBs	Programmzyklus	Mehrere
	Anlauf	Mehrere
	Verzögerungsalarm	4 (1 pro Ereignis)
	Weckalarme	4 (1 pro Ereignis)
	Prozessalarme	50 (1 pro Ereignis)
	Zeitfehleralarme	1
	Diagnosefehleralarme	1
	Ziehen oder Stecken von Modulen	1
	Fehler bei Baugruppenträger oder Station	1
	Uhrzeitalarm	Mehrere
	Status	1
	Update	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Byte pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Byte • Int, UInt: 6 Byte • DInt, UDInt: 12 Byte

¹ Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

Tabelle A-13 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	1
Typ	Ethernet
HMI-Gerät	4
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> • 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV • 8 CPU/CPU-Verbindungen (Client oder Server) für GET/PUT-Daten • 6 Verbindungen für dynamische Zuordnung zu GET/PUT oder offener Benutzerkommunikation • Max. 64 Verbindungen für Sicherheitszertifikate

Technische Daten	Beschreibung
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu Logik)	Wandler potentialgetrennt, 1500 V AC (Typprüfung) ¹
Kabelart	CAT5e geschirmt
Schnittstellen	
Anzahl PROFINET-Schnittstellen	1
Anzahl PROFIBUS-Schnittstellen	0
Schnittstelle	
Hardware für Schnittstelle	
Anzahl Ports	1
Integrierter Switch	Nein
RJ-45 (Ethernet)	Ja; X1
Protokolle	
PROFINET IO-Controller	Ja
PROFINET IO-Device	Ja
SIMATIC-Kommunikation	Ja
Offene IE-Kommunikation	Ja
Webserver	Ja
Medienredundanz	Nein
PROFINET IO-Controller	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein
Offene IE-Kommunikation	Ja
IRT	Nein
MRP	Nein
PROFenergy	Ja. Die S7-1200 CPU unterstützt nur die PROFenergy-Einheit (mit I-Device-Funktionalität).
Priorisierter Anlauf	Ja (max. 16 PROFINET-Geräte)
Max. Anzahl anschließbarer IO-Devices	16
Max. Anzahl anschließbarer IO-Devices für RT	16
Max. Anzahl davon in Reihe	16
Max. Anzahl gleichzeitig aktivierbarer/deaktivierbarer IO-Devices	8
Aktualisierungszeiten	Der Mindestwert der Aktualisierungszeit ist auch abhängig von der für PROFINET IO eingestellten Kommunikationskomponente, der Anzahl von IO-Devices und der Menge der konfigurierten Anwenderdaten.
Mit RT	
Sendetakt von 1 ms	1 ms bis 512 ms
PROFINET IO-Device	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein

Technische Daten		Beschreibung
Offene IE-Kommunikation		Ja
IRT, unterstützt		Nein
MRP, unterstützt		Nein
PROFlenergy		Ja
Shared Device		Ja
Max. Anzahl IO-Controller mit Shared Device		2
SIMATIC-Kommunikation		
S7-Kommunikation, als Server		Ja
S7-Kommunikation, als Client		Ja
Max. Anwenderdaten pro Auftrag		Siehe Online-Hilfe (S7-Kommunikation, Umfang der Anwenderdaten)
Offene IE-Kommunikation		
TCP/IP:		Ja
	Max. Länge der Daten	8 KB
	Unterstützung für mehrere passive Anschlüsse pro Port	Ja
ISO-on-TCP (RFC1006):		Ja
	Max. Länge der Daten	8 KB
UDP		Ja
	Max. Länge der Daten	1472 Byte
DHCP		Nein
SNMP		Ja
DCP		Ja
LLDP		Ja

¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr aufgrund von gefährlichen Spannungen bei kurzfristigen Netzwerkstörungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

Tabelle A-14 Stromversorgung

Technische Daten		CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Spannungsbereich		85 bis 264 V AC	20,4 V DC bis 28,8 V DC	
Netzfrequenz		47 bis 63 Hz	--	
Eingangsstrom	nur CPU bei max. Last	60 mA bei 120 V AC 30 mA bei 240 V AC	300 mA bei 24 V DC	300 mA bei 24 V DC
	CPU mit allen Erweiterungsbaugruppen bei max. Last	180 mA bei 120 V AC 90 mA bei 240 V AC	900 mA bei 24 V DC	
Einschaltstrom (max.)		20 A bei 264 V AC	12 A bei 28,8 V DC	
I ² t		0,8 A ² s	0,5 A ² s	
Elektrische Trennung (Eingangsleistung zu Logik)		1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funktionserde		max. 0,5 mA	--	

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Verzögerungszeit (Spannungsverlust)	20 mA bei 120 V AC 80 mA bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender austauschbar	3 A, 250 V, träge		

Tabelle A-15 Gebersversorgung

Technische Daten	CPU 1211C AC/DC/Relais	CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L+ minus 4 V DC (min.)	
Nennausgangsstrom (max.)	300 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Gebersversorgung)	Nicht elektrisch getrennt		

A.4.3 Digitale Eingänge und Ausgänge

Tabelle A-16 Digitale Eingänge

Technische Daten	CPU 1211C AC/DC/Relais, CPU 1211C DC/DC/Relais und CPU 1211C DC/DC/DC
Anzahl der Eingänge	6
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, nominal
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Filterzeiten	µs-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
HSC-Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	100/80 kHz (Ea.0 bis Ea.5)
Anzahl gleichzeitig eingeschalteter Eingänge	6 bei 60 °C horizontal, 50 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge

Tabelle A-17 Digitale Ausgänge

Technische Daten	CPU 1211C AC/DC/Relais und CPU 1211C DC/DC/Relais	CPU 1211C DC/DC/DC
Ausgänge	4	
Typ	Relais, mechanisch	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1	
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Maximale Schaltfrequenz Relais	1 Hz	--
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Frequenz Impulsgenerator	Nicht empfehlenswert ¹	100 kHz (Aa.0 bis Aa.3) ² , min. 2 Hz
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)	
Parallele Ausgänge für erhöhte Last	Nein	
Anzahl gleichzeitig eingeschalteter Ausgänge	4 bei 60 °C horizontal, 50 °C vertikal	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

- ¹ Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.
- ² Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

A.4.4 Analoge Eingänge

Tabelle A-18 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32512 bis 32767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 1271).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

A.4.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A-19 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
Abtastzeit	4,17 ms	5 ms	25 ms

A.4.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A-20 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.4.4.3 Messbereiche der analogen Eingänge für Spannung (CPUs)

Tabelle A-21 Darstellung Analogeingang für Spannung (CPUs)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,852 V	Überlauf
32512	7F00		
32511	7EFF	11,759 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Negative Werte		Negative Werte werden nicht unterstützt	

A.4.5 Schaltpläne der CPU 1211C

Tabelle A-22 CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)

① 24-V-DC-Geberversorgung
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.

② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).
Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

Hinweis 2: Eine der beiden Klemmen L1 oder N (L2) kann an eine Spannungsquelle bis 240 V AC angeschlossen werden. Die Klemme N kann als L2 betrachtet werden und muss nicht geerdet werden. Für die Klemmen L1 und N (L2) ist kein Verpolungsschutz erforderlich.

Hinweis 3: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

Tabelle A-23 Anschlussbelegung für die CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Geberversorgung 24 V DC	--	DO a.2
5	M / Geberversorgung 24 V DC	--	DO a.3
6	1M	--	Kein Anschluss
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--

Pin	X10	X11 (vergoldet)	X12
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

Tabelle A-24 CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)

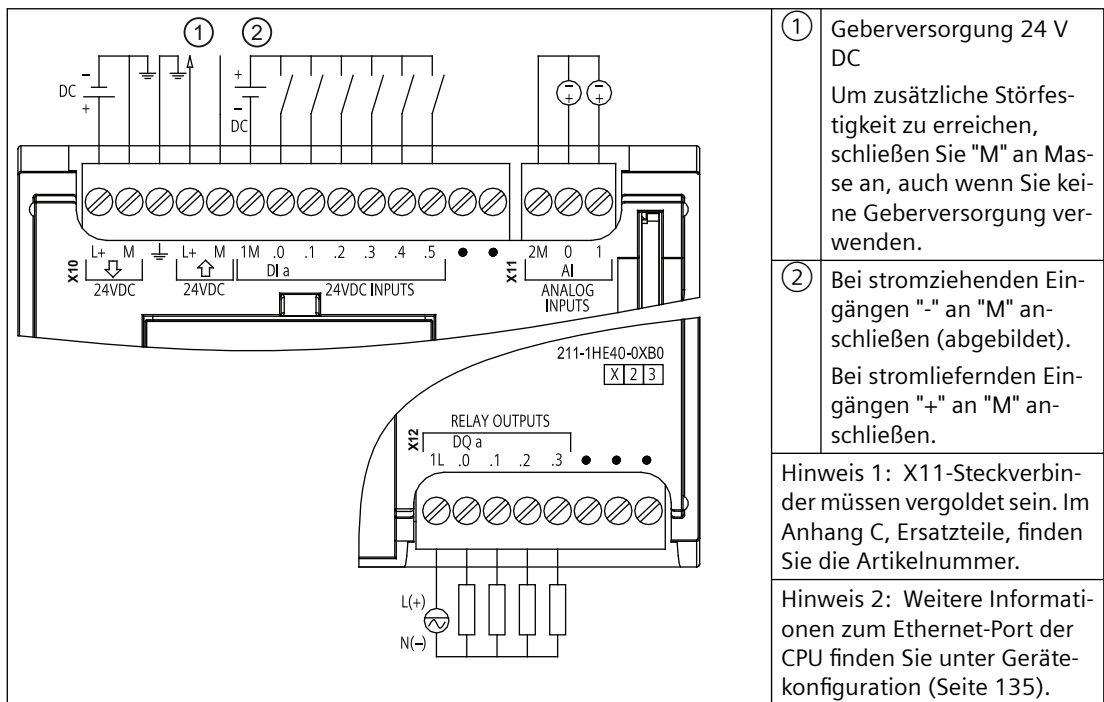


Tabelle A-25 Anschlussbelegung für die CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Geberversorgung 24 V DC	--	DO a.2
5	M / Geberversorgung 24 V DC	--	DO a.3
6	1M	--	Kein Anschluss
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--
10	DI a.3	--	--

Pin	X10	X11 (vergoldet)	X12
11	DI a.4	--	--
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

Tabelle A-26 CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

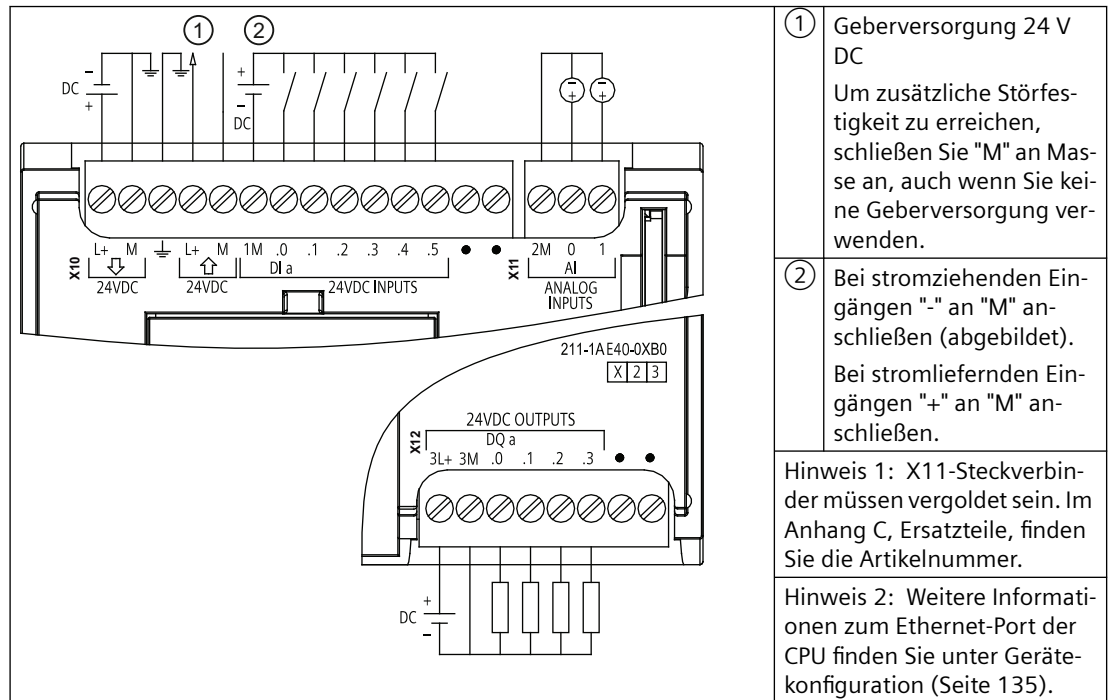


Tabelle A-27 Anschlussbelegung für die CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / Gebersversorgung 24 V DC	--	DO a.1
5	M / Gebersversorgung 24 V DC	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	Kein Anschluss
8	DI a.1	--	Kein Anschluss
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--

Pin	X10	X11 (vergoldet)	X12
12	DI a.5	--	--
13	Kein Anschluss	--	--
14	Kein Anschluss	--	--

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

A.5 CPU 1212C

A.5.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A-28 Allgemein

Technische Daten	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Artikelnummer	6ES7212-1BE40-0XB0	6ES7212-1HE40-0XB0	6ES7212-1AE40-0XB0
Abmessungen B x H x T (mm)	90 x 100 x 75		
Versandgewicht	425 Gramm	385 Gramm	370 Gramm
Leistungsverlust	11 W	9 W	
Verfügbare elektrischer Strom (SM- und CM-Bus)	max 1000 mA (5 V DC)		
Verfügbare elektrischer Strom (24 V DC)	max. 300 mA (Geberversorgung)		
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang		

Tabelle A-29 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeicher (Siehe "Allgemeine technische Daten" (Seite 1253), "Spei- cherung im inter- nen CPU-Spei- cher".)	Arbeitsspeicher	75 KB
	Ladespeicher	2 MB intern, erweiterbar bis auf SD-Kartengröße
	Remanent	14 kByte
Integrierte digitale E/A	8 Eingänge/6 Ausgänge	
Integrierte analoge E/A	2 Eingänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	
Merker (M)	4096 Byte	

Technische Daten	Beschreibung
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs) • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs)
Zusätzliche Signalmodule	max. 2 SMs
Erweiterung SB, CB, BB	max. 1
Zusätzliche Kommunikationsmodule	max. 3 CMs
Schnelle Zähler	<p>Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge. Siehe "Anschlussbelegung des Hardwareeingangs" (Seite 568) für CPU 1212C: Zuweisungen der HSC-Standardadressen.</p> <ul style="list-style-type: none"> • 100/180 kHz (Ea.0 bis Ea.5) • 30/120 kHz (Ea.6 bis Ea.7)
Impulsausgänge ²	<p>Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge</p> <ul style="list-style-type: none"> • 100 kHz (Aa.0 bis Aa.3) • 20 kHz (Aa.4 bis Aa.5)
Eingänge für Impulsabgriff	8
Verzögerungsalarme	4 gesamt mit Auflösung von 1 ms
Weckalarme	4 gesamt mit Auflösung von 1 ms
Flankenalarme	8 steigend und 8 fallend (12 und 12 mit optionalem Signalboard)
Memory Card	SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

¹ Die langsamere Geschwindigkeit gilt, wenn der HSC als A/B-Zähler konfiguriert ist.

² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A-30 Leistung

Art der Anweisung		Ausführungsgeschwindigkeit	
		Direkte Adressierung (E, A und M)	DB-Zugriff
Boolescher Wert		0,08 µs/Anweisung	
Verschieben	Move_Bool	0,3 µs/Anweisung	1,17 µs/Anweisung
	Move_Word	0,137 µs/Anweisung	1,0 µs/Anweisung
	Move_Real	0,72 µs/Anweisung	1,0 µs/Anweisung
Realzahlarithmetik	Add Real	1,48 µs/Anweisung	1,78 µs/Anweisung

Hinweis

Viele Variablen wirken sich auf die gemessenen Zeiten aus. Die oben angegebenen Leistungszeiten gelten für die schnellsten Anweisungen in dieser Kategorie und fehlerfreie Programme.

A.5.2 Von der CPU 1212C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A-31 Von der CPU 1212C unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung
Bausteine	Typ	OB, FB, FC, DB
	Größe	Bis zur Größe des Arbeitsspeichers
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	FB und FC: 1 bis 65535 (z. B. FB 1 bis FB 65535) DB: 1 bis 59999
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ¹
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
OBs	Programmzyklus	Mehrere
	Anlauf	Mehrere
	Verzögerungsalarm	4 (1 pro Ereignis)
	Weckalarme	4 (1 pro Ereignis)
	Prozessalarme	50 (1 pro Ereignis)
	Zeitfehleralarme	1
	Diagnosefehleralarme	1
	Ziehen oder Stecken von Modulen	1
	Fehler bei Baugruppenträger oder Station	1
	Uhrzeitalarm	Mehrere
	Status	1
	Update	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Byte pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Byte • Int, UInt: 6 Byte • DInt, UDInt: 12 Byte

¹ Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

Tabelle A-32 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	1
Typ	Ethernet
HMI-Gerät	4
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> • 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV • 8 CPU/CPU-Verbindungen (Client oder Server) für GET/PUT-Daten • 6 Verbindungen für dynamische Zuordnung zu GET/PUT oder offener Benutzerkommunikation • Max. 64 Verbindungen für Sicherheitszertifikate
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu Logik)	Wandler potentialgetrennt, 1500 V AC (Typprüfung) ¹
Kabelart	CAT5e geschirmt
Schnittstellen	
Anzahl PROFINET-Schnittstellen	1
Anzahl PROFIBUS-Schnittstellen	0
Schnittstelle	
Hardware für Schnittstelle	
Anzahl Ports	1
Integrierter Switch	Nein
RJ-45 (Ethernet)	Ja; X1
Protokolle	
PROFINET IO-Controller	Ja
PROFINET IO-Device	Ja
SIMATIC-Kommunikation	Ja
Offene IE-Kommunikation	Ja
Webserver	Ja
Medienredundanz	Nein
PROFINET IO-Controller	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein
Offene IE-Kommunikation	Ja
IRT	Nein
MRP	Nein
PROFlenergy	Ja. Die S7-1200 CPU unterstützt nur die PROFlenergy-Einheit (mit I-Device-Funktionalität).
Priorisierter Anlauf	Ja (max. 16 PROFINET-Geräte)
Max. Anzahl anschließbarer IO-Devices	16
Max. Anzahl anschließbarer IO-Devices für RT	16
Max. Anzahl davon in Reihe	16

Technische Daten		Beschreibung
Max. Anzahl gleichzeitig aktivierbarer/deaktivierbarer IO-Devices		8
Aktualisierungszeiten		Der Mindestwert der Aktualisierungszeit ist auch abhängig von der für PROFINET IO eingestellten Kommunikationskomponente, der Anzahl von IO-Devices und der Menge der konfigurierten Anwenderdaten.
Mit RT		
Sendetakt von 1 ms		1 ms bis 512 ms
PROFINET IO-Device		
Dienste		
PG/OP-Kommunikation		Ja
S7-Routing		Ja
Taktsynchroner Betrieb		Nein
Offene IE-Kommunikation		Ja
IRT, unterstützt		Nein
MRP, unterstützt		Nein
PROFenergy		Ja
Shared Device		Ja
Max. Anzahl IO-Controller mit Shared Device		2
SIMATIC-Kommunikation		
S7-Kommunikation, als Server		Ja
S7-Kommunikation, als Client		Ja
Max. Anwenderdaten pro Auftrag		Siehe Online-Hilfe (S7-Kommunikation, Umfang der Anwenderdaten)
Offene IE-Kommunikation		
TCP/IP:		Ja
	Max. Länge der Daten	8 KB
	Unterstützung für mehrere passive Anschlüsse pro Port	Ja
ISO-on-TCP (RFC1006):		Ja
	Max. Länge der Daten	8 KB
UDP		Ja
	Max. Länge der Daten	1472 Byte
DHCP		Nein
SNMP		Ja
DCP		Ja
LLDP		Ja

¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr aufgrund von gefährlichen Spannungen bei kurzfristigen Netzwerkstörungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

Tabelle A-33 Stromversorgung

Technische Daten	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Spannungsbereich	85 bis 264 V AC	20,4 V DC bis 28,8 V DC	
Netzfrequenz	47 bis 63 Hz	--	

Technische Daten		CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Einschaltstrom (max. Last)	Nur CPU	80 mA bei 120 V AC 40 mA bei 240 V AC	400 mA bei 24 V DC	
	CPU mit allen Erweite- rungsbaugruppen	240 mA bei 120 V AC 120 mA bei 240 V AC	1200 mA bei 24 V DC	
Einschaltstrom (max.)		20 A bei 264 V AC	12 A bei 28,8 V DC	
I ² t		0,8 A ² s	0,5 A ² s	
Elektrische Trennung (Eingangsleistung zu Logik)		1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funkti- onserde		max. 0,5 mA	--	
Verzögerungszeit (Spannungsverlust)		20 mA bei 120 V AC 80 mA bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender aus- tauschbar		3 A, 250 V, träge		

Tabelle A-34 Geberversorgung

Technische Daten	CPU 1212C AC/DC/Relais	CPU 1212C DC/DC/Relais	CPU 1212C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L+ minus 4 V Min. DC	
Nennausgangsstrom (max.)	300 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Geberver- sorgung)	Nicht elektrisch getrennt		

A.5.3 Digitale Eingänge und Ausgänge

Tabelle A-35 Digitale Eingänge

Technische Daten	CPU 1212C AC/DC/Relais, DC/DC/Relais und DC/DC/DC
Anzahl der Eingänge	8
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, nominal
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Filterzeiten	µs-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0

Technische Daten	CPU 1212C AC/DC/Relais, DC/DC/Relais und DC/DC/DC
HSC-Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	100/80 kHz (Ea.0 bis Ea.5) 30 /20 kHz (Ea.6 bis Ea.7)
Anzahl gleichzeitig eingeschalteter Eingänge	4 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal 8 bei 55 °C horizontal oder 45 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge

Tabelle A-36 Digitale Ausgänge

Technische Daten	CPU 1212C AC/DC/Relais und DC/DC/Relais	CPU 1212C DC/DC/DC
Ausgänge	6	
Typ	Relais, mechanisch	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω, wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 μA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	2	1
Elektrische Trennung (Gruppe zu Gruppe)	1500 V AC ¹	--
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 μs von Aus nach Ein max. 3,0 μs von Ein nach Aus
Schaltverzögerung (Aa.4 bis Aa.5)	max. 10 ms	max. 5 μs von Aus nach Ein max. 20 μs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator	Nicht empfohlen ²	100 kHz (Aa.0 bis Aa.3) ³ , 2 Hz min. 20 kHz (Aa.4 bis Aa.5) ³
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)	
Parallele Ausgänge für erhöhte Last	Nein	

Technische Daten	CPU 1212C AC/DC/Relais und DC/DC/Relais	CPU 1212C DC/DC/DC
Anzahl gleichzeitig eingeschalteter Ausgänge	3 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal 6 bei 55 °C horizontal oder 45 °C vertikal	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

- ¹ Die elektrische Trennung zwischen Gruppen von Relais trennt die Netzspannung von SELV/PELV und trennt unterschiedliche Phasen bis zu 250 V AC von der Leitung zur Erde.
- ² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.
- ³ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10 % des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

A.5.4 Analoge Eingänge

Tabelle A-37 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32512 bis 32767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 1284).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

A.5.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A-38 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
Abtastzeit	4,17 ms	5 ms	25 ms

A.5.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A-39 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.5.4.3 Messbereiche der analogen Eingänge für Spannung (CPUs)

Tabelle A-40 Darstellung Analogeingang für Spannung (CPUs)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,852 V	Überlauf
32512	7F00		
32511	7EFF	11,759 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Negative Werte		Negative Werte werden nicht unterstützt	

A.5.5 Schaltpläne der CPU 1212C

Tabelle A-41 CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)

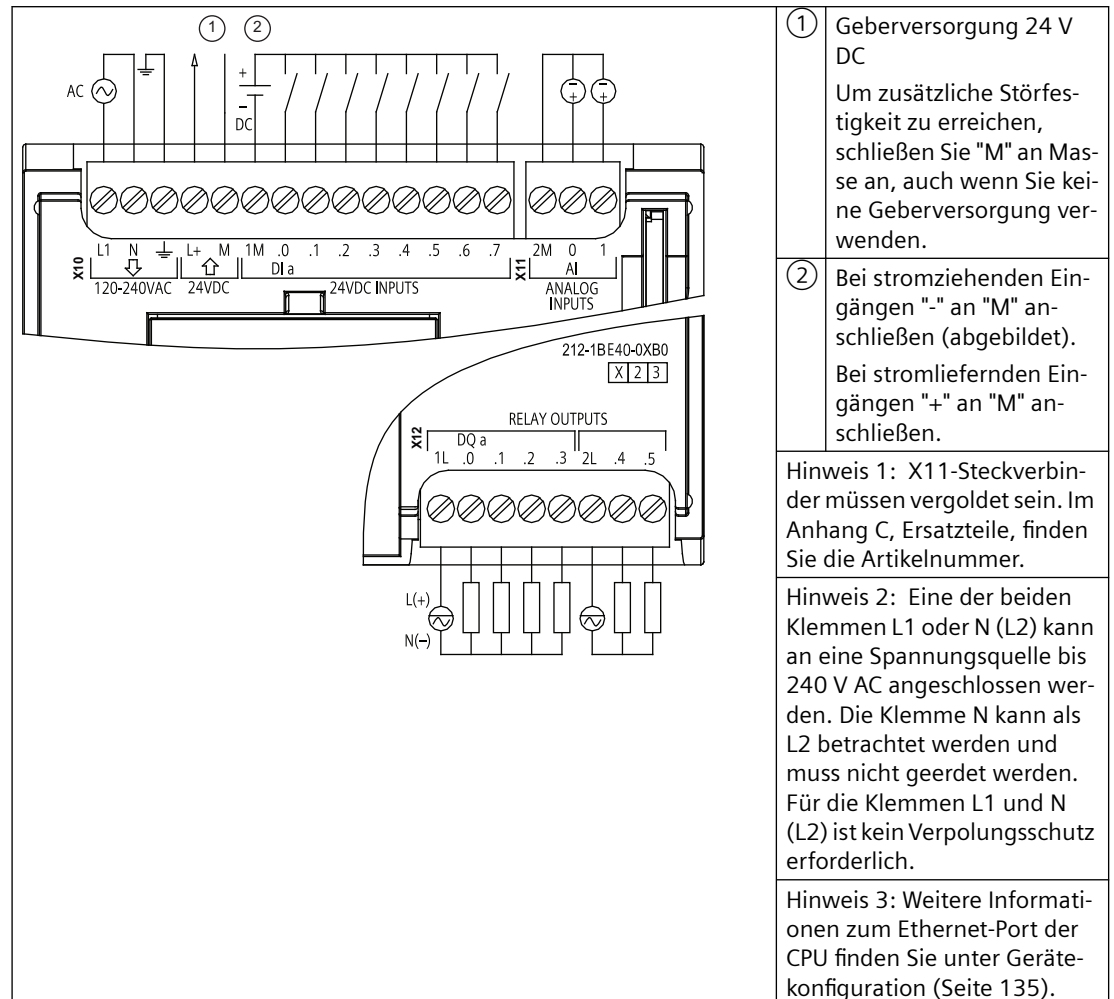


Tabelle A-42 Anschlussbelegung für die CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Geberversorgung 24 V DC	--	DO a.2
5	M / Geberversorgung 24 V DC	--	DO a.3
6	1M	--	2L
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5

Pin	X10	X11 (vergoldet)	X12
9	DI a.2	--	--
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tabelle A-43 CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)

① Geberversorgung 24 V DC
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.

② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).
Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

Hinweis 2: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

Tabelle A-44 Anschlussbelegung für die CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Geberversorgung 24 V DC	--	DO a.2
5	M / Geberversorgung 24 V DC	--	DO a.3
6	1M	--	2L
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	--

Pin	X10	X11 (vergoldet)	X12
10	DI a.3	--	--
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Tabelle A-45 CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

① Geberversorgung 24 V DC
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.

② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

Hinweis 2: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

Tabelle A-46 Anschlussbelegung für die CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / Geberversorgung 24 V DC	--	DO a.1
5	M / Geberversorgung 24 V DC	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	--
10	DI a.3	--	--

Pin	X10	X11 (vergoldet)	X12
11	DI a.4	--	--
12	DI a.5	--	--
13	DI a.6	--	--
14	DI a.7	--	--

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

A.6 CPU 1214C

A.6.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A-47 Allgemein

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Artikelnummer	6ES7214-1BG40-0XB0	6ES7214-1HG40-0XB0	6ES7214-1AG40-0XB0
Abmessungen (B x H x T) (mm)	110 x 100 x 75		
Versandgewicht	475 Gramm	435 Gramm	415 Gramm
Leistungsverlust	14 W	12 W	
Verfügbarer elektrischer Strom (SM- und CM-Bus)	max. 1.600 mA (5 V DC)		
Verfügbarer elektrischer Strom (24 V DC)	max. 400 mA (Geberversorgung)		
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang		

Tabelle A-48 CPU-Merkmale

Technische Daten	Beschreibung	
Anwenderspeicher (Siehe "Allgemeine technische Daten" (Seite 1253), "Speicherung im internen CPU-Speicher".)	Arbeitsspeicher	100 KB
	Ladespeicher	4 MB intern, erweiterbar bis auf SD-Kartengröße
	Remanent	14 kByte
Integrierte digitale E/A	14 Eingänge/10 Ausgänge	
Integrierte analoge E/A	2 Eingänge	
Größe des Prozessabbilds	1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)	

Technische Daten	Beschreibung
Merker (M)	8192 Byte
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs) • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs)
Zusätzliche Signalmodule	max. 8 SMs
Erweiterung SB, CB, BB	max. 1
Zusätzliche Kommunikationsmodule	max. 3 CM
Schnelle Zähler	<p>Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge. Siehe "Anschlussbelegung des Hardwareeingangs" (Seite 568) für CPU 1214C: Zuweisungen der HSC-Standardadressen.</p> <ul style="list-style-type: none"> • 100/180 kHz (Ea.0 bis Ea.5) • 30/120 kHz (Ea.6 bis Eb.5)
Impulsausgänge ²	<p>Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge</p> <ul style="list-style-type: none"> • 100 kHz (Aa.0 bis Aa.3) • 20 kHz (Aa.4 bis Ab.1)
Eingänge für Impulsabgriff	14
Verzögerungsalarme	4 mit Auflösung von 1 ms
Weckalarme	4 mit Auflösung von 1 ms
Flankenalarme	12 steigend und 12 fallend (16 und 16 mit optionalem Signalboard)
Memory Card	SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

¹ Die langsamere Geschwindigkeit gilt, wenn der HSC als A/B-Zähler konfiguriert ist.

² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A-49 Leistung

Art der Anweisung		Ausführungsgeschwindigkeit	
		Direkte Adressierung (E, A und M)	DB-Zugriff
Boolescher Wert		0,08 µs/Anweisung	
Verschieben	Move_Bool	0,3 µs/Anweisung	1,17 µs/Anweisung
	Move_Word	0,137 µs/Anweisung	1,0 µs/Anweisung
	Move_Real	0,72 µs/Anweisung	1,0 µs/Anweisung
Realzahlarithmetik	Add Real	1,48 µs/Anweisung	1,78 µs/Anweisung

Hinweis

Viele Variablen wirken sich auf die gemessenen Zeiten aus. Die oben angegebenen Leistungszeiten gelten für die schnellsten Anweisungen in dieser Kategorie und fehlerfreie Programme.

A.6.2 Von der CPU 1214C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A-50 Von der CPU 1214C unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung
Bausteine	Typ	OB, FB, FC, DB
	Größe	OB, FB, FC: 64 KB DB: bis zur Größe des Arbeitsspeichers
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	FB und FC: 1 bis 65535 (z. B. FB 1 bis FB 65535) DB: 1 bis 59999
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ¹
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
OBs	Programmzyklus	Mehrere
	Anlauf	Mehrere
	Verzögerungsalarme	4 (1 pro Ereignis)
	Weckalarme	4 (1 pro Ereignis)
	Prozessalarme	50 (1 pro Ereignis)
	Zeitfehleralarme	1
	Diagnosefehleralarme	1
	Ziehen oder Stecken von Modulen	1
	Fehler bei Baugruppenträger oder Station	1
	Uhrzeitalarm	Mehrere
	Status	1
	Update	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
MC-PreServo	1	
MC-PostServo	1	
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Bytes pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Bytes • Int, UInt: 6 Bytes • DInt, UDInt: 12 Bytes

¹ Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

Tabelle A-51 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	1
Typ	Ethernet
HMI-Gerät	4
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> • 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV • 8 CPU/CPU-Verbindungen (Client oder Server) für GET/PUT-Daten • 6 Verbindungen für dynamische Zuordnung zu GET/PUT oder offener Benutzerkommunikation • Max. 64 Verbindungen für Sicherheitszertifikate
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu Logik)	Wandler potentialgetrennt, 1500 V AC (Typprüfung) ¹
Kabelart	CAT5e geschirmt
Schnittstellen	
Anzahl PROFINET-Schnittstellen	1
Anzahl PROFIBUS-Schnittstellen	0
Schnittstelle	
Hardware für Schnittstelle	
Anzahl Ports	1
Integrierter Switch	Nein
RJ-45 (Ethernet)	Ja; X1
Protokolle	
PROFINET IO-Controller	Ja
PROFINET IO-Device	Ja
SIMATIC-Kommunikation	Ja
Offene IE-Kommunikation	Ja
Webserver	Ja
Medienredundanz	Nein
PROFINET IO-Controller	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein
Offene IE-Kommunikation	Ja
IRT	Nein
MRP	Nein
PROFlenergy	Ja. Die S7-1200 CPU unterstützt nur die PROFlenergy-Einheit (mit I-Device-Funktionalität).
Priorisierter Anlauf	Ja (max. 16 PROFINET-Geräte)
Max. Anzahl anschließbarer IO-Devices	16
Max. Anzahl anschließbarer IO-Devices für RT	16
Max. Anzahl davon in Reihe	16

Technische Daten		Beschreibung
Max. Anzahl gleichzeitig aktivierbarer/deaktivierbarer IO-Devices		8
Aktualisierungszeiten		Der Mindestwert der Aktualisierungszeit ist auch abhängig von der für PROFINET IO eingestellten Kommunikationskomponente, der Anzahl von IO-Devices und der Menge der konfigurierten Anwenderdaten.
Mit RT		
Sendetakt von 1 ms		1 ms bis 512 ms
PROFINET IO-Device		
Dienste		
PG/OP-Kommunikation		Ja
S7-Routing		Ja
Taktsynchroner Betrieb		Nein
Offene IE-Kommunikation		Ja
IRT, unterstützt		Nein
MRP, unterstützt		Nein
PROFenergy		Ja
Shared Device		Ja
Max. Anzahl IO-Controller mit Shared Device		2
SIMATIC-Kommunikation		
S7-Kommunikation, als Server		Ja
S7-Kommunikation, als Client		Ja
Max. Anwenderdaten pro Auftrag		Siehe Online-Hilfe (S7-Kommunikation, Umfang der Anwenderdaten)
Offene IE-Kommunikation		
TCP/IP:		Ja
	Max. Länge der Daten	8 KB
	Unterstützung für mehrere passive Anschlüsse pro Port	Ja
ISO-on-TCP (RFC1006):		Ja
	Max. Länge der Daten	8 KB
UDP		Ja
	Max. Länge der Daten	1472 Byte
DHCP		Nein
SNMP		Ja
DCP		Ja
LLDP		Ja

¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr aufgrund von gefährlichen Spannungen bei kurzfristigen Netzwerkstörungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

Tabelle A-52 Stromversorgung

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Spannungsbereich	85 bis 264 V AC	20,4 V DC bis 28,8 V DC	
Netzfrequenz	47 ... 63 Hz	--	

Technische Daten		CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Einschaltstrom (max. Last)	Nur CPU	100 mA bei 120 V AC 50 mA bei 240 V AC	500 mA bei 24 V DC	
	CPU mit allen Erweiterungsbaugruppen	300 mA bei 120 V AC 150 mA bei 240 V AC	1500 mA bei 24 V DC	
Einschaltstrom (max.)		20 A bei 264 V AC	12 A bei 28,8 V DC	
I ² t		0,8 A ² s	0,5 A ² s	
Elektrische Trennung (Eingangsleistung zu Logik)		1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funktionserde		max. 0.5 mA	-	
Verzögerungszeit (Spannungsverlust)		20 mA bei 120 V AC 80 mA bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender austauschbar		3 A, 250 V, träge		

Tabelle A-53 Geberversorgung

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L+ minus 4 V DC (min.)	
Nennausgangsstrom (max.)	400 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Sensorversorgung)	Nicht elektrisch getrennt		

A.6.3 Digitale Eingänge und Ausgänge

Tabelle A-54 Digitale Eingänge

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
Anzahl der Eingänge	14		
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)		
Nennspannung	24 V DC bei 4 mA, nominal		
Zulässige Dauerspannung	max. 30 V DC		
Stoßspannung	35 V DC für 0,5 s		
Signal logisch 1 (min.)	15 V DC bei 2,5 mA		
Signal logisch 0 (max.)	5 V DC bei 1 mA		
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)		
Potentialgetrennte Gruppen	1		
Filterzeiten	µs-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0		

A.6 CPU 1214C

Technische Daten	CPU 1214C AC/DC/Relais	CPU 1214C DC/DC/Relais	CPU 1214C DC/DC/DC
HSC-Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	100/80 kHz (Ea.0 bis Ea.5) 30/20 kHz (Ea.6 bis Eb.5)		
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> • 7 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 14 bei 55 °C horizontal oder 45 °C vertikal 		
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge		

Tabelle A-55 Digitale Ausgänge

Technische Daten	CPU 1214C AC/DC/Relais und DC/DC/Relais	CPU 1214C DC/DC/DC
Ausgänge	10	
Typ	Relais, mechanisch	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω, wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	2	1
Elektrische Trennung (Gruppe zu Gruppe)	1500 V AC ¹	--
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Schaltverzögerung (Aa.4 bis Ab.1)	max. 10 ms	max. 5 µs von Aus nach Ein max. 20 µs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator	Nicht empfohlen ²	100 kHz (Aa.0 bis Aa.3) ³ , 2 Hz min. 20 kHz (Aa.4 bis Ab.1) ³
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)	
Parallele Ausgänge für erhöhte Last	Nein	

Technische Daten	CPU 1214C AC/DC/Relais und DC/DC/Relais	CPU 1214C DC/DC/DC
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> • 5 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 10 bei 55 °C horizontal oder 45 °C vertikal 	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

- ¹ Die elektrische Trennung zwischen Gruppen von Relais trennt die Netzspannung von SELV/PELV und trennt unterschiedliche Phasen bis zu 250 V AC von der Leitung zur Erde.
- ² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge verwenden zu können.
- ³ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10 % des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

A.6.4 Analoge Eingänge

Tabelle A-56 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32512 bis 32767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 1296).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

A.6.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A-57 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
Abtastzeit	4,17 ms	5 ms	25 ms

A.6.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A-58 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.6.4.3 Messbereiche der analogen Eingänge für Spannung (CPUs)

Tabelle A-59 Darstellung Analogeingang für Spannung (CPUs)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,852 V	Überlauf
32512	7F00		
32511	7EFF	11,759 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Negative Werte		Negative Werte werden nicht unterstützt	

A.6.5 Schaltpläne der CPU 1214C

Tabelle A-60 CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)

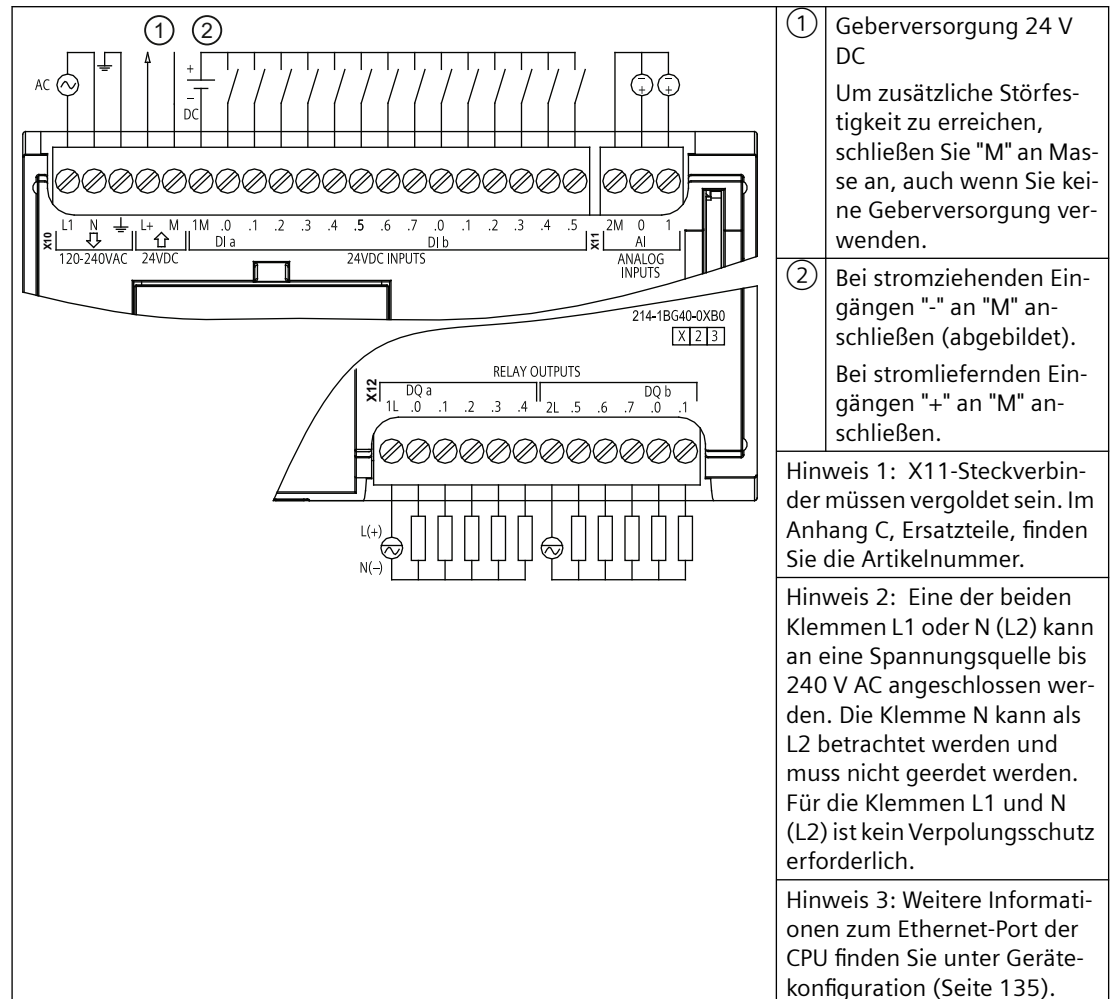


Tabelle A-61 Anschlussbelegung für die CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120-240 V AC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Gebersversorgung 24 V DC	--	DO a.2
5	M / Gebersversorgung 24 V DC	--	DO a.3
6	1M	--	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5

Pin	X10	X11 (vergoldet)	X12
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO 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	--	--

Tabelle A-62 CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)

The diagram illustrates the terminal connections for the CPU 1214C DC/DC/Relais. It shows three main terminal blocks: X10, X11, and X12. X10 is divided into sections for 24VDC power supply (L+, M, N(-)), 24VDC inputs (DI a, DI b), and analog inputs (AI 0, AI 1). X11 is used for relay outputs (DO a, DO b). X12 is used for relay outputs (DO a, DO b). The diagram also shows the internal wiring of the CPU, including the DC power supply and the relay outputs.

① Geberversorgung 24 V DC
Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.

② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).
Bei stromliefernden Eingängen "+" an "M" anschließen.

Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

Hinweis 2: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

Tabelle A-63 Anschlussbelegung für die CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AI 0	DO a.0
3	Funktionserde	AI 1	DO a.1
4	L+ / Geberversorgung 24 V DC	--	DO a.2

Pin	X10	X11 (vergoldet)	X12
5	M / Gebersversorgung 24 V DC	--	DO a.3
6	1M	--	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO 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	--	--

Tabelle A-64 CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)

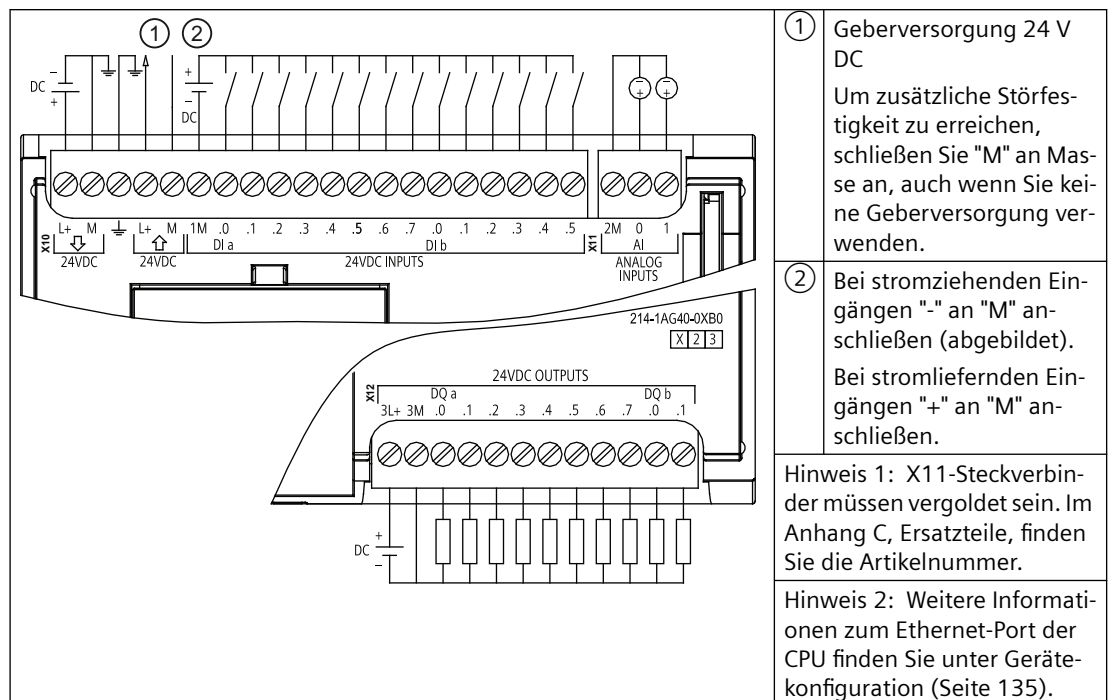


Tabelle A-65 Anschlussbelegung für die CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	3L+
2	M / 24 V DC	AI 0	3M
3	Funktionserde	AI 1	DO a.0
4	L+ / Geberversorgung 24 V DC	--	DO a.1
5	M / Geberversorgung 24 V DC	--	DO a.2
6	1M	--	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO 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	--	--

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

A.7 CPU 1215C

A.7.1 Allgemeine technische Daten und Leistungsmerkmale

Tabelle A-66 Allgemein

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Artikelnummer	6ES7215-1BG40-0XB0	6ES7215-1HG40-0XB0	6ES7215-1AG40-0XB0
Abmessungen (B x H x T) (mm)	130 x 100 x 75		
Versandgewicht	585 Gramm	550 Gramm	520 Gramm
Leistungsverlust	14 W	12 W	

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Verfügbare elektrischer Strom (SM- und CM-Bus)	max. 1600 mA (5 V DC)		
Verfügbare elektrischer Strom (24 V DC)	max. 400 mA (Geberversorgung)		
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang		

Tabelle A-67 CPU-Merkmale

Technische Daten		Beschreibung
Anwenderspeicher (Siehe "Allgemeine technische Daten" (Seite 1253), "Speicherung im internen CPU-Speicher".)	Arbeitsspeicher	125 KB
	Ladespeicher	4 MB intern, erweiterbar bis auf SD-Kartengröße
	Remanent	14 kByte
Integrierte digitale E/A		14 Eingänge/10 Ausgänge
Integrierte analoge E/A		2 Eingänge/2 Ausgänge
Größe des Prozessabbilds		1024 Byte Eingänge (E)/1024 Byte Ausgänge (A)
Merker (M)		8192 Byte
Temporärer (lokaler) Speicher		<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs) • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs)
Zusätzliche Signalmodule		max. 8 SMs
Erweiterung SB, CB, BB		max. 1
Zusätzliche Kommunikationsmodule		max. 3 CM
Schnelle Zähler		<p>Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge. Siehe "Anschlussbelegung des Hardwareeingangs" (Seite 568) für CPU 1215C: Zuweisungen der HSC-Standardadressen.</p> <ul style="list-style-type: none"> • 100/180 kHz (Ea.0 bis Ea.5) • 30/120 kHz (Ea.6 bis Eb.5)
Impulsausgänge ²		<p>Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge</p> <ul style="list-style-type: none"> • 100 kHz (Aa.0 bis Aa.3) • 20 kHz (Aa.4 bis Ab.1)
Eingänge für Impulsabgriff		14
Verzögerungsalarme		4 mit Auflösung von 1 ms
Weckalarme		4 mit Auflösung von 1 ms
Flankenalarme		12 steigend und 12 fallend (16 und 16 mit optionalem Signalboard)
Memory Card		SIMATIC Memory Card (optional)

Technische Daten	Beschreibung
Genauigkeit Echtzeituhr	+/- 60 Sekunden/Monat
Pufferung Echtzeituhr	Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

- ¹ Die langsamere Geschwindigkeit gilt, wenn der HSC als A/B-Zähler konfiguriert ist.
- ² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

Tabelle A-68 Leistung

Art der Anweisung		Ausführungsgeschwindigkeit	
		Direkte Adressierung (E, A und M)	DB-Zugriff
Boolescher Wert		0,08 µs/Anweisung	
Verschieben	Move_Bool	0,3 µs/Anweisung	1,17 µs/Anweisung
	Move_Word	0,137 µs/Anweisung	1,0 µs/Anweisung
	Move_Real	0,72 µs/Anweisung	1,0 µs/Anweisung
Realzahlarithmetik	Add Real	1,48 µs/Anweisung	1,78 µs/Anweisung

Hinweis

Viele Variablen wirken sich auf die gemessenen Zeiten aus. Die oben angegebenen Leistungszeiten gelten für die schnellsten Anweisungen in dieser Kategorie und fehlerfreie Programme.

A.7.2 Von der CPU 1215C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A-69 Von der CPU 1215C unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung
Bausteine	Typ	OB, FB, FC, DB
	Größe	OB, FB, FC: 64 kBytes DB: bis zur Größe des Arbeitsspeichers
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	FB und FC: 1 bis 65535 (z. B. FB 1 bis FB 65535) DB: 1 bis 59999
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ¹
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.

Element		Beschreibung
OBs	Programmzyklus	Mehrere
	Anlauf	Mehrere
	Verzögerungsalarme	4 (1 pro Ereignis)
	Weckalarne	4 (1 pro Ereignis)
	Prozessalarne	50 (1 pro Ereignis)
	Zeitfehleralarne	1
	Diagnosefehleralarne	1
	Ziehen oder Stecken von Modulen	1
	Fehler bei Baugruppenträger oder Station	1
	Uhrzeitalarm	Mehrere
	Status	1
	Update	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Bytes pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Bytes • Int, UInt: 6 Bytes • DInt, UDInt: 12 Bytes

¹ Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

Tabelle A-70 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	2
Typ	Ethernet
HMI-Gerät	4
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> • 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV • 8 CPU/CPU-Verbindungen (Client oder Server) für GET/PUT-Daten • 6 Verbindungen für dynamische Zuordnung zu GET/PUT oder offener Benutzerkommunikation • Max. 64 Verbindungen für Sicherheitszertifikate

Technische Daten	Beschreibung
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu Logik)	Wandler potentialgetrennt, 1500 V AC (Typprüfung) ¹
Kabelart	CAT5e geschirmt
Schnittstellen	
Anzahl PROFINET-Schnittstellen	1
Anzahl PROFIBUS-Schnittstellen	0
Schnittstelle	
Hardware für Schnittstelle	
Anzahl Ports	2
Integrierter Switch	Ja
RJ-45 (Ethernet)	Ja; X1
Protokolle	
PROFINET IO-Controller	Ja
PROFINET IO-Device	Ja
SIMATIC-Kommunikation	Ja
Offene IE-Kommunikation	Ja
Webserver	Ja
Medienredundanz	Ja
PROFINET IO-Controller	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein
Offene IE-Kommunikation	Ja
IRT	Nein
MRP	Ja; als MRP-Client
PROFIenergy	Ja. Die S7-1200 CPU unterstützt nur die PROFIenergy-Einheit (mit I-Device-Funktionalität).
Priorisierter Anlauf	Ja (max. 16 PROFINET-Geräte)
Max. Anzahl anschließbarer IO-Devices	16
Max. Anzahl anschließbarer IO-Devices für RT	16
Max. Anzahl davon in Reihe	16
Max. Anzahl gleichzeitig aktivierbarer/deaktivierbarer IO-Devices	8
Aktualisierungszeiten	Der Mindestwert der Aktualisierungszeit ist auch abhängig von der für PROFINET IO eingestellten Kommunikationskomponente, der Anzahl von IO-Devices und der Menge der konfigurierten Anwenderdaten.
Mit RT	
Sendetakt von 1 ms	1 ms bis 512 ms
PROFINET IO-Device	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein

Technische Daten	Beschreibung
Offene IE-Kommunikation	Ja
IRT, unterstützt	Nein
MRP, unterstützt	Ja
PROFenergy	Ja
Shared Device	Ja
Max. Anzahl IO-Controller mit Shared Device	2
SIMATIC-Kommunikation	
S7-Kommunikation, als Server	Ja
S7-Kommunikation, als Client	Ja
Max. Anwenderdaten pro Auftrag	Siehe Online-Hilfe (S7-Kommunikation, Umfang der Anwenderdaten)
Offene IE-Kommunikation	
TCP/IP:	Ja
Max. Länge der Daten	8 KB
Unterstützung für mehrere passive Anschlüsse pro Port	Ja
ISO-on-TCP (RFC1006):	Ja
Max. Länge der Daten	8 KB
UDP:	Ja
Max. Länge der Daten	1472 Byte
DHCP	Nein
SNMP	Ja
DCP	Ja
LLDP	Ja

- ¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr aufgrund von gefährlichen Spannungen bei kurzfristigen Netzwerkstörungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

Tabelle A-71 Stromversorgung

Technische Daten		CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Spannungsbereich		85 bis 264 V AC	20,4 V DC bis 28,8 V DC	
Netzfrequenz		47 bis 63 Hz	--	
Einschaltstrom (max. Last)	Nur CPU	100 mA bei 120 V AC 50 mA bei 240 V AC	500 mA bei 24 V DC	
	CPU mit allen Erweiterungsbaugruppen	300 mA bei 120 V AC 150 mA bei 240 V AC	1500 mA bei 24 V DC	
Einschaltstrom (max.)		20 A bei 264 V AC	12 A bei 28,8 V DC	
I ² t		0,8 A ² s	0,5 A ² s	
Elektrische Trennung (Eingangsleistung zu Logik)		1500 V AC	Nicht elektrisch getrennt	
Kriechstrom an Erde, AC-Leitung an Funktionserde		max. 0,5 mA	-	

A.7 CPU 1215C

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Verzögerungszeit (Spannungsverlust)	20 mA bei 120 V AC 80 mA bei 240 V AC	10 ms bei 24 V DC	
Interne Sicherung, nicht durch Anwender austauschbar	3 A, 250 V, träge		

Tabelle A-72 Gebersversorgung

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Spannungsbereich	20,4 bis 28,8 V DC	L+ minus 4 V DC (min.)	
Nennausgangsstrom (max.)	400 mA (kurzschlussfest)		
Max. Welligkeit/Störströme (<10 MHz)	< 1 V Spitze-Spitze	Wie Eingangsleitung	
Elektrische Trennung (CPU-Logik zu Gebersversorgung)	Nicht elektrisch getrennt		

A.7.3 Digitale Eingänge und Ausgänge

Tabelle A-73 Digitale Eingänge

Technische Daten	CPU 1215C AC/DC/Relais	CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Anzahl der Eingänge	14		
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)		
Nennspannung	24 V DC bei 4 mA, nominal		
Zulässige Dauerspannung	max. 30 V DC		
Stoßspannung	35 V DC für 0,5 s		
Signal logisch 1 (min.)	15 V DC bei 2,5 mA		
Signal logisch 0 (max.)	5 V DC bei 1 mA		
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)		
Potentialgetrennte Gruppen	1		
Filterzeiten	us-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0		
HSC-Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	100/80 kHz (Ea.0 bis Ea.5) 30/20 kHz (Ea.6 bis Eb.5)		
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> 7 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal 14 bei 55 °C horizontal oder 45 °C vertikal 		
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt, 50 m geschirmt für HSC-Eingänge		

Tabelle A-74 Digitale Ausgänge

Technische Daten	CPU 1215C AC/DC/Relais und CPU 1215C DC/DC/Relais	CPU 1215C DC/DC/DC
Ausgänge	10	
Typ	Relais, mechanisch	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	min. 20 V DC
Signal logisch 0 bei 10 k Ω Last	--	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	max. 10 μ A
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	2	1
Elektrische Trennung (Gruppe zu Gruppe)	1500 V AC ¹	--
Induktive Klemmspannung	--	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 μ s von Aus nach Ein max. 3,0 μ s von Ein nach Aus
Schaltverzögerung (Aa.4 bis Ab.1)	max. 10 ms	max. 5 μ s von Aus nach Ein max. 20 μ s von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	--
Frequenz Impulsgenerator	Nicht empfohlen ²	100 kHz (Aa.0 bis Aa.3) ³ , 2 Hz min. 20 kHz (Aa.4 bis Ab.1) ³
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)	
Parallele Ausgänge für erhöhte Last	Nein	
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> • 5 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 10 bei 55 °C horizontal oder 45 °C vertikal 	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

¹ Die elektrische Trennung zwischen Gruppen von Relais trennt die Netzspannung von SELV/PELV und trennt unterschiedliche Phasen bis zu 250 V AC von der Leitung zur Erde.

² Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge verwenden zu können.

³ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10 % des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

A.7.4 Analogeingänge und -ausgänge

Tabelle A-75 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32512 bis 32767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 1308).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

A.7.4.1 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A-76 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
Abtastzeit	4,17 ms	5 ms	25 ms

A.7.4.2 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A-77 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.7.4.3 Messbereiche der analogen Eingänge für Spannung (CPUs)

Tabelle A-78 Darstellung Analogeingang für Spannung (CPUs)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,852 V	Überlauf
32512	7F00		
32511	7EFF	11,759 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Negative Werte		Negative Werte werden nicht unterstützt	

A.7.4.4 Technische Daten der Analogausgänge

Tabelle A-79 Analoge Ausgänge

Technische Daten	Beschreibung
Ausgänge	2
Typ	Strom
Vollausschlagsbereich	0 ... 20 mA
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	20,01 bis 23,52 mA
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	siehe Fußnote ¹
Datenwort Überlaufbereich	32512 bis 32767
Auflösung	10 Bit
Ausgangsimpedanz	≤ 500 Ω max.
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0 % / 3,5 % des Vollausschlags
Einschwingzeit	2 ms
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

¹ Bei Überlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: "Ersatzwert aufschalten" oder "Letzten Wert halten".

Tabelle A-80 Darstellung Analogausgang für Strom (CPU 1215C und CPU 1217C)

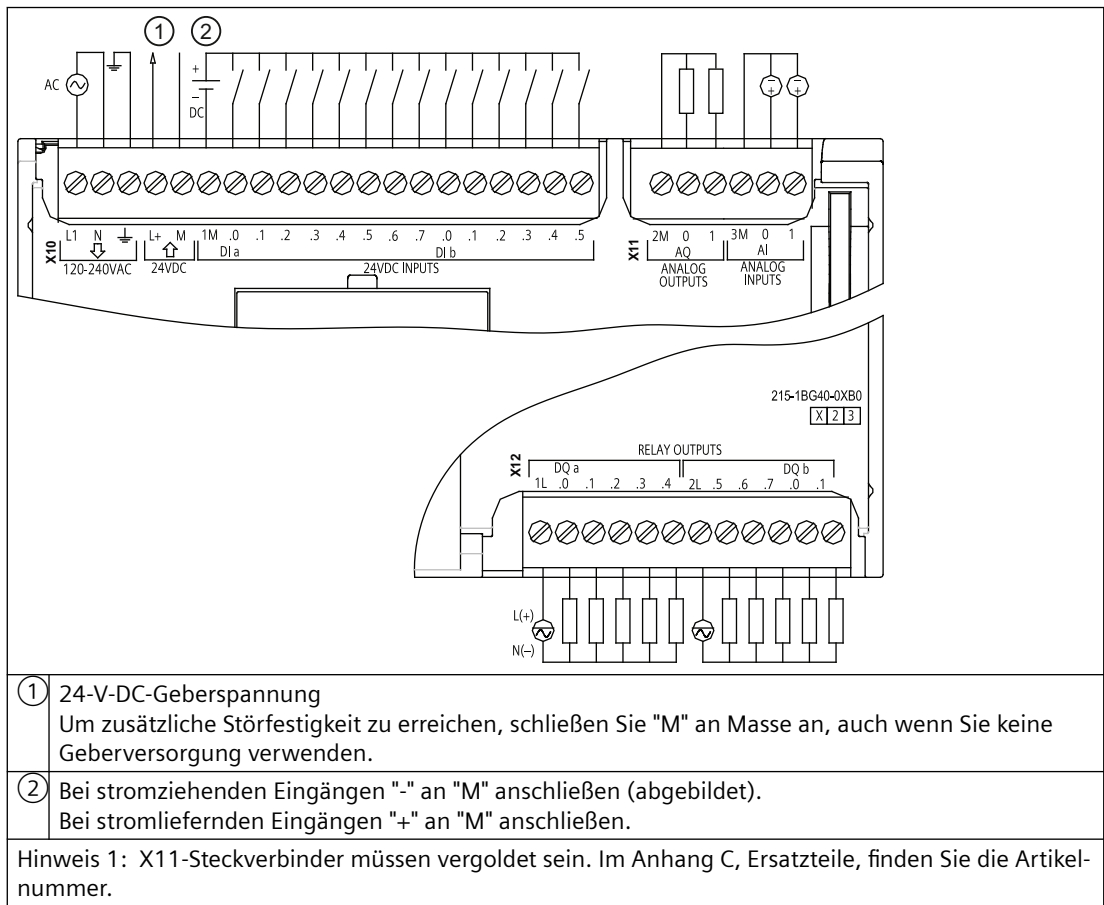
System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	

System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32511	7EFF	23,52 mA	Überschwingbereich
27649	6C01		
27648	6C00	20 mA	Bemessungsbereich
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Negative Werte		Negative Werte werden nicht unterstützt	

¹ Bei Überlauf verhalten sich die Analoggänge entsprechend den eingestellten Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: "Ersatzwert aufschalten" oder "Letzten Wert halten".

A.7.5 Schaltpläne der CPU 1215C

Tabelle A-81 CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)



Hinweis 2: Eine der beiden Klemmen L1 oder N (L2) kann an eine Spannungsquelle bis 240 V AC angeschlossen werden. Die Klemme N kann als L2 betrachtet werden und muss nicht geerdet werden. Für die Klemmen L1 und N (L2) ist kein Verpolungsschutz erforderlich.

Hinweis 3: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

Tabelle A-82 Anschlussbelegung für die CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 120-240 V AC	2 M	1L
2	N / 120 - 240 V AC	AO 0	DO a.0
3	Funktionserde	AO 1	DO a.1
4	L+ / Geberversorgung 24 V DC	3M	DO a.2
5	M / Geberversorgung 24 V DC	AI 0	DO a.3
6	1M	AI 1	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO 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	--	--

Tabelle A-83 CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)

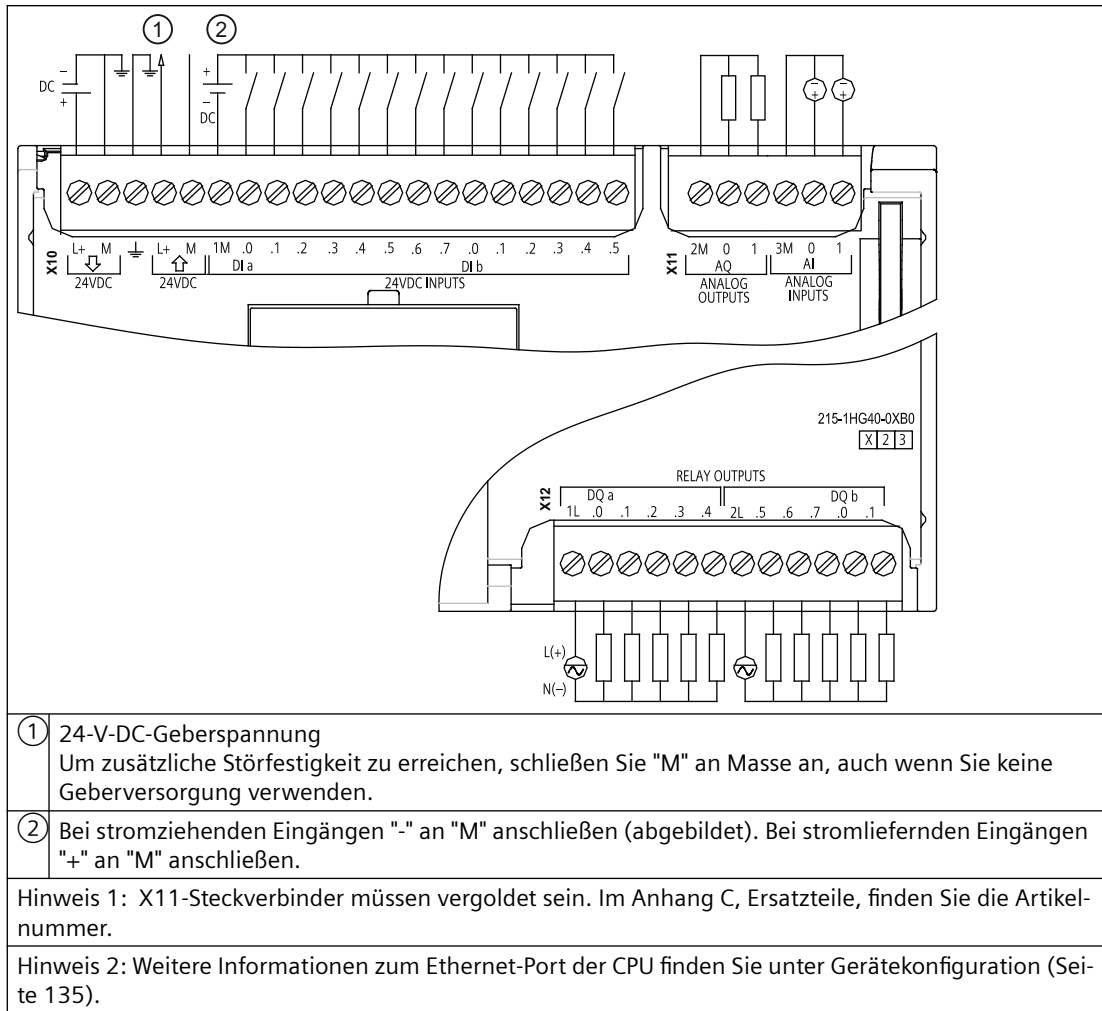


Tabelle A-84 Anschlussbelegung für die CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L+ / 24 V DC	2 M	1L
2	M / 24 V DC	AO 0	DO a.0
3	Funktionserde	AO 1	DO a.1
4	L+ / Geberversorgung 24 V DC	3M	DO a.2
5	M / Geberversorgung 24 V DC	AI 0	DO a.3
6	1M	AI 1	DO a.4
7	DI a.0	--	2L
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO b.1

Pin	X10	X11 (vergoldet)	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	--	--

Tabelle A-85 CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

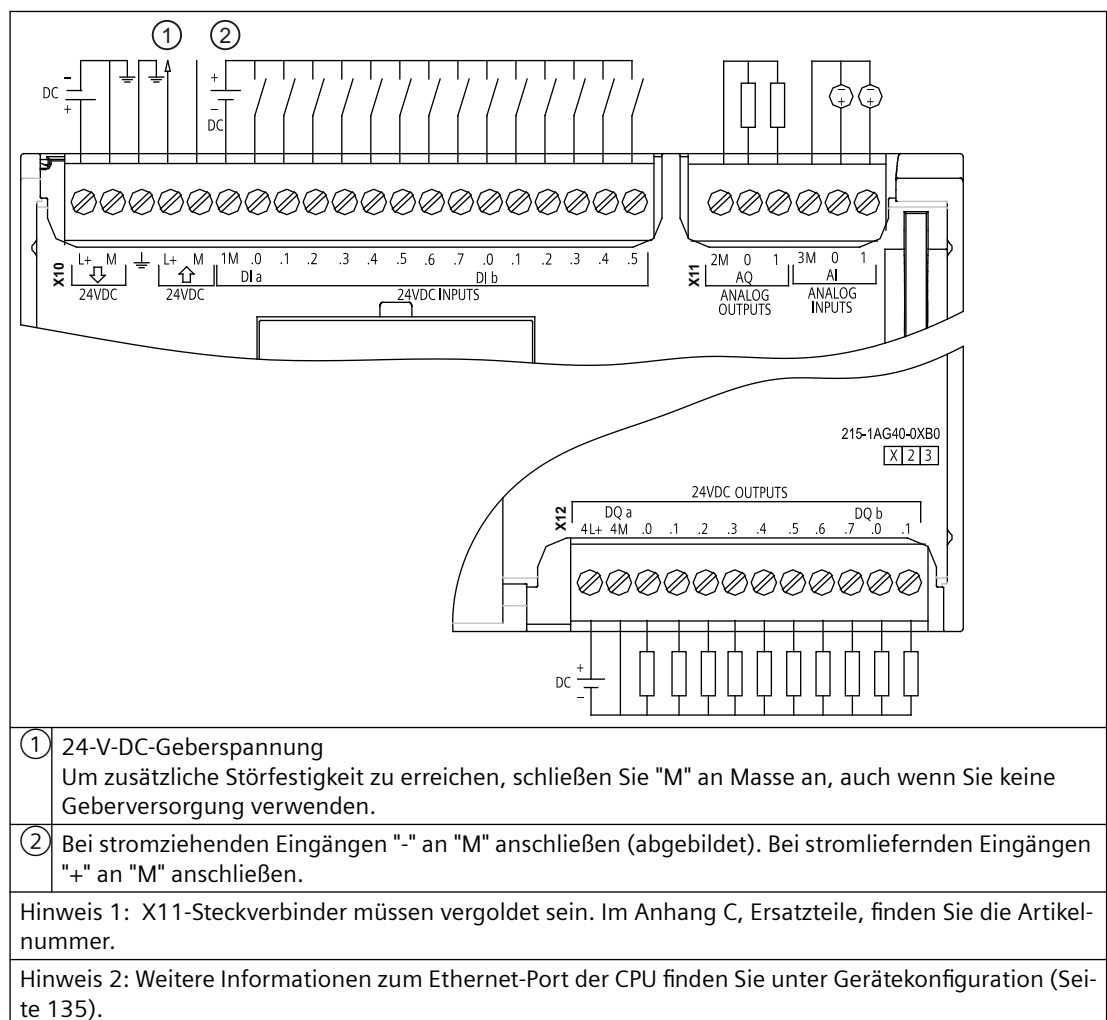


Tabelle A-86 Anschlussbelegung für die CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)

Pin	X10	X11 (vergoldet)	X12
1	L1 / 24 V DC	2 M	4L+
2	M / 24 V DC	AO 0	4M
3	Funktionserde	AO 1	DO a.0
4	L+ / Geberversorgung 24 V DC	3M	DO a.1
5	M / Geberversorgung 24 V DC	AI 0	DO a.2
6	1M	AI 1	DO a.3
7	DI a.0	--	DO a.4
8	DI a.1	--	DO a.5
9	DI a.2	--	DO a.6
10	DI a.3	--	DO a.7
11	DI a.4	--	DO b.0
12	DI a.5	--	DO 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	--	--

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

A.8 CPU 1217C**A.8.1 Allgemeine technische Daten und Leistungsmerkmale**

Tabelle A-87 Allgemein

Technische Daten	CPU 1217C DC/DC/DC
Artikelnummer	6ES7217-1AG40-0XB0
Abmessungen (B x H x T) (mm)	150 x 100 x 75
Versandgewicht	530 Gramm
Leistungsverlust	12 W
Verfügbarer elektrischer Strom (SM- und CM-Bus)	max. 1.600 mA (5 V DC)

Technische Daten	CPU 1217C DC/DC/DC
Verfügbare elektrischer Strom (24 V DC)	max. 400 mA (Geberversorgung)
Stromaufnahme digitaler Eingang (24 V DC)	4 mA/Eingang

Tabelle A-88 CPU-Merkmale

Technische Daten		Beschreibung
Anwenderspeicher (Siehe "Allgemeine technische Daten" (Seite 1253), "Speicherung im internen CPU-Speicher".)	Arbeitsspeicher	150 KB
	Ladespeicher	4 MB, intern, erweiterbar bis auf SD-Kartengröße
	Remanent	14 kByte
Integrierte digitale E/A		14 Eingänge/10 Ausgänge
Integrierte analoge E/A		2 Eingänge/2 Ausgänge
Größe des Prozessabbilds		1024 Bytes Eingänge (E)/1024 Bytes Ausgänge (A)
Merker (M)		8192 Byte
Temporärer (lokaler) Speicher		<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich der zugehörigen FBs und FCs) • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs)
Zusätzliche Signalmodule		max. 8 SMs
Erweiterung SB, CB, BB		max. 1
Zusätzliche Kommunikationsmodule		max. 3 CM
Schnelle Zähler		Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge (siehe HW-Konfigurationstabelle CPU 1217C Digitaler Eingang (DI)) (Seite 1320) <ul style="list-style-type: none"> • 1 MHz (Eb.2 bis Eb.5) • 100/180 kHz (Ea.0 bis Ea.5) • 30/120 kHz (Ea.6 bis Eb.1)
Impulsausgänge		Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge (siehe HW-Konfigurationstabelle CPU 1217C Digitaler Ausgang (DO)) (Seite 1320) <ul style="list-style-type: none"> • 1 MHz (Aa.0 bis Aa.3) • 100 kHz (Aa.4 bis Ab.1)
Eingänge für Impulsabgriff		14
Verzögerungsalarme		4 mit Auflösung von 1 ms
Weckalarne		4 mit Auflösung von 1 ms
Flankenalarne		12 steigend und 12 fallend (16 und 16 mit optionalem Signalboard)
Memory Card		SIMATIC Memory Card (optional)
Genauigkeit Echtzeituhr		+/- 60 Sekunden/Monat
Pufferung Echtzeituhr		Typ. 20 Tage/min. 12 Tage bei 40 °C (wartungsfreier Hochleistungskondensator)

¹ Die langsamere Geschwindigkeit gilt, wenn der HSC als A/B-Zähler konfiguriert ist.

Tabelle A-89 Leistung

Art der Anweisung		Ausführungsgeschwindigkeit	
		Direkte Adressierung (E, A und M)	DB-Zugriff
Boolescher Wert		0,08 µs/Anweisung	
Verschieben	Move_Bool	0,3 µs/Anweisung	1,17 µs/Anweisung
	Move_Word	0,137 µs/Anweisung	1,0 µs/Anweisung
	Move_Real	0,72 µs/Anweisung	1,0 µs/Anweisung
Realzahlarithmetik	Add Real	1,48 µs/Anweisung	1,78 µs/Anweisung

Hinweis

Viele Variablen wirken sich auf die gemessenen Zeiten aus. Die oben angegebenen Leistungszeiten gelten für die schnellsten Anweisungen in dieser Kategorie und fehlerfreie Programme.

A.8.2 Von der CPU 1217C unterstützte Zeiten, Zähler und Codebausteine

Tabelle A-90 Von der CPU 1217C unterstützte Bausteine, Zeiten und Zähler

Element		Beschreibung
Bausteine	Typ	OB, FB, FC, DB
	Größe	OB, FB, FC: 64 kBytes DB: bis zur Größe des Arbeitsspeichers
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Adressbereich für FBs, FCs und DBs	FB und FC: 1 bis 65535 (z. B. FB 1 bis FB 65535) DB: 1 bis 59999
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarmereignis-OBs ¹
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.

Element		Beschreibung
OBs	Programmzyklus	Mehrere
	Anlauf	Mehrere
	Verzögerungsalarme	4 (1 pro Ereignis)
	Weckalarne	4 (1 pro Ereignis)
	Prozessalarne	50 (1 pro Ereignis)
	Zeitfehleralarne	1
	Diagnosefehleralarne	1
	Ziehen oder Stecken von Modulen	1
	Fehler bei Baugruppenträger oder Station	1
	Uhrzeitalarm	Mehrere
	Status	1
	Update	1
	Profil	1
	MC-Interpolator	1
	MC-Servo	1
	MC-PreServo	1
MC-PostServo	1	
Zeiten	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Bytes pro Zeit
Zähler	Typ	IEC
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, Größe abhängig von der Zählart <ul style="list-style-type: none"> • SInt, USInt: 3 Bytes • Int, UInt: 6 Bytes • DInt, UDInt: 12 Bytes

¹ Sicherheitsprogramme verwenden zwei Schachtelungsebenen. Das Anwenderprogramm hat somit in Sicherheitsprogrammen eine Schachtelungstiefe von vier Ebenen.

Tabelle A-91 Kommunikation

Technische Daten	Beschreibung
Schnittstellen	2
Typ	Ethernet
HMI-Gerät	4
Programmiergerät (PG)	1
Anschlüsse	<ul style="list-style-type: none"> • 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRC • 8 CPU/CPU-Verbindungen (Client oder Server) für GET/PUT-Daten • 6 Verbindungen für dynamische Zuordnung zu GET/PUT oder offener Benutzerkommunikation • Max. 64 Verbindungen für Sicherheitszertifikate

Technische Daten	Beschreibung
Datenraten	10/100 MBit/s
Elektrische Trennung (externes Signal zu Logik)	Wandler potentialgetrennt, 1500 V AC (Typprüfung) ¹
Kabelart	CAT5e geschirmt
Schnittstellen	
Anzahl PROFINET-Schnittstellen	1
Anzahl PROFIBUS-Schnittstellen	0
Schnittstelle	
Hardware für Schnittstelle	
Anzahl Ports	2
Integrierter Switch	Ja
RJ-45 (Ethernet)	Ja; X1
Protokolle	
PROFINET IO-Controller	Ja
PROFINET IO-Device	Ja
SIMATIC-Kommunikation	Ja
Offene IE-Kommunikation	Ja
Webserver	Ja
Medienredundanz	Ja
PROFINET IO-Controller	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein
Offene IE-Kommunikation	Ja
IRT	Nein
MRP	Ja; als MRP-Client
PROFIenergy	Ja. Die S7-1200 CPU unterstützt nur die PROFIenergy-Einheit (mit I-Device-Funktionalität).
Priorisierter Anlauf	Ja (max. 16 PROFINET-Geräte)
Max. Anzahl anschließbarer IO-Devices	16
Max. Anzahl anschließbarer IO-Devices für RT	16
Max. Anzahl davon in Reihe	16
Max. Anzahl gleichzeitig aktivierbarer/deaktivierbarer IO-Devices	8
Aktualisierungszeiten	Der Mindestwert der Aktualisierungszeit ist auch abhängig von der für PROFINET IO eingestellten Kommunikationskomponente, der Anzahl von IO-Devices und der Menge der konfigurierten Anwenderdaten.
Mit RT	
Sendetakt von 1 ms	1 ms bis 512 ms
PROFINET IO-Device	
Dienste	
PG/OP-Kommunikation	Ja
S7-Routing	Ja
Taktsynchroner Betrieb	Nein

Technische Daten		Beschreibung
Offene IE-Kommunikation		Ja
IRT, unterstützt		Nein
MRP, unterstützt		Ja
PROFenergy		Ja
Shared Device		Ja
Max. Anzahl IO-Controller mit Shared Device		2
SIMATIC-Kommunikation		
S7-Kommunikation, als Server		Ja
S7-Kommunikation, als Client		Ja
Max. Anwenderdaten pro Auftrag		Siehe Online-Hilfe (S7-Kommunikation, Umfang der Anwenderdaten)
Offene IE-Kommunikation		
TCP/IP:		Ja
	Max. Länge der Daten	8 KB
	Unterstützung für mehrere passive Anschlüsse pro Port	Ja
ISO-on-TCP (RFC1006):		Ja
	Max. Länge der Daten	8 KB
UDP:		Ja
	Max. Länge der Daten	1472 Byte
DHCP		Nein
SNMP		Ja
DCP		Ja
LLDP		Ja

- ¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr aufgrund von gefährlichen Spannungen bei kurzfristigen Netzwerkstörungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

Tabelle A-92 Stromversorgung

Technische Daten		CPU 1217C DC/DC/DC
Spannungsbereich		20,4 V DC bis 28,8 V DC
Netzfrequenz		--
Eingangstrom (max. Last)	Nur CPU	600 mA bei 24 V DC
	CPU mit allen Erweiterungsbaugruppen	1600 mA bei 24 V DC
Einschaltstrom (max.)		12 A bei 28,8 V DC
I ² t		0,5 A ² s
Elektrische Trennung (Eingangsleistung zu Logik)		Nicht elektrisch getrennt
Verzögerungszeit (nach Spannungsverlust)		10 ms bei 24 V DC
Interne Sicherung, nicht durch Anwender austauschbar		3 A, 250 V, träge

Tabelle A-93 Gebersversorgung

Technische Daten	CPU 1217C DC/DC/DC
Spannungsbereich	L+ minus 4 V DC (min.)
Nennausgangsstrom (max.)	400 mA (kurzschlussfest)
Max. Welligkeit/Störströme (<10 MHz)	Wie Eingangsleitung
Elektrische Trennung (CPU-Logik zu Gebersversorgung)	Nicht elektrisch getrennt

A.8.3 Digitale Eingänge und Ausgänge

Tabelle A-94 Digitale Eingänge

Technische Daten	CPU 1217C DC/DC/DC
Anzahl der Eingänge	14: Gesamt: 10: Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend) 4: Differential (RS422/RS485)
Typ: Stromziehend/stromliefernd (IEC Typ 1 stromziehend)	Ea.0 bis Ea.7, Eb.0 bis Eb.1
Nennspannung	24 V DC bei 4 mA, nominal
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Filterzeiten	us-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
HSC-Eingangstaktfrequenzen (max.) (Pegel logisch 1 = 15 bis 26 V DC)	100/80 kHz (Ea.0 bis Ea.5) 30/20 kHz (Ea.6 bis Eb.1)
Typ: Differentialeingang (RS422/RS485)	Eb.2 bis Eb.5 (0,2+ 0,2- bis 0,5+ 0,5-)
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert (RS422/RS485-Charakteristik)
Integrierter Abschluss und Vorspannung	390 Ω bis 2M an Eb'-, 390 Ω bis +5 V an Eb'-, (Vorspannung AUS, wenn T/B spannungslos) 220 Ω zwischen Eb'+ und Eb'-
Eingangsimpedanz Empfänger	100 Ω einschließlich Vorspannung und Abschluss
Differentialempfänger Ansprechgrenze/Sensibilität	min. +/- 0,2 V, 60 mV typische Hysterese (RS422/RS485-Charakteristik)
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Filterzeiten	us-Einstellungen: 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0 ms-Einstellungen: 0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 1 MHz (Eb.2 bis Eb.5) A/B-Zähler: 1 MHz (Eb.2 bis Eb.5)

Technische Daten	CPU 1217C DC/DC/DC
Versatz Kanal-Kanal Differentialeingang	max. 40 ns
Allgemeine technische Daten (alle Digitaleingänge)	
Anzahl gleichzeitig eingeschalteter Eingänge	5 stromziehende/stromliefernde Eingänge (keine benachbarten Punkte) und 4 Differentialeingänge bei 60 °C horizontal oder 50 °C vertikal 14 bei 55 °C horizontal oder 45 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt 50 m geschirmt bei HSC-Eingängen (stromziehend/stromliefernd) 50 m geschirmt, verdreht bei allen Differentialeingängen

Tabelle A-95 HW-Konfigurationstabelle CPU 1217C Digitaleingang (DI)

Eingang	Typ und Frequenz
Dla.0	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.1	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.2	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.3	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.4	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.5	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 100 kHz
Dla.6	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 30 kHz
Dla.7	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 30 kHz
Dlb.0	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 30 kHz
Dlb.1	Typ: Eingang Typ 1, 24 V, stromziehend/stromliefernd Eingangsfrequenz schneller Zähler: max. 30 kHz
Dlb.2+ .2-	Typ: Differentialeingang RS422/RS485 Eingangsfrequenz schneller Zähler: max. 1 MHz
Dlb.3+ .3-	Typ: Differentialeingang RS422/RS485 Eingangsfrequenz schneller Zähler: max. 1 MHz
Dlb.4+ .4-	Typ: Differentialeingang RS422/RS485 Eingangsfrequenz schneller Zähler: max. 1 MHz
Dlb.5+ .5-	Typ: Differentialeingang RS422/RS485 Eingangsfrequenz schneller Zähler: max. 1 MHz

Tabelle A-96 Digitale Ausgänge

Technische Daten	CPU 1217C DC/DC/DC
Ausgänge	10 gesamt 6: MOSFET, elektronisch (stromliefernd) 4: Differential (RS422/RS485)
Typ: MOSFET, elektronisch (stromliefernder Ausgang)	Aa.4 bis Ab.1
Spannungsbereich	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	min. 20 V DC
Signal logisch 0 bei 10 k Ω Last	max. 0,1 V DC
Strom (max.)	0,5 A
Lampenlast	5 W
Widerstand bei EIN	max. 0,6 Ω
Kriechstrom pro Ausgang	max. 10 μ A
Einschaltstrom	8 A für max. 100 ms
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Induktive Klemmspannung	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.4 bis Ab.1)	max. 1,0 μ s von Aus nach Ein max. 3,0 μ s von Ein nach Aus
Maximale Schaltfrequenz Relais	--
Frequenz Impulsgenerator	max. 100 kHz (Aa.4 bis Ab.1) ¹ , min. 2 Hz
Typ: Differentialausgang (RS422/RS485)	Aa.0 bis Aa.3 (.0+ 0- bis .3+ .3-)
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert (RS422/RS485-Charakteristik)
Differentialausgangsspannung Sender	min. 2 V bei RL = 100 Ω , min. 1,5 V bei RL = 54 Ω (RS422/RS485-Charakteristik)
Integrierter Abschluss	100 Ω zwischen Aa'+ und Aa'-
Ausgangsimpedanz Treiber	100 Ω einschließlich Abschluss
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Schaltverzögerung (DAa.0 bis DAa.3)	max. 100 ns
Versatz Kanal-Kanal Differentialausgang	max. 40 ns
Frequenz Impulsgenerator	1 MHz (Aa.0 bis Aa.3), min. 2 Hz
Allgemeine technische Daten (alle Digitalausgänge)	
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja
Parallele Ausgänge für redundante Laststeuerung	Ja (nur Aa.4 bis Ab.1; mit demselben Bezugsleiter)
Parallele Ausgänge für erhöhte Last	Nein

Technische Daten	CPU 1217C DC/DC/DC
Anzahl gleichzeitig eingeschalteter Ausgänge	3 Ausgänge MOSFET, elektronisch (stromziehend) (keine benachbarten Punkte) und 4 Differentialausgänge bei 60 °C horizontal oder 50 °C vertikal 10 bei 55 °C horizontal oder 45 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt

¹ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

Tabelle A-97 HW-Konfigurationstabelle CPU 1217C Digitalausgang (DO)

Ausgang	Typ und Frequenz
DQa.0+ .0-	Typ: Differentialausgang RS422/RS485 Ausgangsfrequenz Impulsgenerator: max. 1 MHz, min. 2 Hz
DQa.1+ .1-	Typ: Differentialausgang RS422/RS485 Ausgangsfrequenz Impulsgenerator: max. 1 MHz, min. 2 Hz
DQa.2+ .2-	Typ: Differentialausgang RS422/RS485 Ausgangsfrequenz Impulsgenerator: max. 1 MHz, min. 2 Hz
DQa.3+ .3-	Typ: Differentialausgang RS422/RS485 Ausgangsfrequenz Impulsgenerator: max. 1 MHz, min. 2 Hz
DQa.4	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz
DQa.5	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz
DQa.6	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz
DQa.7	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz
DQb.0	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz
DQb.1	Typ: 24-V-stromliefernder Ausgang Ausgangsfrequenz Impulsgenerator: max. 100 kHz, min. 2 Hz

A.8.4 Analogeingänge und -ausgänge

A.8.4.1 Technische Daten der analogen Eingänge

Tabelle A-98 Analoge Eingänge

Technische Daten	Beschreibung
Anzahl der Eingänge	2
Typ	Spannung (Eintakteingang)
Vollausschlagsbereich	0 bis 10 V
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	10,001 bis 11,759 V

Technische Daten	Beschreibung
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	11,760 bis 11,852 V
Überlaufbereich (Datenwort)	32512 bis 32767
Auflösung	10 Bits
Max. Stehspannung	35 V DC
Glättung	Keine, schwach, mittel oder stark Siehe Tabelle der Schrittantwort (ms) für die analogen Eingänge der CPU (Seite 1324).
Rauschunterdrückung	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0% / 3,5% des Vollausschlags
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar

A.8.4.2 Schrittantwort der integrierten analogen Eingänge der CPU

Tabelle A-99 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Unterdrückungsfrequenz (Integrationszeit)		
	60 Hz	50 Hz	10 Hz
Keine (1 Zyklus): Keine Mittelwertbildung	50 ms	50 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	60 ms	70 ms	200 ms
Mittel (16 Zyklen): 16 Abtastwerte	200 ms	240 ms	1150 ms
Stark (32 Zyklen): 32 Abtastwerte	400 ms	480 ms	2300 ms
Abtastzeit	4,17 ms	5 ms	25 ms

A.8.4.3 Abtastzeit der integrierten analogen Ports der CPU

Tabelle A-100 Abtastzeit der integrierten analogen Eingänge der CPU

Unterdrückungsfrequenz (Auswahl Integrationszeit)	Abtastzeit
60 Hz (16,6 ms)	4,17 ms
50 Hz (20 ms)	5 ms
10 Hz (100 ms)	25 ms

A.8.4.4 Messbereiche der analogen Eingänge für Spannung (CPUs)

Tabelle A-101 Darstellung Analogeingang für Spannung (CPUs)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,852 V	Überlauf
32512	7F00		
32511	7EFF	11,759 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
34	22	12 mV	
0	0	0 V	
Negative Werte		Negative Werte werden nicht unterstützt	

A.8.4.5 Technische Daten der Analogausgänge

Tabelle A-102 Analoge Ausgänge

Technische Daten	Beschreibung
Ausgänge	2
Typ	Strom
Vollausschlagsbereich	0 ... 20 mA
Vollausschlag (Datenwort)	0 bis 27648
Überschwingbereich	20,01 bis 23,52 mA
Überschwingbereich (Datenwort)	27649 bis 32511
Überlaufbereich	siehe Fußnote ¹
Datenwort Überlaufbereich	32512 bis 32767
Auflösung	10 Bit
Ausgangsimpedanz	≤ 500 Ω max.
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	3,0 % / 3,5 % des Vollausschlags
Einschwingzeit	2 ms
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar

¹ Bei Überlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: "Ersatzwert aufschalten" oder "Letzten Wert halten".

Tabelle A-103 Darstellung Analogausgang für Strom (CPU 1215C und CPU 1217C)

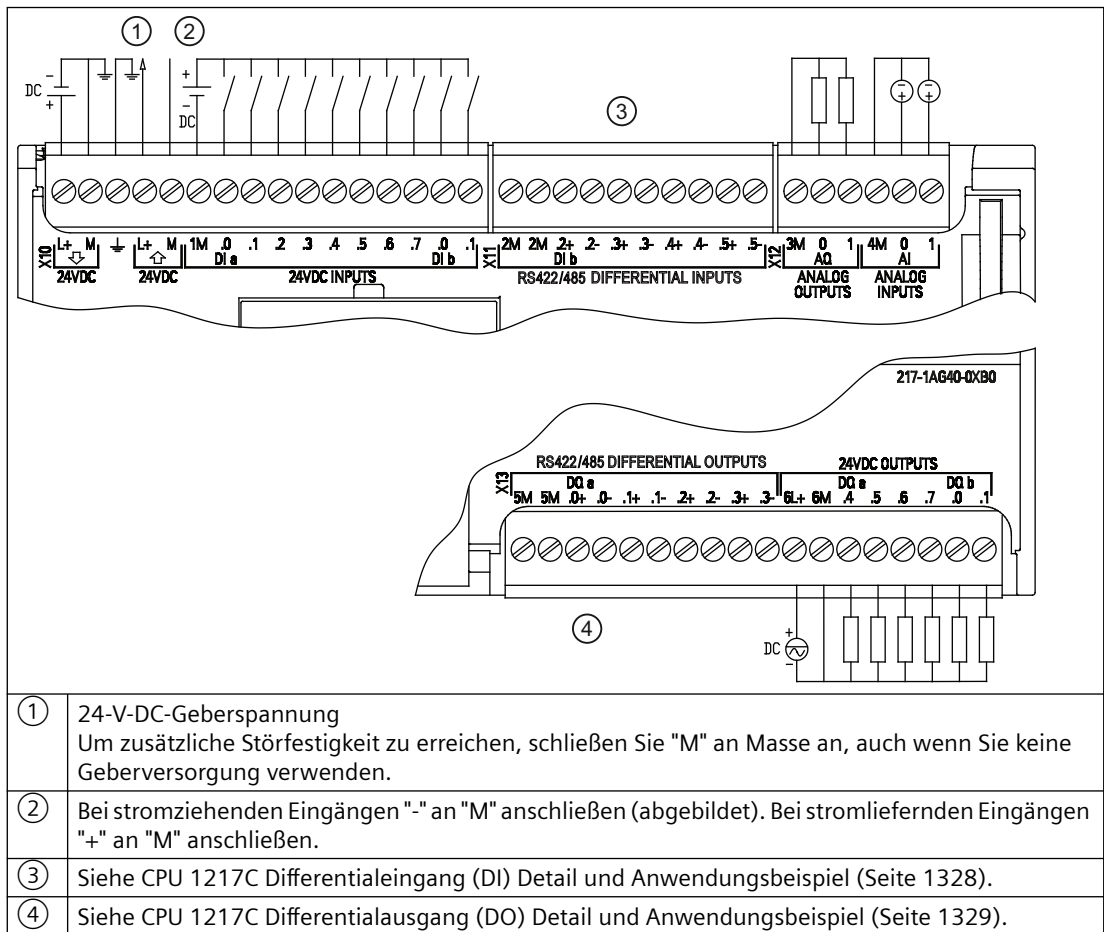
System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	

System		Stromausgangsbereich	
Dezimal	Hexadezimal	0 mA bis 20 mA	
32511	7EFF	23,52 mA	Überschwingbereich
27649	6C01		
27648	6C00	20 mA	Bemessungsbereich
20736	5100	15 mA	
34	22	0,0247 mA	
0	0	0 mA	
Negative Werte		Negative Werte werden nicht unterstützt	

¹ Bei Überlauf verhalten sich die Analoggänge entsprechend den eingestellten Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: "Ersatzwert aufschalten" oder "Letzten Wert halten".

A.8.5 Schaltpläne der CPU 1217C

Tabelle A-104 CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)



Hinweis 1: X12-Steckverbinder müssen vergoldet sein. In Anhang C, Ersatzteile (Seite 1446) finden Sie die Artikelnummer.

Hinweis 2: Weitere Informationen zum Ethernet-Port der CPU finden Sie unter Gerätekonfiguration (Seite 135).

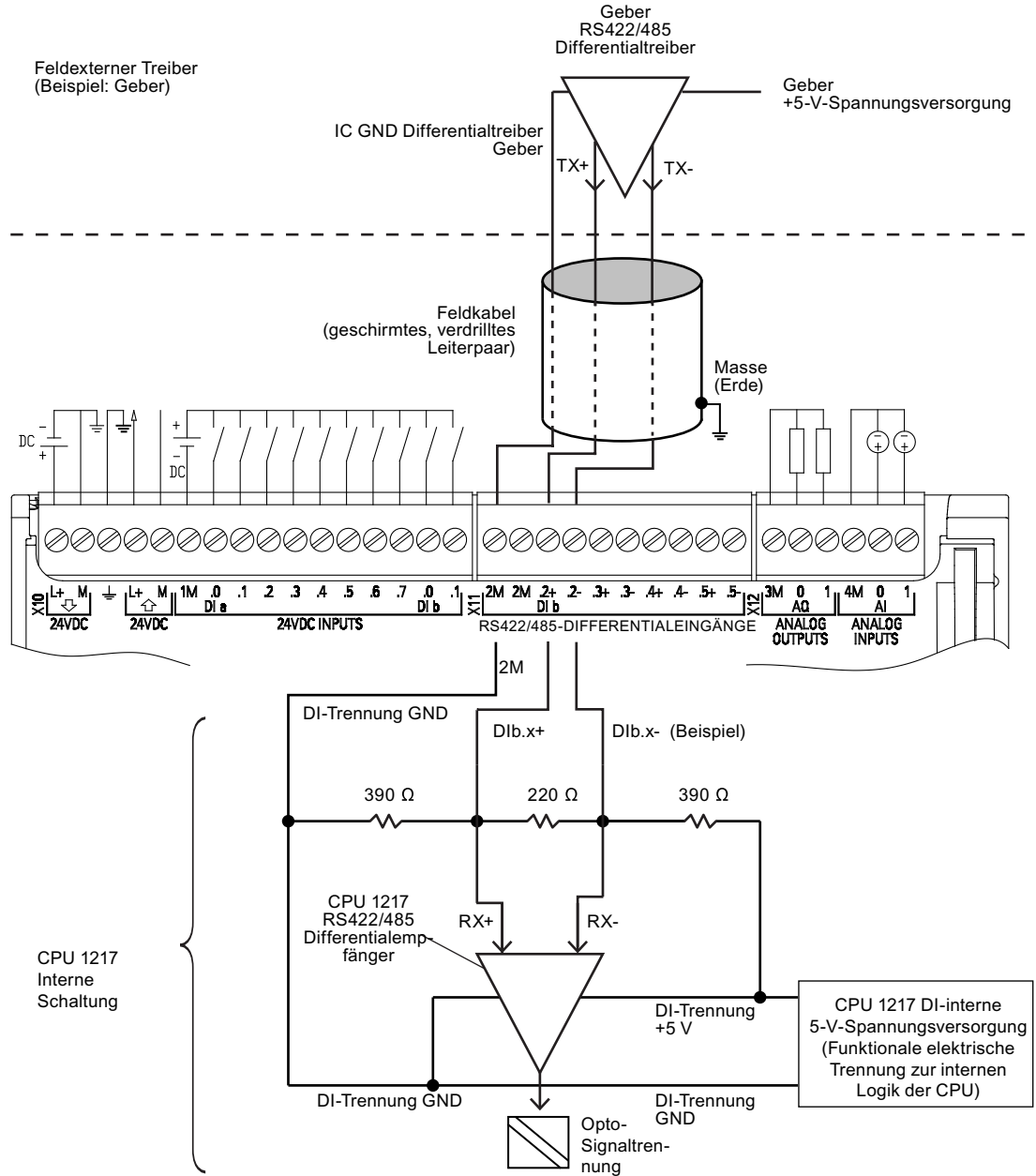
Tabelle A-105 Anschlussbelegung für die CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)

Pin	X10	X11	X12 (vergoldet)	X13
1	L+ / 24 V DC	2M	3M	5M
2	M / 24 V DC	2M	AO 0	5M
3	Funktionserde	DI b.2+	AO 1	DO a.0+
4	L+ / Geberversorgung 24 V DC	DI b.2-	4M	DO a.0-
5	M / Geberversorgung 24 V DC	DI b.3+	AI 0	DO a.1+
6	1M	DI b.3-	AI 1	DO a.1-
7	DI a.0	DI b.4+	--	DO a.2+
8	DI a.1	DI b.4-	--	DO a.2-
9	DI a.2	DI b.5+	--	DO a.3+
10	DI a.3	DI b.5-	--	DO a.3-
11	DI a.4	--	--	6L+
12	DI a.5	--	--	6M
13	DI a.6	--	--	DO a.4
14	DI a.7	--	--	DO a.5
15	DI b.0	--	--	DO a.6
16	DI b.1	--	--	DO a.7
17	--	--	--	DO b.0
18	--	--	--	DO b.1

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

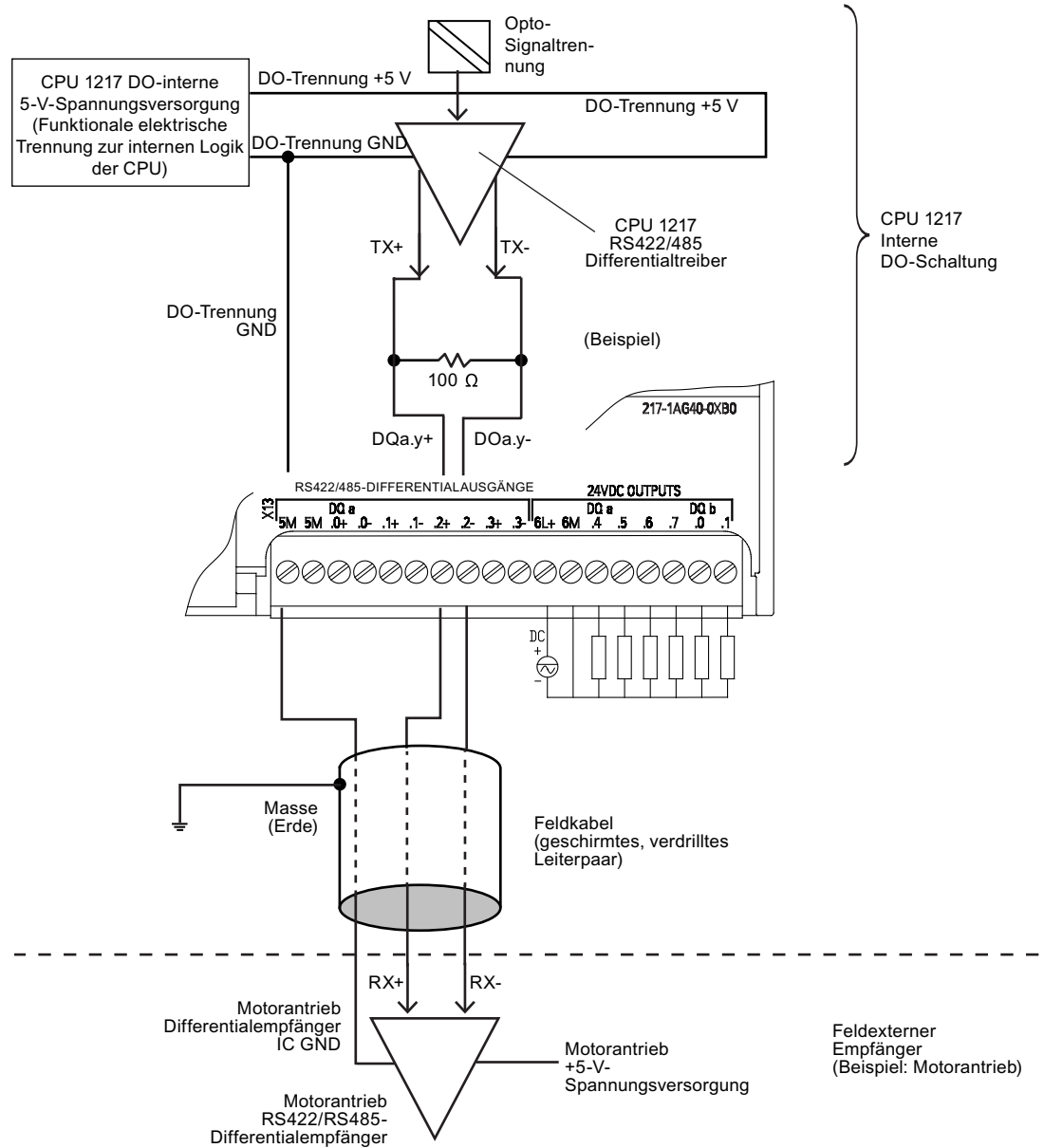
A.8.6 CPU 1217C Differentialeingang (DI) Detail und Anwendungsbeispiel



Hinweise

- Bei jedem digitalen Differentialeingang ist die Vorspannung AUS, wenn die Schraubklemmen im Klemmenblock offen sind.
- Integrierte(r) DI Abschluss und Vorspannung = 100 Ω Äquivalentimpedanz.
- Der Gleichtaktspannungsbereich wird durch den integrierten DI-Abschluss und Vorspannungswiderstände begrenzt. Einzelheiten finden Sie in den technischen Daten.

A.8.7 CPU 1217C Differentialausgang (DO) Detail und Anwendungsbeispiel



Hinweis

- Der Gleichtaktspannungsbereich wird durch den integrierten DO-Abschlusswiderstand begrenzt. Einzelheiten finden Sie in den technischen Daten.

A.9 Digitale Signalmodule (SMs)

A.9.1 Technische Daten für das digitale Eingangsmodul SM 1221

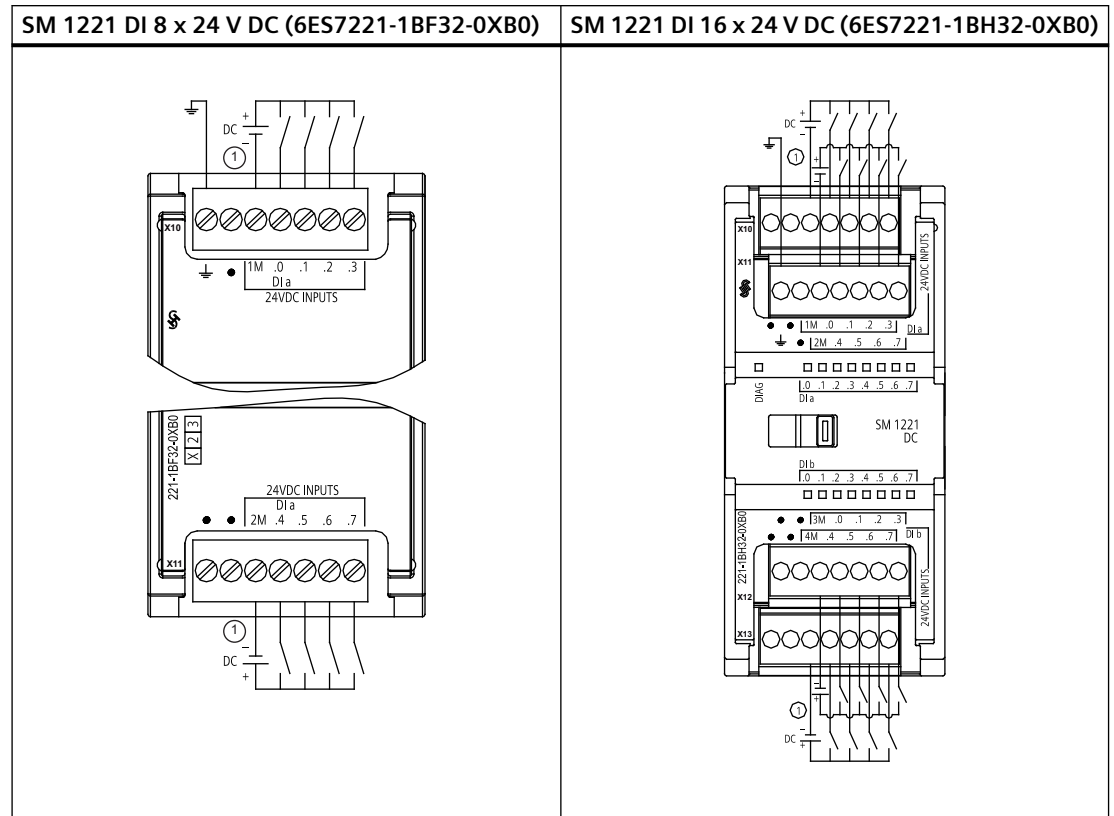
Tabelle A-106 Allgemeine technische Daten

Modell	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Artikelnummer	6ES7221-1BF32-0XB0	6ES7221-1BH32-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	
Gewicht	170 Gramm	210 Gramm
Leistungsverlust	1,5 W	2,5 W
Stromaufnahme (SM-Bus)	105 mA	130 mA
Stromaufnahme (24 V DC)	4 mA / Eingang	

Tabelle A-107 Digitale Eingänge

Modell	SM 1221 DI 8 x 24 V DC	SM 1221 DI 16 x 24 V DC
Anzahl der Eingänge	8	16
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)	
Nennspannung	24 V DC bei 4 mA, nominal	
Zulässige Dauerspannung	max. 30 V DC	
Stoßspannung	35 V DC für 0,5 s	
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	
Signal logisch 0 (max.)	5 V DC bei 1 mA	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	2	4
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)
Anzahl gleichzeitig eingeschalteter Eingänge	8	16
Leitungslänge (Meter)	500 m geschirmt, 300 m ungeschirmt	

Tabelle A-108 Schaltpläne der Digitaleingangs-SMs



① Bei stromziehenden Eingängen "-" an "M" anschließen (gezeigt). Bei stromliefernden Eingängen "+" an "M" anschließen.

Tabelle A-109 Anschlussbelegung für das SM 1221 DI 8 x 24 V DC (6ES7221-1BF32-0XB0)

Pin	X10	X11
1	Funktionserde	Kein Anschluss
2	Kein Anschluss	Kein Anschluss
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

Tabelle A-110 Anschlussbelegung für das SM 1221 DI 16 x 24 V DC (6ES7221-1BH32-0XB0)

Pin	X10	X11	X12	X13
1	Kein Anschluss	Funktionserde	Kein Anschluss	Kein Anschluss
2	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
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 Technische Daten für das digitale Ausgangsmodul SM 1222 mit 8 Ausgängen

Tabelle A-111 Allgemeine technische Daten

Modell	SM 1222 DO 8 x Relais	SM 1222 DO 8 x Relais (Umschaltung)	SM 1222 DQ 8 x 24 V DC
Artikelnummer	6ES7222-1HF32-0XB0	6ES7222-1XF32-0XB0	6ES7222-1BF32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	70 x 100 x 75	45 x 100 x 75
Gewicht	190 Gramm	310 Gramm	180 Gramm
Leistungsverlust	4,5 W	5 W	1,5 W
Stromaufnahme (SM-Bus)	120 mA	140 mA	120 mA
Stromaufnahme (24 V DC)	11 mA / Relaispule	16,7 mA / Relaispule	50 mA

Tabelle A-112 Digitale Ausgänge

Modell	SM 1222 DO 8 x Relais	SM 1222 DO 8 x Relais (Umschaltung)	SM 1222 DQ 8 x 24 V DC
Ausgänge	8	8	8
Typ	Relais, mechanisch	Relaiswechselkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	--	min. 20 V DC
Signal logisch 0 bei 10 k Ω Last	--	--	max. 0,1 V DC
Strom (max.)	2,0 A	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	30 W DC/200 W AC	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	--	--	max. 10 μ A
Einschaltstrom	7 A bei geschlossenen Kontakten	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein	
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule-Logik)	1500 V AC (Spule zu Kontakt)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	2	8	1
Strom je Leiter (max.)	10 A	2 A	4 A
Induktive Klemmspannung	--	--	L+ minus 48 V, 1 W Verlustleistung

Modell	SM 1222 DO 8 x Relais	SM 1222 DO 8 x Relais (Umschaltung)	SM 1222 DQ 8 x 24 V DC
Schaltverzögerung	max. 10 ms	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	1 Hz	--
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	10.000.000 Schaltspiele auf/zu	--
Lebensdauer der Kontakte bei Nennlast (Schließerkontakt)	100.000 Schaltspiele auf/zu	100.000 Schaltspiele auf/zu	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja		
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)		
Parallele Ausgänge für erhöhte Last	Nein		
Anzahl gleichzeitig eingeschalteter Ausgänge	8	<ul style="list-style-type: none"> • 4 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 8 bei 55 °C horizontal oder 45 °C vertikal 	8
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

A.9.3 Technische Daten für das digitale Ausgangsmodul SM 1222 mit 16 Ausgängen

Tabelle A-113 Allgemeine technische Daten

Modell	SM 1222 DO 16 x Relais	SM 1222 DQ 16 x 24 V DC	SM 1222 DQ 16 x 24 V DC stromziehend
Artikelnummer	6ES7222-1HH32-0XB0	6ES7222-1BH32-0XB0	6ES7222-1BH32-1XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Gewicht	260 Gramm	220 Gramm	220 Gramm
Leistungsverlust	8,5 W	2,5 W	2,5 W
Stromaufnahme (SM-Bus)	135 mA	140 mA	140 mA
Stromaufnahme (24 V DC)	11 mA / Relaispule	100 mA	40 mA

Tabelle A-114 Digitale Ausgänge

Modell	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 V DC	SM 1222 DQ 16 x 24 V DC stromziehend
Ausgänge	16	16	16
Typ	Relais, mechanisch	MOSFET, elektronisch (stromliefernd)	MOSFET, elektronisch (stromziehend)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	-	min. 20 V DC	0,5 V DC
Signal logisch 0 bei 10 k Ω Last	-	max. 0,1 V DC	24 V (typ.) minus 0,75 V DC
Strom (max.)	2,0 A	0,5 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω	max. 0,5 Ω
Kriechstrom pro Ausgang	--	max. 10 μ A	max. 75 μ A
Einschaltstrom	7 A bei geschlossenen Kon- takten	8 A für max. 100 ms	8 A für max. 100 ms
Überlastschutz	Nein		Ja, durch Strombegren- zung geschützter Bereich 1 A bis 3,5 A
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kon- takt) Keine (Spule-Logik)	707 V DC (Typprüfung)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	4	1	1
Strom je Leiter (max.)	10 A	8 A	Strombegrenzung
Induktive Klemmspannung	-	L+ minus 48 V, 1 W Ver- lustleistung	45 V
Schaltverzögerung	max. 10 ms	max. 50 μ s von Aus nach Ein max. 200 μ s von Ein nach Aus	max. 20 μ s von Aus nach Ein max. 350 μ s von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	-	-
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	-	-
Lebensdauer der Kontakte bei Nenn- last (Schließerkontakt)	100.000 Schaltspiele auf/zu	-	-
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatz- wert (Voreinstellung 0)	Letzter Wert oder Ersatz- wert (Voreinstellung 0)	Letzter Wert oder Ersatz- wert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja	Ja	Ja (M-lesend)
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)		
Parallele Ausgänge für erhöhte Last	Nein		

Modell	SM 1222 DQ 16 x Relais	SM 1222 DQ 16 x 24 V DC	SM 1222 DQ 16 x 24 V DC stromziehend
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> • 8 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 16 bei 55 °C horizontal oder 45 °C vertikal 	16	16
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt		

Siehe auch

Neue Funktionen (Seite 35)

Tabelle A-115 Schaltpläne der Digitalausgangs-SMs mit 8 Ausgängen

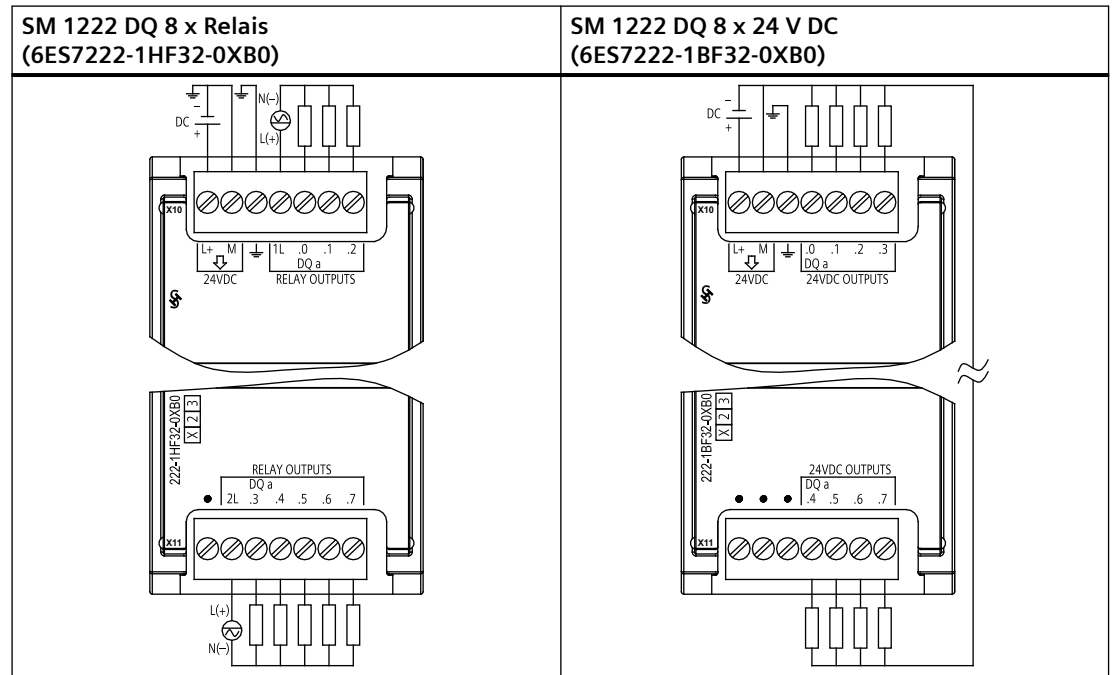


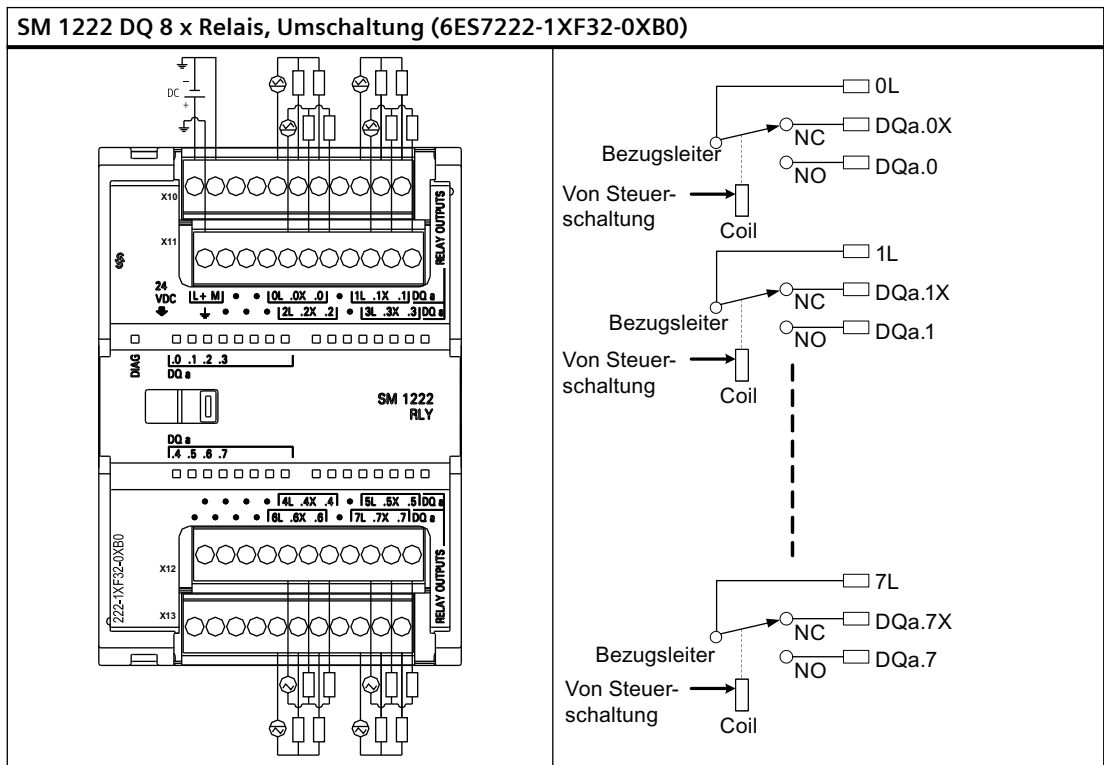
Tabelle A-116 Anschlussbelegung für das SM 1222 DQ 8 x Relais (6ES7222-1HF32-0XB0)

Pin	X10	X11
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	2L
3	Funktionserde	DO a.3
4	1L	DO a.4
5	DO a.0	DO a.5
6	DO a.1	DO a.6
7	DO a.2	DO a.7

Tabelle A-117 Anschlussbelegung für das SM 1222 DQ 8 x 24 V DC (6ES7222-1BF32-0XB0)

Pin	X10	X11
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
4	DO a.0	DO a.4
5	DO a.1	DO a.5
6	DO a.2	DO a.6
7	DO a.2	DO a.7

Tabelle A-118 Schaltplan für das Umschaltungs-Digitalausgangs-SM mit 8 Relaisausgängen



Ein Umschaltrelaisausgang steuert zwei Stromkreise über eine gemeinsame Klemme: einen Öffner- und einen Schließerkontakt. Am Beispiel von Ausgang "0" dargestellt, ist bei ausgeschaltetem Ausgang die gemeinsame Spannung (0L) mit dem Öffnerkontakt (.0X) verbunden und vom Schließerkontakt (.0) getrennt. Ist der Ausgang eingeschaltet, wird die Spannung (0L) vom Öffnerkontakt (.0X) getrennt und mit dem Schließerkontakt (.0) verbunden.

Tabelle A-119 Anschlussbelegung für das SM 1222 DQ 8 x Relais, Umschaltung (6ES7222-1XF32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss

Pin	X10	X11	X12	X13
4	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
5	0L	2L	4L	6L
6	DO a.0X	DO a.2X	DO a.4X	DO a.6X
7	DO a.0	DO a.2	DO a.4	DO a.6
8	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
9	1L	3L	5L	7L
10	DO a.1X	DO a.3X	DO a.5X	DO a.7X
11	DO a.1	DO a.3	DO a.5	DO a.7

Tabelle A-120 Schaltpläne der Digitalausgangs-SMs mit 16 Ausgängen

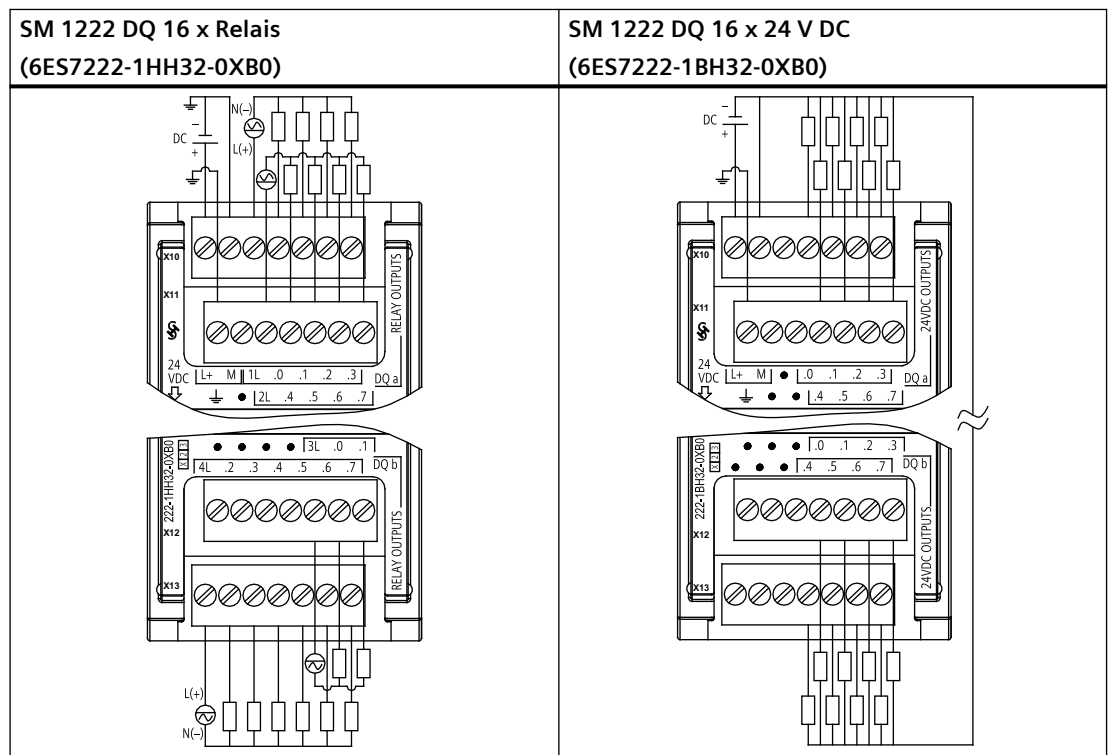


Tabelle A-121 Anschlussbelegung für das SM 1222 DQ 16 x Relais (6ES7222-1HH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	4L
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	DO b.2
3	1L	2L	Kein Anschluss	DO b.3
4	DO a.0	DO a.4	Kein Anschluss	DO b.4
5	DO a.1	DO a.5	3L	DO b.5
6	DO a.2	DO a.6	DO b.0	DO b.6
7	DO a.3	DO a.7	DO b.1	DO b.7

Tabelle A-122 Anschlussbelegung für das SM 1222 DQ 16 x 24 V DC (6ES7222-1BH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	DO a.0	DO a.4	DO b.0	DO b.4
5	DO a.1	DO a.5	DO b.1	DO b.5
6	DO a.2	DO a.6	DO b.2	DO b.6
7	DO a.3	DO a.7	DO b.3	DO b.7

Tabelle A-123 Schaltplan des Digitalausgangs-SM mit 16 Ausgängen, 24 V DC stromziehend

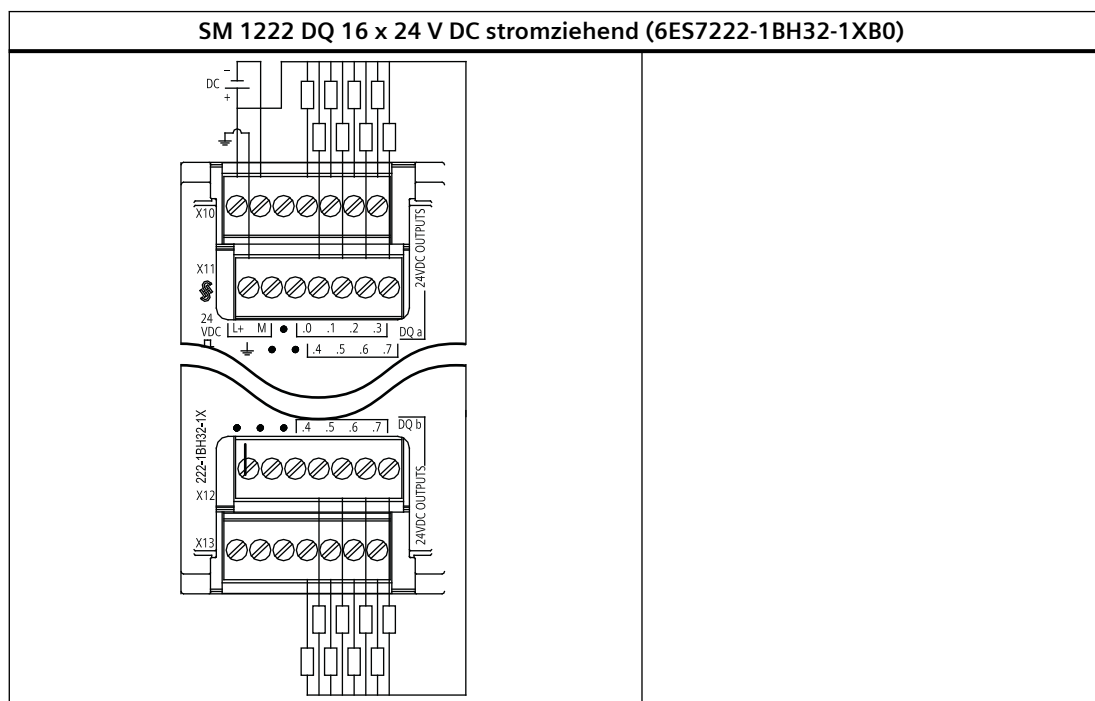


Tabelle A-124 Anschlussbelegung für das SM 1222 DQ 16 x 24 V DC stromziehend (6ES7222-1BH32-1XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	DO a.0	DO a.4	DO b.0	DO b.4
5	DO a.1	DO a.5	DO b.1	DO b.5
6	DO a.2	DO a.6	DO b.2	DO b.6
7	DO a.3	DO a.7	DO b.3	DQ b.7

A.9.4 Technische Daten für das digitale Eingabe-/Ausgabemodul SM 1223 (V DC)

Tabelle A-125 Allgemeine technische Daten

Modell	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	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 stromziehend
Artikelnummer	6ES7223-1PH32-0 XB0	6ES7223-1PL32-0 XB0	6ES7223-1BH32- 0XB0	6ES7223-1BL32-0 XB0	6ES7223-1BL32-1 XB0
Abmessungen B x H x T (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
Gewicht	230 Gramm	350 Gramm	210 Gramm	310 Gramm	310 Gramm
Leistungsverlust	5,5 W	10 W	2,5 W	4,5 W	4,5 W
Stromaufnahme (SM- Bus)	145 mA	180 mA	145 mA	185 mA	185 mA
Stromaufnahme (24 V DC)	4 mA / Eingang 11 mA / Relaisspule		150 mA	200 mA	40 mA

Tabelle A-126 Digitale Eingänge

Modell	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	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 stromzie- hend
Anzahl der Eingänge	8	16	8	16	16
Typ	Stromziehend/ stromliefernd (IEC Typ 1 strom- ziehend)	Stromziehend/ stromliefernd (IEC Typ 1 strom- ziehend)	Stromziehend/ stromliefernd (IEC Typ 1 strom- ziehend)	Stromziehend/ stromliefernd (IEC Typ 1 strom- ziehend)	Stromziehend/ stromliefernd (IEC Typ 1 strom- ziehend)
Nennspannung	24 V DC bei 4 mA, nominal	24 V DC bei 4 mA, nominal	24 V DC bei 4 mA, nominal	24 V DC bei 4 mA, nominal	24 V DC bei 4 mA, nominal
Zulässige Dauerspan- nung	max. 30 V DC	max. 30 V DC	max. 30 V DC	max. 30 V DC	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s	35 V DC für 0,5 s	35 V DC für 0,5 s	35 V DC für 0,5 s	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	15 V DC bei 2,5 mA	15 V DC bei 2,5 mA	15 V DC bei 2,5 mA	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA	5 V DC bei 1 mA	5 V DC bei 1 mA	5 V DC bei 1 mA	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprü- fung)	707 V DC (Typprü- fung)	707 V DC (Typprü- fung)	707 V DC (Typprü- fung)	707 V DC (Typprü- fung)
Potentialgetrennte Gruppen	2	2	2	2	2
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4

A.9 Digitale Signalmodule (SMs)

Modell	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	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 stromzie- hend
Anzahl gleichzeitig ein- geschalteter Eingänge	8	<ul style="list-style-type: none"> 8 (keine be- nachbarten Punkte) bei 60 °C horizontal oder 50 °C ver- tikal 16 bei 55 °C horizontal oder 45 °C ver- tikal 	8	16	16
Leitungslänge (Meter)	500 m geschirmt, 300 m unge- schirmt	500 m geschirmt, 300 m unge- schirmt	500 m geschirmt, 300 m unge- schirmt	500 m geschirmt, 300 m unge- schirmt	500 m geschirmt, 300 m unge- schirmt

Tabelle A-127 Digitale Ausgänge

Modell	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	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 stromziehend
Ausgänge	8	16	8	16	16
Typ	Relais, mecha- nisch	Relais, mecha- nisch	MOSFET, elektro- nisch (stromlie- fernd)	MOSFET, elektro- nisch (stromlie- fernd)	MOSFET, elektro- nisch (stromzie- hend)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC	20,4 bis 28,8 V DC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	--	--	min. 20 V DC	min. 20 V DC	0,5 V DC
Signal logisch 0 bei 10 kΩ Last	--	--	max. 0,1 V DC	max. 0,1 V DC	24 V (typ.) minus 0,75 V DC
Strom (max.)	2,0 A	2,0 A	0,5 A	0,5 A	0,5 A
Lampenlast	30 W DC/200 W AC	30 W DC/200 W AC	5 W	5 W	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,2 Ω wenn neu	max. 0,6 Ω	max. 0,6 Ω	max. 0,5 Ω
Kriechstrom pro Aus- gang	--	--	max. 10 μA	max. 10 μA	max. 75 μA
Einschaltstrom	7 A bei geschlosse- nen Kontakten	7 A bei geschlosse- nen Kontakten	8 A für max. 100 ms	8 A für max. 100 ms	Strombegrenzung
Überlastschutz	Nein	Nein	Nein	Nein	Ja, durch Strombe- grenzung ge- schützter Bereich 1 A bis 3,5 A

Modell	SM 1223 DI 8 x 24 V DC, DQ 8 x Relais	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais	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 stromziehend
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule-Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule-Logik)	707 V DC (Typprüfung)	707 V DC (Typprüfung)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	2	4	1	1	1
Strom je Leiter	10 A	8 A	4 A	8 A	8 A
Induktive Klemmspannung	--	--	L+ minus 48 V, 1 W Verlustleistung	L+ minus 48 V, 1 W Verlustleistung	45 V
Schaltverzögerung	max. 10 ms	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus	max. 20 µs von Aus nach Ein max. 350 µs von Ein nach Aus
Maximale Schaltfrequenz Relais	1 Hz	1 Hz	--	--	--
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	10.000.000 Schaltspiele auf/zu	--	--	--
Lebensdauer der Kontakte bei Nennlast (Schließerkontakt)	100.000 Schaltspiele auf/zu	100.000 Schaltspiele auf/zu	--	--	--
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja	Ja	Ja	Ja	Ja (M-lesend)
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)	Ja (mit demselben Bezugsleiter)	Ja (mit demselben Bezugsleiter)	Ja (mit demselben Bezugsleiter)	Ja (mit demselben Bezugsleiter)
Parallele Ausgänge für erhöhte Last	Nein	Nein	Nein	Nein	Nein
Anzahl gleichzeitig eingeschalteter Ausgänge	8	<ul style="list-style-type: none"> 8 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal 16 bei 55 °C horizontal oder 45 °C vertikal 	8	16	16
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt

Siehe auch

Neue Funktionen (Seite 35)

Tabelle A-128 Schaltpläne der digitalen Eingabe-/Ausgabemodule (V DC/Relais)

SM 1223 DI 8 x 24 V DC, DQ 8 x Relais (6ES7223-1PH32-0XB0)	SM 1223 DI 16 x 24 V DC, DQ 16 x Relais (6ES7223-1PL32-0XB0)	Hinweise
		<p>① Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.</p>

Tabelle A-129 Anschlussbelegung für das SM 1223 DI 8 x 24 V DC, DQ 8 x Relais (6ES7223-1PH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	1L	2L
4	DI a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

Tabelle A-130 Anschlussbelegung für das SM 1223 DI 16 x 24 V DC, DQ 16 x Relais (6ES7223-1PL32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	1L	3L
2	M / 24 V DC	Kein Anschluss	DO a.0	DO b.0
3	1M	2M	DO a.1	DO b.1
4	DI a.0	DI b.0	DO a.2	DO b.2

Pin	X10	X11	X12	X13
5	DI a.1	DI b.1	DO a.3	DO b.3
6	DI a.2	DI b.2	Kein Anschluss	Kein Anschluss
7	DI a.3	DI b.3	2L	4L
8	DI a.4	DI b.4	DO a.4	DO b.4
9	DI a.5	DI b.5	DO a.5	DO b.5
10	DI a.6	DI b.6	DO a.6	DO b.6
11	DI a.7	DI b.7	DO a.7	DO b.7

Tabelle A-131 Schaltpläne der digitalen Eingabe-/Ausgabemodule (V DC)

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)	Hinweise
		<p>① Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet).</p> <p>Bei stromliefernden Eingängen "+" an "M" anschließen.</p>

Tabelle A-132 Anschlussbelegung für das 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	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	Kein Anschluss	Kein Anschluss
4	DI a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

A.9 Digitale Signalmodule (SMs)

Tabelle A-133 Anschlussbelegung für das 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	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1M	2M	Kein Anschluss	Kein Anschluss
4	DI a.0	DI b.0	DO a.0	DO b.0
5	DI a.1	DI b.1	DO a.1	DO b.1
6	DI a.2	DI b.2	DO a.2	DO b.2
7	DI a.3	DI b.3	DO a.3	DO b.3
8	DI a.4	DI b.4	DO a.4	DO b.4
9	DI a.5	DI b.5	DO a.5	DO b.5
10	DI a.6	DI b.6	DO a.6	DO b.6
11	DI a.7	DI b.7	DO a.7	DO b.7

Tabelle A-134 Schaltplan der digitalen Eingabe-/Ausgabe-Signalmodule (V DC)

SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC stromziehend (6ES7223-1BL32-1XB0)	Hinweise
	<p>① Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.</p>

Tabelle A-135 Anschlussbelegung für das SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC stromziehend (6ES7223-1BL32-1XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss

Pin	X10	X11	X12	X13
3	1M	2M	Kein Anschluss	Kein Anschluss
4	DI a.0	DI b.0	DO a.0	DO b.0
5	DI a.1	DI b.1	DO a.1	DO b.1
6	DI a.2	DI b.2	DO a.2	DO b.2
7	DI a.3	DI b.3	DO a.3	DO b.3
8	DI a.4	DI b.4	DO a.4	DO b.4
9	DI a.5	DI b.5	DO a.5	DO b.5
10	DI a.6	DI b.6	DO a.6	DO b.6
11	DI a.7	DI b.7	DO a.7	DQ b.7

A.9.5 Technische Daten für das digitale Eingabe-/Ausgabemodul SM 1223 (V AC)

Tabelle A-136 Allgemeine technische Daten

Modell	SM 1223 DI 8 x120/230 V AC / DQ 8 x Relais
Artikelnummer	6ES7223-1QH32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75 mm
Gewicht	190 Gramm
Leistungsverlust	7,5 W
Stromaufnahme (SM-Bus)	120 mA
Stromaufnahme (24 V DC)	11 mA je Ausgang, wenn eingeschaltet

Tabelle A-137 Digitale Eingänge

Modell	SM 1223 DI 8 x 120/230 V AC / DQ 8 x Relais
Anzahl der Eingänge	8
Typ	IEC Typ 1
Nennspannung	120 V AC bei 6 mA, 230 V AC bei 9 mA
Zulässige Dauerspannung	264 V AC
Stoßspannung	--
Signal logisch 1 (min.)	79 V AC bei 2,5 mA
Signal logisch 0 (max.)	20 V AC bei 1 mA
Kriechstrom (max.)	1 mA
Elektrische Trennung (Feld zu Logik)	1500 V AC
Potentialgetrennte Gruppen ¹	4
Eingabeverzögerungszeiten	Typisch: 0,2 bis 12,8 ms, vom Anwender einstellbar Maximum: -
Anschluss 2Draht-Näherungssensor (Bero) (max.)	1 mA

A.9 Digitale Signalmodule (SMs)

Modell	SM 1223 DI 8 x 120/230 V AC / DQ 8 x Relais
Leitungslänge	Nicht geschirmt: 300 Meter Geschirmt: 500 Meter
Anzahl gleichzeitig eingeschalteter Eingänge	8

¹ Kanäle in einer Gruppe müssen die gleiche Phase haben.

Tabelle A-138 Digitale Ausgänge

Modell	SM 1223 DI 8 x 120/230 V AC / DQ 8 x Relais
Anzahl der Ausgänge	8
Typ	Relais, mechanisch
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC
Signal logisch 1 bei max. Strom	--
Signal logisch 0 bei 10 k Ω Last	--
Strom (max.)	2,0 A
Lampenlast	30 W DC/200 W AC
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu
Kriechstrom pro Ausgang	--
Einschaltstrom	7 A bei geschlossenen Kontakten
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	1500 V AC (Spule zu Kontakt) Keine (Spule-Logik)
Potentialgetrennte Gruppen	2
Strom je Leiter (max.)	10 A
Induktive Klemmspannung	--
Schaltverzögerung (max.)	10 ms
Maximale Schaltfrequenz Relais	1 Hz
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja
Parallele Ausgänge für redundante Laststeuerung	Ja (mit demselben Bezugsleiter)
Parallele Ausgänge für erhöhte Last	Nein
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> • 4 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 8 bei 55 °C horizontal oder 45 °C vertikal
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt

Tabelle A-139 SM 1223 DI 8 x 120/230 V AC, DQ 8 x Relais (6ES7223-1QH32-0XB0)

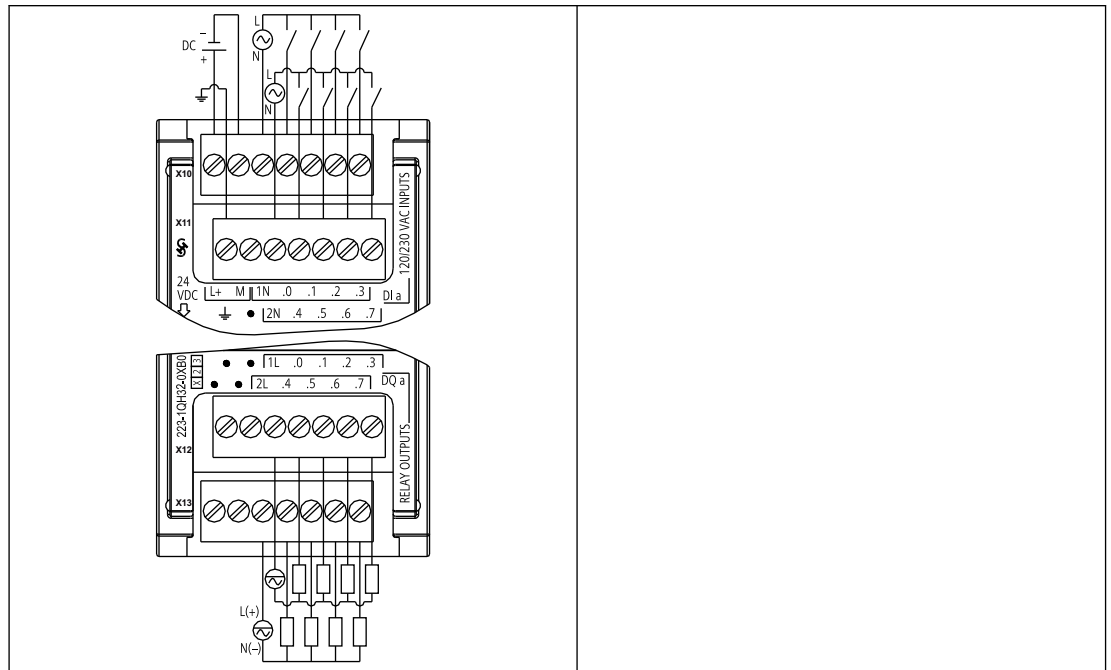


Tabelle A-140 Anschlussbelegung für das SM 1223 DI 8 x 120/240 V AC, DQ 8 x Relais (6ES7223-1QH32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Funktionserde	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	1N	2N	1L	2L
4	DI a.0	DI a.4	DO a.0	DO a.4
5	DI a.1	DI a.5	DO a.1	DO a.5
6	DI a.2	DI a.6	DO a.2	DO a.6
7	DI a.3	DI a.7	DO a.3	DO a.7

A.10 Analoge Signalmodule (SMs)

A.10.1 Technische Daten des SM 1231 Analogeingabemoduls

Tabelle A-141 Allgemeine technische Daten

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Artikelnummer	6ES7231-4HD32-0XB0	6ES7231-4HF32-0XB0	6ES7231-5ND32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75		

A.10 Analoge Signalmodule (SMs)

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Gewicht	180 Gramm		
Leistungsverlust	2,2 W	2,3 W	2,0 W
Stromaufnahme (SM-Bus)	80 mA	90 mA	80 mA
Stromaufnahme (24 V DC)	45 mA		65 mA

Tabelle A-142 Analoge Eingänge

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Anzahl der Eingänge	4	8	4
Typ	Spannung oder Strom (differential): wählbar in Gruppen zu je 2		Spannung oder Strom (differential)
Bereich	±10 V, ±5 V, ±2,5 V, 0 bis 20 mA oder 4 mA bis 20 mA		±10 V, ±5 V, ±2,5 V, ±1,25 V, 0 bis 20 mA oder 4 mA bis 20 mA
Vollausschlag (Datenwort)	-27648 bis 27648 Spannung / 0 bis 27648 Strom		
Überschwing-/Unterschwingbereich (Datenwort) Siehe Abschnitt über Analogeingangsbereiche für Spannung und Strom (Seite 1358).	Spannung: 32511 bis 27649 / -27649 bis -32512 Strom: 32511 bis 27649 / 0 bis -4864		
Überlauf/Unterlauf (Datenwort) Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 1358).	Spannung: 32767 bis 32512 / -32513 bis -32768 Strom 0 bis 20 mA: 32767 bis 32512 / -4865 bis -32768 Strom 4 bis 20 mA: 32767 bis 32512 (Werte unter -4864 weisen auf Drahtbruch hin)		
Auflösung ¹	12 Bit + Vorzeichenbit		15 Bit + Vorzeichenbit
Maximale Stehspannung/-strom	±35 V / ±40 mA		
Glättung	Keine, schwach, mittel oder stark Siehe Abschnitt zu Schrittantwortzeiten (Seite 1357).		
Rauschunterdrückung	400, 60, 50 oder 10 Hz Siehe Abschnitt zu Abtastraten (Seite 1357).		
Eingangsimpedanz Vor der Parametrierung Spannung Strom	≥ 1 MΩ ≥ 9 MΩ, FS 06 und höher ≥ 1 MΩ ≥ 270 Ω, < 290 Ω		≥ 1 MΩ ≥ 1 MΩ < 315 Ω, > 280 Ω
Elektrische Trennung Feld zu Logik Logik zu 24 V DC Feld zu 24 V DC Kanal zu Kanal	Keine		707 V DC (Typprüfung) 707 V DC (Typprüfung) 500 V DC (Typprüfung) Keine
Genauigkeit (25 °C / -20 bis 60 °C)	±0,1% / ±0,2% des Vollausschlags		±0,1% / ±0,3% des Vollausschlags
Messprinzip	Istwertumwandlung		
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz		

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Betriebssignalbereich ¹	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein		
Leitungslänge (Meter)	100 m, verdreht und geschirmt		

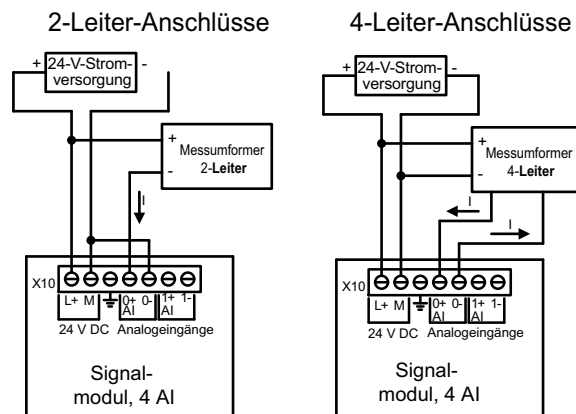
¹ Spannungen außerhalb des Betriebsbereichs, die an einem Kanal angelegt werden, können an anderen Kanälen Störungen verursachen.

Tabelle A-143 Diagnose

Modell	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Überlauf/Unterlauf	Ja		
24-V-DC-Niederspannung	Ja		
Drahtbruch	nur im Bereich 4 bis 20 mA (wenn der Eingang unter -4864 ist; 1,185 mA)		

SM 1231 Strommessung

Sie können die Strommessung mit einem 2- oder 4-Draht-Messumformer wie im folgenden Bild einrichten:



A.10 Analoge Signalmodule (SMs)

Tabelle A-144 Schaltpläne der Analogeingangs-SMs

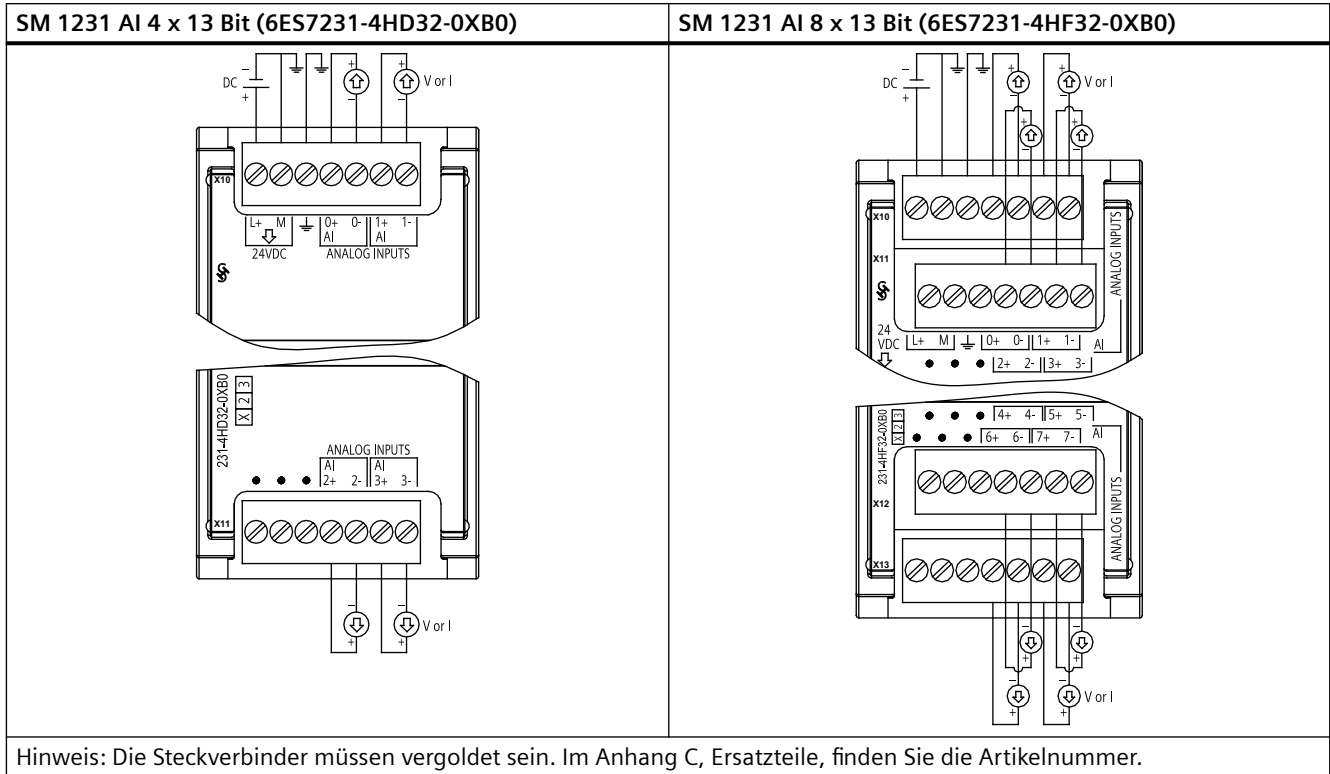


Tabelle A-145 Anschlussbelegung für das SM 1231 AI 4 x 13 Bit (6ES7231-4HD32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Tabelle A-146 Anschlussbelegung für das SM 1231 AI 8 x 13 Bit (6ES7231-4HF32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss	Kein Anschluss
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-

Tabelle A-147 Schaltplan des Analogeingangs-SMs

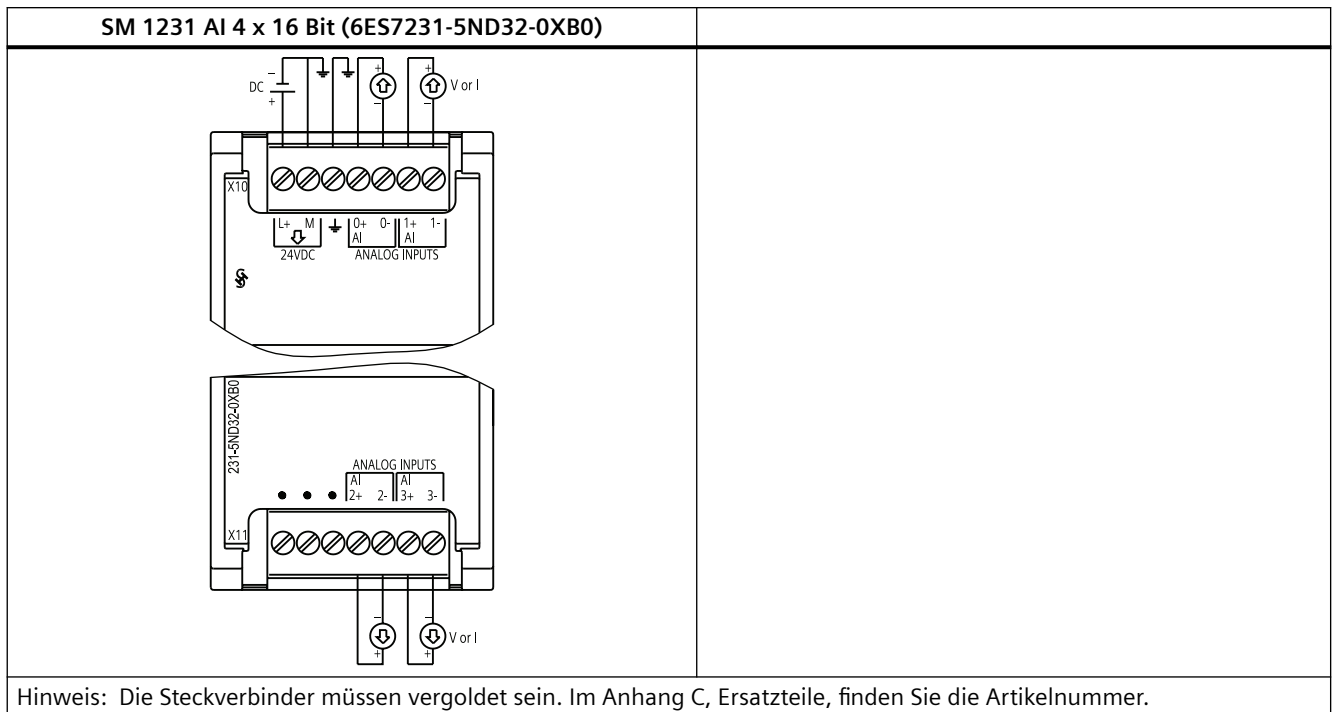


Tabelle A-148 Anschlussbelegung für das SM 1231 AI 4 x 16 Bit (6ES7231-5ND32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
4	AI 0+	AI 2+
5	AI 0-	AI 2-
6	AI 1+	AI 3+
7	AI 1-	AI 3-

Hinweis

Nicht verwendete Spannungseingangskanäle sollten kurzgeschlossen werden.

Nicht verwendete Stromeingangskanäle sollten auf den Bereich 0 bis 20 mA gesetzt werden und/oder das Melden von Drahtbruch sollte deaktiviert werden.

Für den Strommodus konfigurierte Eingänge führen nur dann Schleifenstrom, wenn das Modul eingeschaltet und konfiguriert ist.

Stromeingangskanäle sind nur dann betriebsfähig, wenn der Sender mit externer Spannung versorgt wird.

A.10.2 Technische Daten des SM 1232 Analogausgabemoduls

Tabelle A-149 Allgemeine technische Daten

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Artikelnummer	6ES7232-4HB32-0XB0	6ES7232-4HD32-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75	
Gewicht	180 Gramm	
Leistungsverlust	1,8 W	2,0 W
Stromaufnahme (SM-Bus)	80 mA	
Stromaufnahme (24 V DC)	45 mA (ohne Last)	

Tabelle A-150 Analoge Ausgänge

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Ausgänge	2	4
Typ	Spannung oder Strom	
Bereich	±10 V, 0 bis 20 mA oder 4 mA bis 20 mA	
Auflösung	Spannung: 14 Bit Strom: 13 Bits	
Vollausschlag (Datenwort)	Spannung: -27.648 bis 27.648; Strom: 0 bis 27.648 Siehe Abschnitt über Ausgangsbereiche für Spannung und Strom (Seite 1359).	
Genauigkeit (25 °C / -20 bis 60 °C)	±0,3 % / ±0,6 % des Vollausschlags	
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µs (R), 750 µs (1 uF) Strom: 600 µs (1 mH), 2 ms (10 mH)	
Lastimpedanz	Spannung: ≥ 1000 Ω Strom: ≤ 600 Ω	
Maximale Ausgabe Kurzschlussstrom	Spannungsmodus: ≤ 24 mA Strommodus: ≥ 38,5 mA	
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Elektrische Trennung (Feld zu Logik)	Keine	
Elektrische Trennung (24 V zu Ausgang)	Keine	
Leitungslänge (Meter)	100 m, verdreht und geschirmt	

Tabelle A-151 Diagnose

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Überlauf/Unterlauf	Ja	
Erdschluss (nur Spannungsmodus)	Ja	
Drahtbruch (nur Strommodus) ¹	Ja	
24 V DC Niederspannung ²	Ja	

¹ Kurzschlusserkennung ist nur möglich, wenn die Ausgangsspannung kleiner als -0,5 V oder größer als +0,5 V ist.

² Drahtbrucherkennung ist nur möglich, wenn der Ausgangsstrom größer als 1 mA ist.

Tabelle A-152 Schaltpläne der Analogausgangs-SMs

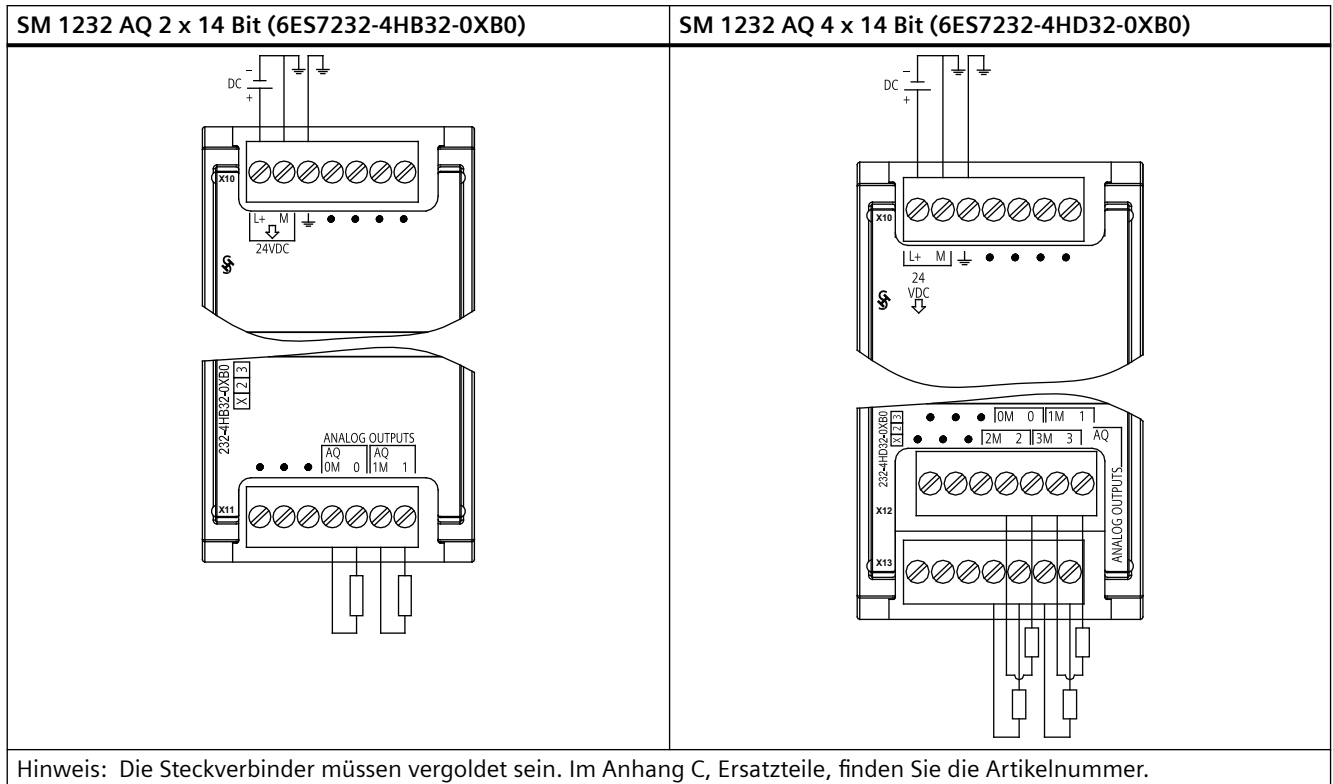


Tabelle A-153 Anschlussbelegung für das SM 1232 AQ 2 x 14 Bit (6ES7232-4HB32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
4	Kein Anschluss	AO 0M
5	Kein Anschluss	AO 0
6	Kein Anschluss	AO 1M
7	Kein Anschluss	AO 1

Tabelle A-154 Anschlussbelegung für das SM 1232 AQ 4 x 14 Bit (6ES7232-4HD32-0XB0)

Pin	X10 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss
4	Kein Anschluss	AO 0M	AO 2M
5	Kein Anschluss	AO 0	AO 2

Pin	X10 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
6	Kein Anschluss	AO 1M	AO 3M
7	Kein Anschluss	AO 1	AIO 3

A.10.3 Technische Daten des SM 1234 Analogein-/Analogausgabemoduls

Tabelle A-155 Allgemeine technische Daten

Technische Daten	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Artikelnummer	6ES7234-4HE32-0XB0
Abmessungen (B x H x T) (mm)	45 x 100 x 75
Gewicht	220 Gramm
Leistungsverlust	2,4 W
Stromaufnahme (SM-Bus)	80 mA
Stromaufnahme (24 V DC)	60 mA (ohne Last)

Tabelle A-156 Analoge Eingänge

Modell	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Anzahl der Eingänge	4
Typ	Spannung oder Strom (differential): wählbar in Gruppen zu je 2
Bereich	± 10 V, ± 5 V, $\pm 2,5$ V, 0 bis 20 mA oder 4 mA bis 20 mA
Vollausschlag (Datenwort)	-27648 bis 27648
Überschwing-/Unterschwingbereich (Datenwort)	Spannung: 32511 bis 27649 / -27649 bis -32512 Strom: 32511 bis 27649 / 0 bis -4864 Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 1358).
Überlauf/Unterlauf (Datenwort)	Spannung: 32767 bis 32512 / -32513 bis -32768 Strom: 32767 bis 32512 / -4865 bis -32768 Siehe Abschnitt über Eingangsbereiche für Spannung und Strom (Seite 1358).
Auflösung	12 Bit + Vorzeichenbit
Maximale Stehspannung/-strom	± 35 V / ± 40 mA
Glättung	Keine, schwach, mittel oder stark Siehe Abschnitt zu Schrittantwortzeiten (Seite 1357).
Rauschunterdrückung	400, 60, 50 oder 10 Hz Siehe Abschnitt zu Abtastraten (Seite 1357).
Eingangsimpedanz	≥ 9 M Ω , FS 07 und höher ≥ 1 M Ω (Spannung) / ≥ 270 Ω , < 290 Ω (Strom)
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / -20 bis 60 °C)	$\pm 0,1\%$ / $\pm 0,2\%$ des Vollausschlags
Analog-Digital-Umsetzzeit	625 μ s (400 Hz Unterdrückung)
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz

Modell	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Betriebssignalebereich ¹	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein
Leitungslänge (Meter)	100 m, verdreht und geschirmt

¹ Spannungen außerhalb des Betriebsbereichs, die an einem Kanal angelegt werden, können an anderen Kanälen Störungen verursachen.

Tabelle A-157 Analoge Ausgänge

Technische Daten	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Ausgänge	2
Typ	Spannung oder Strom
Bereich	±10 V oder 0 bis 20 mA oder 4 mA bis 20 mA
Auflösung	Spannung: 14 Bits, Strom: 13 Bits
Vollausschlag (Datenwort)	Spannung: -27.648 bis 27.648; Strom: 0 bis 27648 Siehe Abschnitt über Ausgangsbereiche für Spannung und Strom (Seite 1359).
Genauigkeit (25 °C / -20 bis 60 °C)	±0,3 % / ±0,6 % des Vollausschlags
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µs (R), 750 µs (1 µF) Strom: 600 µs (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: ≥ 1000 Ω Strom: ≤ 600 Ω
Maximale Ausgabe Kurzschlussstrom	Spannungsmodus: ≤ 24 mA Strommodus: ≥ 38,5 mA
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine
Elektrische Trennung (24 V zu Ausgang)	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt

Tabelle A-158 Diagnose

Modell	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Überlauf/Unterlauf	Ja
Erdschluss (nur Spannungsmodus) ¹	Ja an Ausgängen
Drahtbruch (nur Strommodus) ²	Ja an Ausgängen
24-V-DC-Niederspannung	Ja

¹ Kurzschlusserkennung ist nur möglich, wenn die Ausgangsspannung kleiner als -0,5 V oder größer als +0,5 V ist.

² Drahtbruchererkennung ist nur möglich, wenn der Ausgangsstrom größer als 1 mA ist.

SM 1234 Strommessung

Sie können die Strommessung mit einem 2- oder 4-Draht-Messumformer wie im folgenden Bild einrichten:

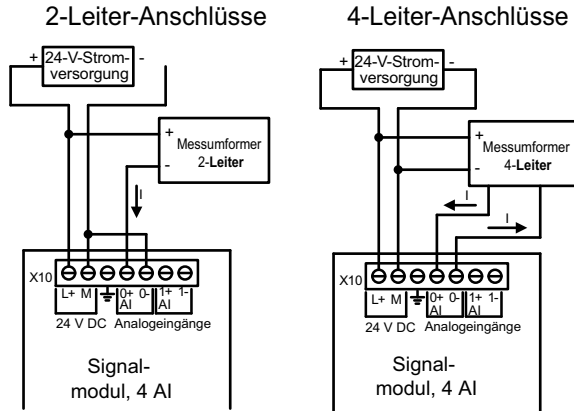


Tabelle A-159 Schaltpläne des analogen Ein-/Ausgangs-SMs

SM 1234 AI 4 x 13 Bit / AQ 2 x 14 Bit (6ES7234-4HE32-0XB0)	
<p>Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.</p>	

Tabelle A-160 Anschlussbelegung für das SM 1234 AI 4 x 13 Bit / AQ 2 x 14 Bit (6ES7234-4HE32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss

Pin	X10 (vergoldet)	X11 (vergoldet)	X13 (vergoldet)
4	AI 0+	AI 2+	AO 0M
5	AI 0-	AI 2-	AO 0
6	AI 1+	AI 3+	AO 1M
7	AI 1-	AI 3-	AO 1

Hinweis

Nicht verwendete Spannungseingangskanäle sollten kurzgeschlossen werden.

Nicht verwendete Stromeingangskanäle sollten auf den Bereich 0 bis 20 mA gesetzt werden und/oder das Melden von Drahtbruch sollte deaktiviert werden.

Für den Strommodus konfigurierte Eingänge führen nur dann Schleifenstrom, wenn das Modul eingeschaltet und konfiguriert ist.

Stromeingangskanäle sind nur dann betriebsfähig, wenn der Sender mit externer Spannung versorgt wird.

A.10.4 Schrittantwort der analogen Eingänge

Tabelle A-161 Schrittantwort (ms), 0 bis Vollausschlag, gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Rauschminderung/Unterdrückungsfrequenz (Auswahl Integrationszeit)			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Keine (1 Zyklus): Keine Mittelwertbildung	4 ms	18 ms	22 ms	100 ms
Schwach (4 Zyklen): 4 Abtastwerte	9 ms	52 ms	63 ms	320 ms
Mittel (16 Zyklen): 16 Abtastwerte	32 ms	203 ms	241 ms	1200 ms
Stark (32 Zyklen): 32 Abtastwerte	61 ms	400 ms	483 ms	2410 ms

A.10.5 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A-162 Abtast- und Modulaktualisierungszeiten für alle Kanäle

Unterdrückungsfrequenz (Integrationszeit)	Abtast- und Modulaktualisierungszeiten für alle Kanäle			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
SM 4-Kanal x 13 Bit	0,625 ms	4,17 ms	5 ms	25 ms
SM 8-Kanal x 13 Bit	1,25 ms	4,17 ms	5 ms	25 ms
SM 4-Kanal x 16 Bit	0,417 ms	0,397 ms	0,400 ms	0,400 ms

A.10.6 Messbereiche der analogen Eingänge für Spannung und Strom (SB und SM)

Tabelle A-163 Darstellung Analogeingang für Spannung (SB und SM)

System		Messbereich Spannung				
Dezimal	Hexadezimal	±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	Überlauf
32512	7F00					
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Überschwingbereich
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Bemessungsbereich
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					Unterschwingbereich
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	
-32513	80FF					Unterlauf
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

¹ 7FFF kann aus einem der folgenden Gründe zurückgegeben werden: Überlauf (siehe Tabelle), bevor gültige Werte vorliegen (zum Beispiel unmittelbar beim Hochfahren) oder wenn ein Drahtbruch erkannt wird.

Tabelle A-164 Darstellung Analogeingang für Strom (SB und SM)

System		Messbereich Strom		
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 mA	
32767	7FFF	> 23,52 mA	> 22,81 mA	Überlauf
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Nennbereich
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Unterschwingbereich
-4864	ED00	-3,52 mA	1,185 mA	
32767 ¹	7FFF		< 1,185 mA	Drahtbruch (4 bis 20 mA)
-32768	8000	< -3,52 mA		Unterlauf (0 bis 20 mA)

¹ Der Drahtbruchwert von 32767 (16#7FFF) wird unabhängig vom Zustand des Drahtbruchalarms immer zurückgegeben.

Siehe auch

Ermitteln der Art eines Drahtbruchs über ein Modul SM 1231 (Seite 1244)

A.10.7 Messbereiche der analogen Ausgänge für Spannung und Strom (SB und SM)

Tabelle A-165 Darstellung Analogausgang für Spannung (SB und SM)

System		Spannungsausgangsbereich		
Dezimal	Hexadezimal	±10 V		
32767	7FFF	Siehe Hinweis 1	Überlauf	
32512	7F00	Siehe Hinweis 1		
32511	7EFF	11,76 V	Überschwingbereich	
27649	6C01			
27648	6C00	10 V	Bemessungsbereich	
20736	5100	7,5 V		
1	1	361,7 µV		
0	0	0 V		
-1	FFFF	-361,7 µV		
-20736	AF00	-7,5 V		
-27648	9400	-10 V		
-27649	93FF			Unterschwingbereich
-32512	8100	-11,76 V		
-32513	80FF	Siehe Hinweis 1		Unterlauf
-32768	8000	Siehe Hinweis 1		

¹ Bei Überlauf oder Unterlauf nehmen die Analogausgänge den Ersatzwert für den Betriebszustand STOP an.

Tabelle A-166 Darstellung Analogausgang für Strom (SB und SM)

System		Stromausgangsbereich			
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 mA		
32767	7FFF	Siehe Hinweis 1	Siehe Hinweis 1	Überlauf	
32512	7F00	Siehe Hinweis 1	Siehe Hinweis 1		
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich	
27649	6C01				
27648	6C00	20 mA	20 mA	Bemessungsbereich	
20736	5100	15 mA	16 mA		
1	1	723,4 nA	4 mA + 578,7 nA		
0	0	0 mA	4 mA		
-1	FFFF		4 mA bis 578,7 nA		Unterschwingbereich
-6912	E500		0 mA		
-6913	E4FF			Nicht möglich. Ausgangswert auf 0 mA begrenzt.	
-32512	8100				
-32513	80FF	Siehe Hinweis 1	Siehe Hinweis 1	Unterlauf	
-32768	8000	Siehe Hinweis 1	Siehe Hinweis 1		

¹ Bei Überlauf oder Unterlauf nehmen die Analogausgänge den Ersatzwert für den Betriebszustand STOP an.

A.11 Thermoelement- und RTD-Signalmodule (SMs)

A.11.1 SM 1231 Thermoelement

Tabelle A-167 Allgemeine technische Daten

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Artikelnummer	6ES7231-5QD32-0XB0	6ES7231-5QF32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	
Gewicht	180 Gramm	190 Gramm
Leistungsverlust	1,5 W	
Stromaufnahme (SM-Bus)	80 mA	
Stromaufnahme (24 V DC) ¹	40 mA	

¹ 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus PLC)

Tabelle A-168 Analoge Eingänge

Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Anzahl der Eingänge	4	8
Bereich Nennbereich (Datenwort) Überlauf/Unterlauf (Datenwort) Überlauf/Unterlauf (Datenwort)	Siehe Thermoelement-Auswahltabelle (Seite 1363).	
Auflösung	Temperaturbereich	0,1 °C / 0,1 °F
	Spannung	15 Bit plus Vorzeichen
Max. Stehspannung	±35 V	
Rauschunterdrückung	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)	
Gleichtaktunterdrückung	> 120 dB bei 120 V AC	
Impedanz	≥ 10 MΩ	
Elektrische Trennung	Feld zu Logik	707 V DC (Typprüfung)
	Feld an 24 V DC	707 V DC (Typprüfung)
	24 V DC an Logik	707 V DC (Typprüfung)
Kanal zu Kanal	120 V AC	
Genauigkeit	Siehe Thermoelement-Auswahltabelle (Seite 1363).	
Wiederholgenauigkeit	±0,05 % Vollausschlag	
Messprinzip	Integrierend	
Aktualisierungszeit Modul	Siehe Auswahltabelle Rauschminderung (Seite 1363).	
Fehler kalte Verbindungsstelle	±1,5 °C	
Leitungslänge (Meter)	Max. 100 m zum Geber	
Leitungswiderstand	max. 100 Ω	

Tabelle A-169 Diagnose

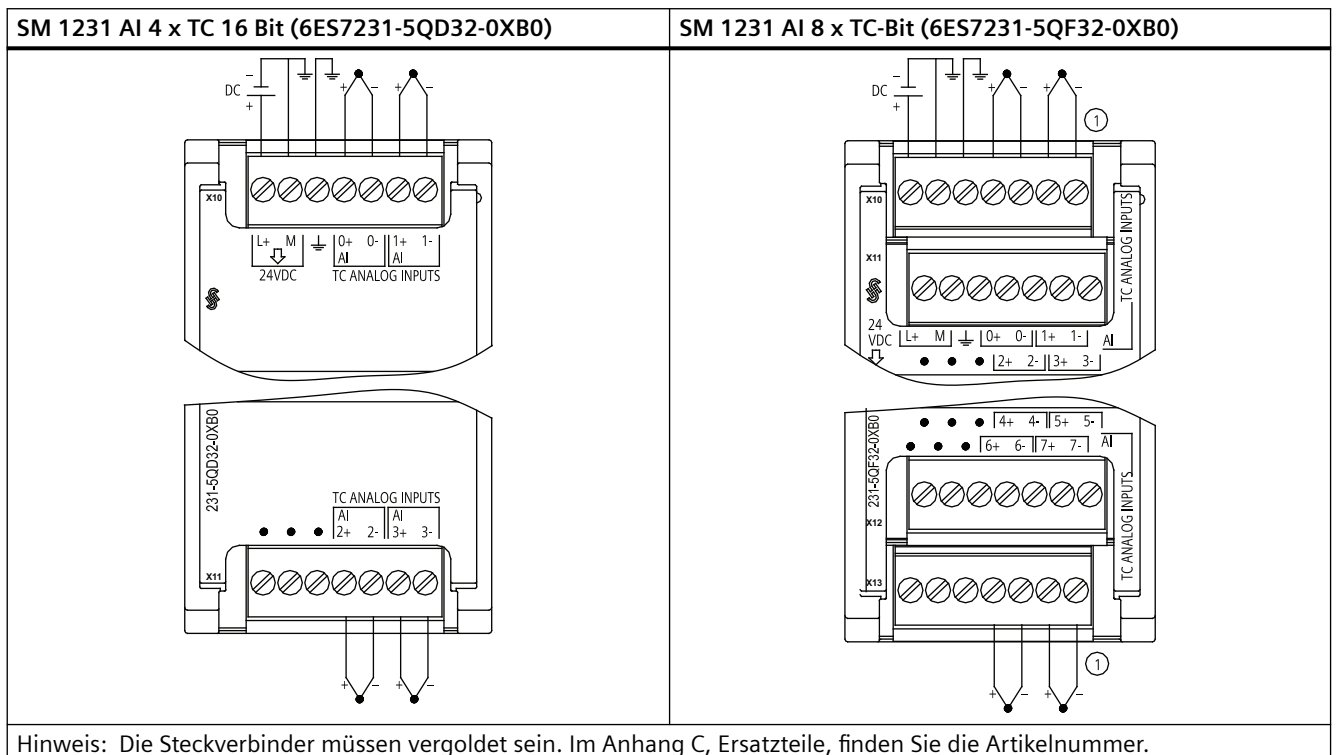
Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Überlauf/Unterlauf ¹	Ja	
Drahtbruch ^{2, 3}	Ja	
24 V DC Niederspannung ¹	Ja	

- 1 Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.
- 2 Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.
- 3 Das Modul führt die Drahtbruchprüfung alle 6 Sekunden durch, wodurch die Aktualisierungszeit alle 6 Sekunden für jeden aktivierten Kanal um 9 ms verlängert wird.

Das analoge Thermoelement-Signalmodul SM 1231 TC misst den Wert der an die Moduleingänge angeschlossenen Spannung. Als Temperaturmessart sind entweder "Thermoelement" oder "Spannung" möglich.

- "Thermoelement": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).
- "Spannung": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.

Tabelle A-170 Schaltpläne der Thermoelement-SMs



① Die Anschlüsse von TC 2, 3, 4 und 5 werden zur besseren Übersichtlichkeit weggelassen.

A.11 Thermoelement- und RTD-Signalmodule (SMs)

Tabelle A-171 Anschlussbelegung für das SM 1231 AI 4 x TC 16 Bit (6ES7231-5QD32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss
2	M / 24 V DC	Kein Anschluss
3	Funktionserde	Kein Anschluss
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

Tabelle A-172 Anschlussbelegung für das SM 1231 AI 8 x TC Bit (6ES7231-5QF32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss	Kein Anschluss
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

Hinweis

Nicht verwendete analoge Eingänge sollten kurzgeschlossen werden.

Die vom Thermoelement nicht verwendeten Kanäle können deaktiviert werden. Es tritt kein Fehler auf, wenn ein nicht verwendeter Kanal deaktiviert wird.

A.11.1.1 Grundlegende Funktionsweise eines Thermoelements

Thermoelemente entstehen, wenn zwei unterschiedliche Metalle elektrisch miteinander verbunden werden. Dadurch wird eine Spannung erzeugt, die proportional zu der Temperatur der Verbindungsstelle ist. Es handelt sich um eine geringe Spannung. Ein Mikrovolt kann viele Grade darstellen. Grundlage für die Temperaturmessung mit Thermoelementen sind das Messen der Spannung eines Thermoelements, das Kompensieren von zusätzlichen Verbindungsstellen und das Linearisieren der Ergebnisse.

Wenn Sie ein Thermoelement an das SM 1231 Thermoelementmodul anschließen, werden die beiden Leitungen der unterschiedlichen Metalle am Signalanschluss des Moduls angeschlossen. Die Stelle, an der die beiden unterschiedlichen Leitungen miteinander verbunden werden, bildet den Sensor des Thermoelements.

Zwei weitere Thermoelemente entstehen an der Stelle, an der die unterschiedlichen Leitungen an den Signalanschluss angeschlossen werden. Die Temperatur des Anschlusses erzeugt eine Spannung, die zu der Spannung des Thermoelementsensors addiert wird. Wird diese Spannung nicht ausgeglichen, weicht die ausgegebene Temperatur von der Temperatur des Sensors ab.

Durch die Kompensation der kalten Verbindungsstelle wird das Thermoelement am Anschluss ausgeglichen. Tabellen für Thermoelemente basieren auf einer Bezugstemperatur an der Verbindungsstelle, üblicherweise Null Grad Celsius. Durch die Kompensation der kalten Verbindungsstelle wird der Anschluss auf Null Grad Celsius kompensiert. Die Spannung, die durch das Thermoelement des Anschlusses addiert wird, wird durch die Kompensation der kalten Verbindungsstelle korrigiert. Die Temperatur des Moduls wird innen gemessen und dann in einen Wert umgewandelt, der zur Sensorumsetzung addiert wird. Die korrigierte Sensorumsetzung wird dann mittels der Thermoelement-Tabellen linearisiert.

Für die optimale Funktionsweise der Kompensation der kalten Verbindungsstelle muss sich das Thermoelementmodul in einer thermisch stabilen Umgebung befinden. Eine langsame Veränderung (weniger als 0,1 °C/Minute) der Temperatur in der Umgebung des Moduls wird innerhalb der Modulspezifikation korrekt ausgeglichen. Auch Luftbewegungen am Modul verursachen Fehler bei der Kompensation der kalten Verbindungsstelle.

Ist eine bessere Kompensation von Fehlern an der kalten Verbindungsstelle erforderlich, kann eine externe isothermische Klemmenleiste verwendet werden. Mit dem Thermoelementmodul kann eine auf 0 °C bezogene oder eine auf 50 °C bezogene Klemmenleiste eingesetzt werden.

A.11.1.2 Auswahltabellen für das SM 1231 Thermoelement

Die Bereiche und Genauigkeit der verschiedenen vom SM 1231 Thermoelement-Signalmodul unterstützten Thermoelementtypen entnehmen Sie bitte der unten stehenden Tabelle.

Tabelle A-173 Thermoelement-Auswahltabelle

Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Genauigkeit Normalbereich ^{3,4} bei 25 °C	Normalbereich ^{1,2,6} Genauigkeit -20 °C bis 60 °C
J	-210,0 °C	-150,0 °C	1.200,0 °C	1.450,0 °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	1.372,0 °C	1.622,0 °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	1.000,0 °C	1.200,0 °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	1.768,0 °C	2.019,0 °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	1.820,0 °C	1.820,0 °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	1.300,0 °C	1.550,0 °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	2.315,0 °C	2.500,0 °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

A.11 Thermoelement- und RTD-Signalmodule (SMs)

Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Genauigkeit Normalbereich ^{3,4} bei 25 °C	Normalbereich ^{1,2,6} Genauigkeit -20 °C bis 60 °C
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1.050,0 °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
Spannung	-32512	-27648 -80 mV	27648 80 mV	32511	±0,05 %	±0,1 %

- ¹ Die Thermoelementwerte unterhalb des Minimums für den unteren Bereich werden als -32768 angegeben.
- ² Die Thermoelementwerte oberhalb des Maximums für den oberen Bereich werden als 32767 angegeben.
- ³ Der interne Fehler an der kalten Verbindungsstelle beträgt ±1,5 °C für alle Bereiche. Dieser Wert ist zu den in dieser Tabelle aufgeführten Fehlern zu addieren. Das Modul benötigt eine Aufwärmzeit von mindestens 30 Minuten, bis die hier genannten Spezifikationen erfüllt werden. Bei Umgebungstemperaturen unter -10 °C kann der interne Fehler an der kalten Verbindungsstelle größer als 1,5 °C sein.
- ⁴ Bei einer abgestrahlten Funkfrequenz von 970 MHz bis 990 MHz kann sich die Genauigkeit des SM 1231 AI 4 x 16 Bit TC verschlechtern.
- ⁵ Unterer Grenzwert 3276,6 mit Meldung in °F
- ⁶ Der Fehler bei der Kompensation an der kalten Verbindungsstelle wurde für Umgebungstemperaturen unter 0 °C nicht charakterisiert und kann den angegebenen Wert überschreiten.

Hinweis

Thermoelementkanal

Für jeden Kanal am Thermoelement-Signalmodul kann ein unterschiedlicher Thermoelementtyp konfiguriert werden (einstellbar in der Software während der Modulkonfiguration).

Tabelle A-174 Rauschminderung und Aktualisierungszeiten für das SM 1231 Thermoelement

Auswahl Unterdrückungsfrequenz	Integrationszeit	Aktualisierungszeit 4-Kanal-Modul(Sekunden)	Aktualisierungszeit 8-Kanal-Modul(Sekunden)
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 Um die Auflösung und Genauigkeit des Moduls beizubehalten, wenn die 400 Hz Unterdrückung ausgewählt ist, beträgt die Integrationszeit 10 ms. Durch diese Auswahl wird auch Rauschen mit 100 Hz und 200 Hz unterdrückt.

Für die Messung von Thermoelementen wird eine Integrationszeit von 100 ms empfohlen. Niedriger eingestellte Integrationszeiten führen zu einem höheren Wiederholgenauigkeitsfehler der Temperaturmessungen.

Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm gegebenenfalls berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Gegebenenfalls können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

Sie können diese Logik mit einer Leseabfrage im "Anlauf-OB" implementieren, wodurch der Betrieb gesperrt wird, bis die Initialisierung abgeschlossen ist. Sie müssen die Leseabfrage mit sofortigem Zugriff implementieren. Wenn der Wert der Leseabfrage des Thermoelements 32767 beträgt, muss das Lesen wiederholt werden, bis sich der Wert ändert. Diese Abfrage braucht bei jedem Modul nur für den Eingang mit der größten Nummer des Moduls ausgeführt zu werden (die Moduleingänge werden der Reihe nach von 0 bis 7 initialisiert).

Darstellung der Analogwerte für ein Thermoelement des Typs J

Die Analogwerte für ein Thermoelement des Typs J finden Sie in der folgenden Tabelle.

Tabelle A-175 Darstellung der Analogwerte von Thermoelementen des Typs J

Typ J in °C	Einheiten		Typ J in °F	Einheiten		Bereich
	Dezimal	Hexadezimal		Dezimal	Hexadezimal	
> 1450,0	32767	7FFF	> 2642,0	32767	7FFF	Überlauf
1450,0	14500	38A4	2642,0	26420	6734	Überschreitung
:	:	:	:	:	:	
1200,1	12001	2EE1	2192,2	21922	55A2	Bemessungsbereich
1200,0	12000	2EE0	2192,0	21920	55A0	
:	:	:	:	:	:	Unterschreitung
-150	1500	FA24	-238,0	-2380	F6B4	
-150,1	-1501	FA23	-238,1	-2381	F6B3	Unterschreitung
:	:	:	:	:	:	
-210	-2100	F7CC	-346,0	-3460	F27C	Unterlauf ¹
< -210,0	-32768	8000	< -346,0	-32768	8000	

¹ Fehlerhafte Verdrahtung (z. B. Verpolung oder offene Eingänge) oder ein Sensorfehler im negativen Bereich (z. B. falscher Typ des Thermoelements) können bewirken, dass das Thermoelementmodul Unterlauf meldet.

A.11.2 SM 1231 RTD

Technische Daten des SM 1231 RTD

Tabelle A-176 Allgemeine technische Daten

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x 16 Bit
Artikelnummer	6ES7231-5PD32-0XB0	6ES7231-5PF32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	70 x 100 x 75
Gewicht	220 Gramm	270 Gramm
Leistungsverlust	1,5 W	
Stromaufnahme (SM-Bus)	80 mA	90 mA
Stromaufnahme (24 V DC) ¹	40 mA	

¹ 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus CPU)

Tabelle A-177 Analoge Eingänge

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x 16 Bit
Anzahl der Eingänge	4	8
Typ	Modulreferenz RTD und Ω	
Bereich Nennbereich (Datenwort) Überschwing-/Unterschwingbereich (Datenwort) Überlauf/Unterlauf (Datenwort)	Siehe Auswahltablette RTD-Geber (Seite 1369).	
Auflösung	Temperatur	0,1 °C / 0,1 °F
	Widerstand	15 Bit plus Vorzeichen
Max. Stehspannung	±35 V	
Rauschunterdrückung	85 dB für die gewählte Rauschminderung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)	
Gleichtaktunterdrückung	>120 dB	
Impedanz	≥ 10 M Ω	
Elektrische Trennung	Feldseite zu Logik	707 V DC (Typprüfung)
	Feld an 24 V DC	707 V DC (Typprüfung)
	24 V DC an Logik	707 V DC (Typprüfung)
Trennung Kanäle untereinander	Keine	
Genauigkeit	Siehe Auswahltablette RTD-Geber (Seite 1369).	
Wiederholgenauigkeit	±0,05 % Vollausschlag	
Maximale Verlustleistung Geber	0,5 mW	
Messprinzip	Integrierend	
Aktualisierungszeit Modul	Siehe Auswahltablette Rauschminderung (Seite 1369).	
Leitungslänge (Meter)	Max. 100 m zum Geber	
Leitungswiderstand	20 Ω , 2,7 Ω für max. 10 Ω RTD	

Tabelle A-178 Diagnose

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x16 Bit
Überlauf/Unterlauf ^{1,2}	Ja	
Drahtbruch ³	Ja	
24 V DC Niederspannung ¹	Ja	

¹ Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

² Für Widerstandsmessbereiche ist die Prüfung auf Unterlauf grundsätzlich nicht aktiviert.

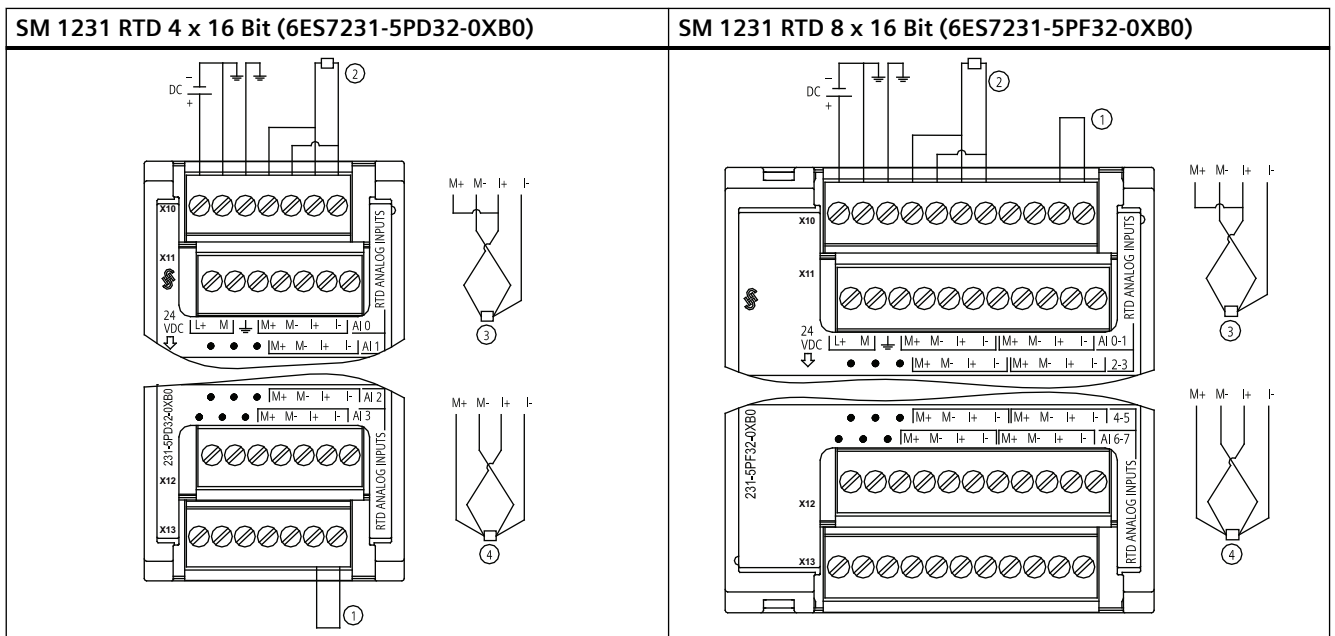
³ Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Signalmodul SM 1231 RTD misst den Wert des an die Moduleingänge angeschlossenen Widerstands. Als Messart kann entweder "Widerstand" oder "Thermischer Widerstand" ausgewählt werden.

- "Widerstand": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.
- "Thermischer Widerstand": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt). Die Werte des klimatischen Bereichs werden in Grad, multipliziert mit Hundert, ausgegeben (Beispiel: 25,34 Grad werden als Dezimalwert 2534 dargestellt).

Das Signalmodul SM 1231 RTD unterstützt Messungen über 2-Leiter-, 3-Leiter- und 4-Leiter-Anschlüsse zum Geberwiderstand.

Tabelle A-179 Schaltpläne der RTD-SMs



- ① Nicht belegte RTD-Eingänge zurückschleifen
 ② 2-Draht-RTD ③ 3-Draht-RTD ④ 4-Draht-RTD

HINWEIS: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

A.11 Thermoelement- und RTD-Signalmodule (SMs)

Tabelle A-180 Anschlussbelegung für das SM SM 1231 RTD 4 x 16 Bit (6ES7231-5PD32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss	Kein Anschluss
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

Tabelle A-181 Anschlussbelegung für das SM 1231 RTD 8 x 16 Bit (6ES7231-5PF32-0XB0)

Pin	X10 (vergoldet)	X11 (vergoldet)	X12 (vergoldet)	X13 (vergoldet)
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss	Kein Anschluss
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	AI 7 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

Hinweis

Die vom RTD nicht verwendeten Kanäle können deaktiviert werden. Es tritt kein Fehler auf, wenn ein nicht verwendeter Kanal deaktiviert wird.

Das RTD-Modul benötigt eine kontinuierliche Stromschleife, um die zusätzliche Stabilisierungszeit zu beseitigen, die bei einem nicht verwendeten Kanal, der nicht deaktiviert ist, automatisch anfällt. Aus Konsistenzgründen muss ein Widerstand an das RTD-Modul angeschlossen werden (z. B. der 2-Draht-RTD-Anschluss).

A.11.2.1 Auswahltabellen für das SM 1231 RTD

Tabelle A-182 Bereiche und Genauigkeit für die verschiedenen Geber, die von den RTD-Modulen unterstützt werden

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 Klima	-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	1.000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1.000,0 °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	1.000,0 °C	±0,5 °C	±1,0 °C
	Pt 200	-243,0 °C	-200,0 °C	850,0 °C	1.000,0 °C	±0,5 °C	±1,0 °C
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1.100,0 °C	1.295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1.100,0 °C	1.295 °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 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 Die RTD-Werte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.

² RTD-Werte oberhalb des Maximums für den oberen Bereich werden als +32767 ausgegeben.

A.11 Thermoelement- und RTD-Signalmodule (SMs)

Tabelle A-183 Widerstand

Bereich	Minimum unterer Bereich	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ¹	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
150 Ω	nicht zutreffend	0 (0 Ω)	27648 (150 Ω)	176.383 Ω	±0,05 %	±0,1 %
300 Ω	nicht zutreffend	0 (0 Ω)	27648 (300 Ω)	352.767 Ω	±0,05 %	±0,1 %
600 Ω	nicht zutreffend	0 (0 Ω)	27648 (600 Ω)	705.534 Ω	±0,05 %	±0,1 %

¹ Die Widerstandswerte oberhalb des Minimums für den oberen Bereich werden als +32767 ausgegeben.

Hinweis

Für alle aktivierte Kanäle ohne angeschlossenen Geber meldet das Modul 32767. Wenn außerdem die Prüfung auf offene Leitungen aktiviert ist, blinken am Modul die entsprechenden roten LED.

Wenn RTD-Bereiche von 500 Ω und 1000 Ω mit anderen Widerständen niederen Werts verwendet werden, kann sich der Fehler auf den zweifachen spezifizierten Fehler erhöhen.

Optimale Genauigkeit für die 10 Ω-RTD-Bereiche ermöglichen 4-Leiter-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Leiter-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

Tabelle A-184 Rauschminderung und Aktualisierungszeiten für die RTD-Module

Auswahl Unterdrückungsfrequenz	Integrationszeit	Aktualisierungszeit (Sekunden)	
		4-Kanal-Modul	8-Kanal-Modul
400 Hz (2,5 ms)	10 ms ¹	4-/2-Draht: 0,142 3-Draht: 0,285	4-/2-Draht: 0,285 3-Draht: 0,525
60 Hz (16,6 ms)	16,67 ms	4-/2-Draht: 0,222 3-Draht: 0,445	4-/2-Draht: 0,445 3-Draht: 0,845
50 Hz (20 ms)	20 ms	4-/2-Draht: 0,262 3-Draht: 0,505	4-/2-Draht: 0,524 3-Draht: 1,015
10 Hz (100 ms)	100 ms	4-/2-Draht: 1,222 3-Draht: 2,445	4-/2-Draht: 2,425 3-Draht: 4,845

¹ Um bei ausgewähltem 400-Hz-Filter die Auflösung und Genauigkeit des Moduls beizubehalten, beträgt die Integrationszeit 10 ms. Durch diese Auswahl wird auch Rauschen mit 100 Hz und 200 Hz unterdrückt.

Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Ggf. können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

Sie können diese Logik mit einer Leseabfrage im "Anlauf-OB" implementieren, wodurch der Betrieb gesperrt wird, bis die Initialisierung abgeschlossen ist. Sie müssen die Leseabfrage mit sofortigem Zugriff implementieren. Wenn der Wert der Leseabfrage des Widerstandstemperatursensors 32767 beträgt, muss das Lesen wiederholt werden, bis sich der Wert ändert. Diese Abfrage braucht bei jedem Modul nur für den Eingang mit der größten Nummer des Moduls ausgeführt zu werden (die Moduleingänge werden der Reihe nach von 0 bis 7 initialisiert).

Darstellung der Analogwerte für RTDs

Die folgenden Tabellen zeigen den digitalisierten Messwert für die Sensoren mit RTD-Standardtemperaturbereich.

Tabelle A-185 Darstellung von Analogwerten für die Widerstandsthermometer PT 100, 200, 500, 1000 und PT 10, 50, 100, 500 GOST (0,003850) Standard

Pt x00 Standard in °C (1 Ziffer = 0,1 °C)	Einheiten		Pt x00 Standard in °F (1 Ziffer = 0,1 °F)	Einheiten		Bereich
	Dezimal	Hexadezimal		Dezimal	Hexadezimal	
> 1000.0	32767	7FFF	> 1832.0	32767	7FFF	Überlauf
1000.0	10000	2710	1832.0	18320	4790	Oberhalb des Bereichs
:	:	:	:	:	:	
850.1	8501	2135	1562.1	15621	3D05	Bemessungsbereich
850.0	8500	2134	1562.0	15620	3D04	
:	:	:	:	:	:	Unterhalb des Bereichs
-200.0	-2000	F830	-328.0	-3280	F330	
-200.1	-2001	F82F	-328.1	-3281	F32F	Unterhalb des Bereichs
:	:	:	:	:	:	
-243.0	-2430	F682	-405.4	-4054	F02A	Unterlauf
< -243.0	-32768	8000	< -405.4	-32768	8000	

A.12 Technologiemodule

A.12.1 SM 1278 4xIO-Link-Master SM

Tabelle A-186 Allgemeine technische Daten

Technische Daten		Signalmodul SM 1278 4xIO-Link-Master
Artikelnummer		6ES7278-4BD32-0XB0
Abmessungen B x H x T (mm)		45 x 100 x 75
Gewicht		150 Gramm
Allgemeine Informationen		
I&M-Daten		Ja, IM0 bis IM3
Versorgungsspannung		
Nennspannung (DC)		24 V DC
Unterer Grenzwert gültiger Bereich (DC)		19,2 V; 20,5 V bei Verwendung von IO-Link (weil die Versorgungsspannung für IO-Link-Geräte auf dem Master mindestens 20 V betragen muss)
Oberer Grenzwert gültiger Bereich (DC)		28,8 V DC
Verpolschutz		Ja
Eingangsstrom		
Stromaufnahme		65 mA, ohne Last
Drehgebersversorgung		
Anzahl der Ausgänge		4
Ausgangsstrom, Nennwert		200 mA pro Kanal
Verlustleistung		
Verlustleistung, typ.		1 W, ohne Portlast
Digitaleingänge/-ausgänge		
Leitungslänge (Meter)		Max. 20 m, ungeschirmt
SDLC		
Leitungslänge (Meter)		Max. 20 m, ungeschirmt
IO-Link		
Anzahl Ports		4
Anzahl Ports, die gleichzeitig gesteuert werden können		4
IO-Link-Protokoll 1.0		Ja
IO-Link-Protokoll 1.1		Ja
Betriebszustand		
IO-Link		Ja
DI		Ja
DO		Ja, max. 100 mA
Anschluss von IO-Link-Geräten		

Technische Daten		Signalmodul SM 1278 4xIO-Link-Master
	Porttyp A	Ja
	Übertragungsgeschwindigkeit	4,8 kBd (COM1)
		38,4 kBd (COM2)
		230,4 kBd (COM3)
	Min. Zykluszeit	2 ms, dynamisch, abhängig von der Länge der Anwenderdaten
	Größe der Prozessdaten, Eingang je Port	max. 32 Byte
	Größe der Prozessdaten, Eingang je Modul	32 Byte
	Größe der Prozessdaten, Ausgang je Port	max. 32 Bytes
	Größe der Prozessdaten, Ausgang je Modul	32 Bytes
	Speichergöße für Geräteparameter	2 KB
Max. Kabellänge ungeschirmt (Meter)	20 m	
Alarmer/Statusinformationen		
Statusanzeige	Ja	
Alarmer		
Diagnosealarm	Ja, die Portdiagnose ist nur im IO-Link-Modus verfügbar	
Diagnosealarmer		
Diagnose	Überwachung der Versorgungsspannung	Ja
	Kurzschluss	Ja
Diagnoseanzeige-LED		
Überwachung der Versorgungsspannung	Ja, blinkende rote DIAG-LED	
Kanalstatusanzeige	Ja, je Kanal eine grüne LED für den Kanalstatus Qn (SIO-Modus) und PORT-Status Cn (IO-Link-Modus)	
Für Kanaldiagnose	Ja, rote Fn-LED	
Für Moduldiagnose	Ja, grüne/rote DIAG-LED	
Potentialtrennung		
Potentialtrennung Kanäle		
Zwischen den Kanälen	Nein	
	Zwischen den Kanälen und dem Rückwandbus	Ja
Isolierung		
Isolierung geprüft mit	707 V DC (Typprüfung)	
Umgebungsbedingungen		
Betriebstemperatur		
Min.	-20 °C	
Max.	60 °C	
Horizontaler Einbau, min.	-20 °C	
Horizontaler Einbau, max.	60 °C	
Vertikaler Einbau, min.	-20 °C	
Vertikaler Einbau, max.	50 °C	

Übersicht über die Reaktionszeit

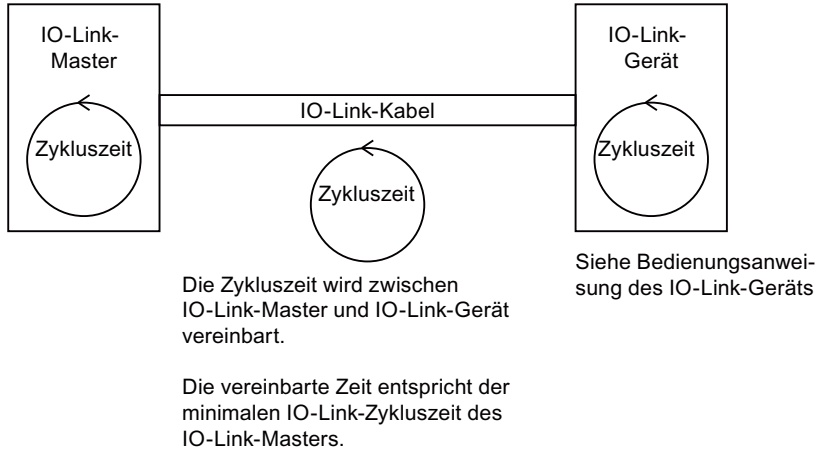


Tabelle A-187 Schaltplan für das SM 1278 IO-Link-Master

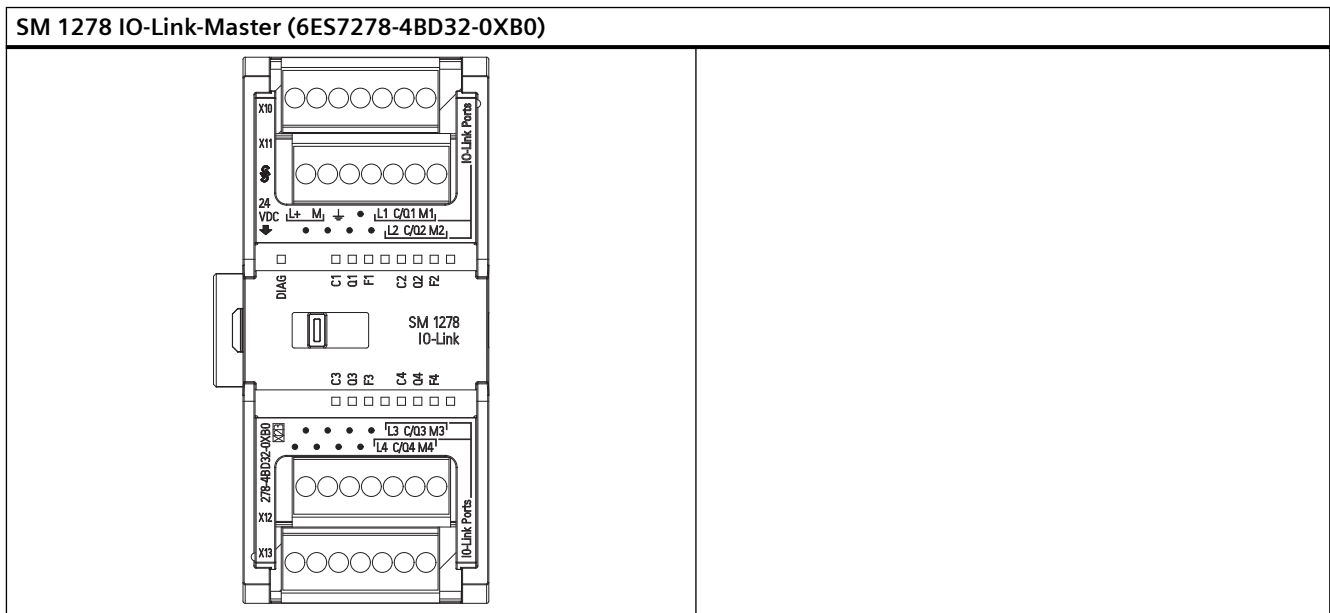


Tabelle A-188 Anschlussbelegung für das SM 1278 IO-Link-Master (6ES7278-4BD32-0XB0)

Pin	X10	X11	X12	X13
1	L+ / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
2	M / 24 V DC	Kein Anschluss	Kein Anschluss	Kein Anschluss
3	Funktionserde	Kein Anschluss	Kein Anschluss	Kein Anschluss
4	Kein Anschluss	Kein Anschluss	Kein Anschluss	Kein Anschluss
5	L ₁	L ₂	L ₃	L ₄
6	C/O ₁	C/O ₂	C/O ₃	C/O ₄
7	M ₁	M ₂	M ₃	M ₄

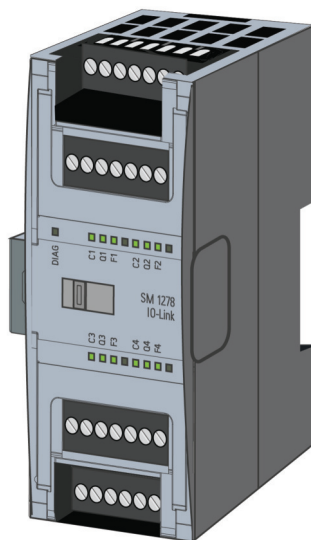
A.12.1.1 SM 1278 4xIO-Link-Master - Überblick

Der SM 1278 4xIO-Link-Master ist ein Modul mit 4 Ports, das als Signalmodul und als Kommunikationsmodul fungiert. Jeder Port kann im IO-Link-Modus, als einzelner 24-V-DC-Digitaleingang oder als 24-V-DC-Digitalausgang betrieben werden.

Der IO-Link-Master programmiert die azyklische Kommunikation mit einem IO-Link-Gerät mit Hilfe des Funktionsbausteins (FB) IO_LINK_DEVICE in Ihrem STEP 7 S7-1200 Steuerungsprogramm. Der FB IO_LINK_DEVICE gibt an, welcher IO-Link-Master in Ihrem Programm verwendet wird und welche Ports der Master für den Datenaustausch nutzt.

Auf der Website Siemens Industry Online Support (<https://support.industry.siemens.com/cs/ww/de>) sind Details zum Arbeiten mit der IO-Link-Bibliothek zu finden. Geben Sie im Suchfeld der Website "IO-Link" ein, um Informationen über IO-Link-Produkte und deren Einsatz aufzurufen.

Ansicht des Moduls



Eigenschaften

Technische Merkmale

- IO-Link-Master nach IO-Link-Spezifikation V1.1 (siehe Website des IO-Link-Konsortiums (<http://io-link.com/en/index.php>) mit weiteren Informationen)
- Serielles Kommunikationsmodul mit vier Schnittstellen (Kanälen)
- Datenübertragungsrate COM1 (4.8 kbaud), COM2 (38.4 kbaud), COM3 (230.4 kbaud)
- SIO-Modus (Standard IO-Modus)
- Anschluss von bis zu vier IO-Link-Geräten (3-Leiter-Anschluss) oder vier Standard-Aktoren oder Standard-Gebern
- Programmierbare Diagnosefunktion für jede Schnittstelle

Unterstützte Funktionen

- Identifikationsdaten für I&M (Installation und Instandhaltung)
- Firmware-Update
- IO-Link-Parametrierung mittels S7-PCT-Portkonfigurationstool, STEP 7 Professional und einer S7-1200 V4.0- oder höheren CPU. In STEP 7 Professional V15 kann die IO-Link-Parametrierung (mit Hilfe von V2.1 HSP oder höher) im TIA Portal mit begrenztem Funktionsumfang erfolgen.
- PQI (Port Qualifier Information) Bits
- Sichern und Wiederherstellen mit IO-Link-Bibliothek-FBs

IO-Link ist eine Punkt-zu-Punkt-Verbindung zwischen einem Master und einem Gerät. Als Geräte können sowohl konventionelle als auch intelligente Sensoren/Aktoren über ungeschirmte Standardkabel in der bewährten 3-Leiter-Technik am IO-Link angeschlossen werden. IO-Link ist abwärts kompatibel mit konventionellen digitalen Sensoren und Aktoren. Stromkreis und Datenkanal sind in bewährter 24 V DC-Technik ausgeführt.

Weitere Informationen über die SIMATIC-IO-Link-Technologie finden Sie im "IO-Link Systemhandbuch" auf der Website Siemens Industry Online-Support (<http://support.automation.siemens.com>).

Hinweis

IO-Link Parameterdaten

Bei einem Austausch des SM 4xIO-Link-Mastermoduls werden die Parameterdaten nicht automatisch zugewiesen.

 VORSICHT

Ziehen und Stecken

Wird das SM 4xIO-Link-Mastermodul bei eingeschalteter Last gesteckt, können dadurch gefährliche Zustände in Ihrer Anlage entstehen.

Das S7-1200-Automatisierungssystem kann beschädigt werden.
--

Ziehen und stecken Sie das SM 4xIO-Link-Mastermodul nur, wenn die Last ausgeschaltet ist.

Auswirkungen des Rücksetzens auf die Werkseinstellungen

Mit der Funktion "Rücksetzen auf Werkseinstellungen" können Sie die mit S7-PCT am Lieferzustand vorgenommene Parametrierung wiederherstellen.

Nach dem "Rücksetzen auf Werkseinstellungen" wird das SM 1278 4xIO-Link-Modul wie folgt parametrierung:

- Die Ports sind im DE-Modus
- Die Ports sind den relativen Adressen 0.0 bis 0.3 zugeordnet

- Der PortQualifier ist deaktiviert
- Die Wartungsdaten 1 bis 3 sind gelöscht

Hinweis

Beim Rücksetzen auf die Werkseinstellungen werden die Geräteparameter gelöscht und der Lieferzustand wiederhergestellt.

Wenn Sie ein SM 1278 4xIO-Link-Signalmodul entfernen, setzen Sie es vor dem Einlagern auf die Werkseinstellungen zurück.


Vorgehensweise

Gehen Sie zum "Rücksetzen auf die Werkseinstellungen" so vor, wie in der S7-PCT Online-Hilfe unter "Masterkonfiguration > Register 'Befehle' beschrieben.

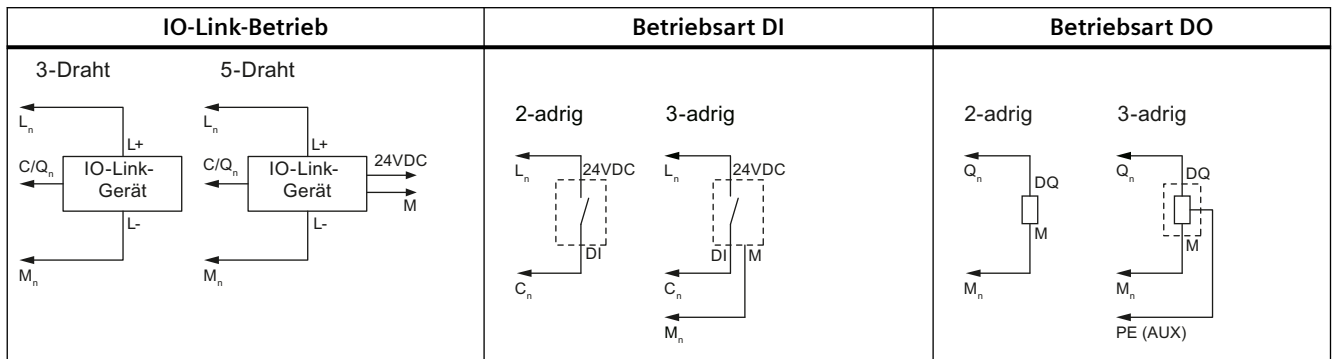
A.12.1.2 Anschluss

Näheres zur Anschlussbelegung finden Sie in der Tabelle Anschlussbelegung für SM 1278 I/O-Link-Master (6ES 278-4BD32-0XB0). (Seite 1372)

Die folgende Tabelle zeigt die Klemmenbelegung für den SM 1278 4xIO-Link-Master:

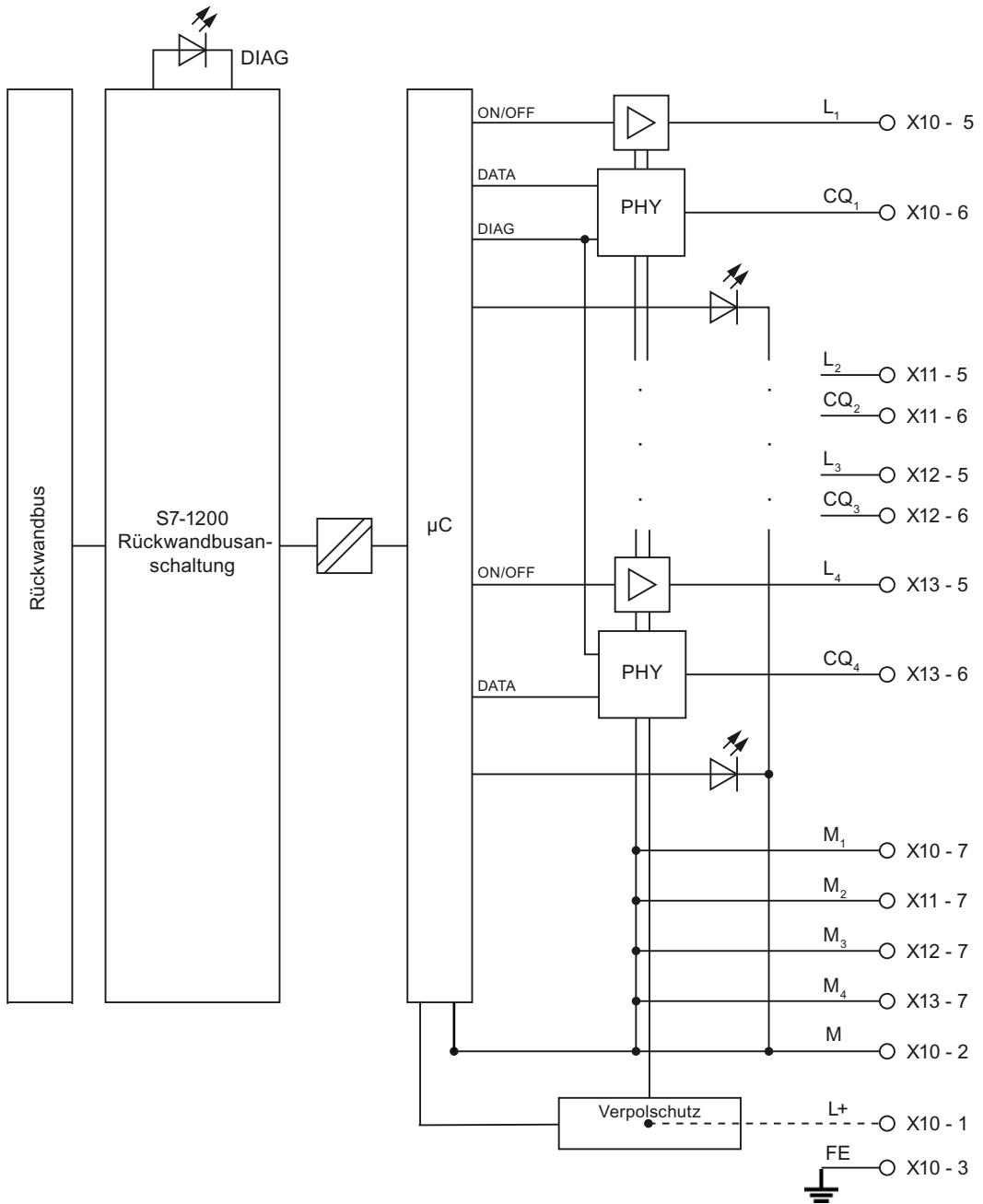
Pin	X10	X11	X12	X13	Hinweise	BaseUnits
7	M ₁	M ₂	M ₃	M ₄	<ul style="list-style-type: none"> • M_n: Masse zu Slave • C/O_n: SDLC, DI oder DO • L_n: 24 V DC an Slave • M: Masse • L+: 24 V DC an Master • RES: reserviert; darf nicht belegt werden 	A1
6	C/O ₁	C/O ₂	C/O ₃	C/O ₄		
5	L ₁	L ₂	L ₃	L ₄		
4	RES	RES	RES	RES		
3	 (Funktionserde)	RES	RES	RES		
2	M	RES	RES	RES		
1	L+	RES	RES	RES		

Die folgende Tabelle zeigt Anschlussbeispiele; dabei ist n = Schnittstellenummer.



Hinweis

Angeschlossene Sensoren müssen die über Anschluss L_n des Master-Moduls bereitgestellte Spannungsversorgung nutzen.



A.12.1.3 Parameter/Adressbereich

SM 1278 4xIO-Link-Master konfigurieren

Für die Modulintegration, Parametrierung und Inbetriebnahme benötigen Sie STEP 7 V13 oder höher. Für bestimmte Funktionen benötigen Sie auch S7-PCT (Port Configuration Tool).

Die folgende Tabelle zeigt, wann Sie S7-PCT benötigen:

	SM 1278 V2.0 4xIO-Link Master	SM 1278 V2.1 4xIO-Link Master
STEP7 V13.x und STEP V14.x	S7-PCT erforderlich	S7-PCT erforderlich
STEP 7 15.x	S7-PCT erforderlich	S7-PCT für Basisfunktionen nicht erforderlich S7-PCT für erweiterte Funktionen nicht erforderlich

Weitere Informationen finden Sie im Systemhandbuch zu SIMATIC IO-Link (<https://support.industry.siemens.com/cs/ww/de/view/65949252>).

Die folgende Tabelle zeigt die Parameter für den SM 1278 4xIO-Link-Master:

Parameter	Wertebereich	Voreinstellung	Konfiguration in RUN	Effizienzbereich
Diagnoseschnittstelle 1	<ul style="list-style-type: none"> • Deaktivieren • Aktivieren 	Deaktivieren	Ja	Schnittstelle (Kanal)
Diagnoseschnittstelle 2	<ul style="list-style-type: none"> • Deaktivieren • Aktivieren 	Deaktivieren	Ja	Schnittstelle (Kanal)
Diagnoseschnittstelle 3	<ul style="list-style-type: none"> • Deaktivieren • Aktivieren 	Deaktivieren	Ja	Schnittstelle (Kanal)
Diagnoseschnittstelle 4	<ul style="list-style-type: none"> • Deaktivieren • Aktivieren 	Deaktivieren	Ja	Schnittstelle (Kanal)

Parameter Diagnose für Schnittstelle 1 bis 4 aktivieren

Mit diesem Parameter kann die Diagnose für bestimmte Schnittstellen der vier IO-Link-Schnittstellen aktiviert werden.

Die Schnittstellenzuordnungen sind wie folgt:

Schnittstelle 1 → Kanal 1

Schnittstelle 2 → Kanal 2

Schnittstelle 3 → Kanal 3

Schnittstelle 4 → Kanal 4

Die maximale Größe der Ein- und Ausgangsadressen des SM 4xIO-Link Master beträgt jeweils 32 Byte. Sie können Adressbereiche mit dem S7-PCT-Portkonfigurationstool oder ab V15 V2.1 HSP oder höher mit der Hardwarekonfiguration des TIA Portal zuweisen.

Parameter-Datensatz

Parametrieren im Anwenderprogramm

Sie können das Gerät während der Laufzeit konfigurieren.

Parameter während der Laufzeit ändern

Die Modulparameter sind in Datensatz 128 enthalten. Mit der Anweisung WRREC können Sie die änderbaren Parameter an das Modul übertragen..

Nach dem Rücksetzen (Aus- und Einschalten) der CPU überschreibt die CPU die Parameter, die während der Parametrierung mit Anweisung WRREC in das Modul übertragen wurden.

Anweisung für die Parametrierung

Für die Parametrierung des E/A-Moduls im Anwenderprogramm steht die folgende Anweisung zur Verfügung:

Anweisung	Anwendung
SFB 53 WRREC	Übertragung der änderbaren Parameter in das Modul.

Fehlermeldung

Bei einem Fehler wird der folgende Rückgabewert gemeldet:

Fehlercode	Bedeutung
80B1 _H	Fehlerhafte Datenlänge
80E0 _H	Fehlerhafte Header-Information
80E1 _H	Parameterfehler

Datensatz-Struktur

Die folgende Tabelle zeigt die IO-Link-Parameter:

Offset	Bezeichnung	Typ	Voreinstellung	Beschreibung
0	Version	1 Byte	0x02	Zeigt die Struktur von Datensatz 0x02 für den IO-Link-Master nach IO-Link V1.1
1	Parameterlänge	1 Byte	0x02	Parameterlänge (2 Byte + 2 Header)
IO-Link-Startparameter				
2	Schnittstellendiagnose (Schnittstelle 1 bis n)	1 Byte	0x00	Aktivieren der Diagnose für Schnittstelle 1 bis n
3	IOL-Eigenschaften	1 Byte	0x00	Moduleigenschaften

Die folgende Tabelle zeigt den Versionsdatensatz:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserviert		Hauptversion (00)		Nebenversion (0010)			

Die folgende Tabelle zeigt den Datensatz für die Schnittstellendiagnose:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserviert				EN_Port4	EN_Port3	EN_Port2	EN_Port1

EN_Portx:

0 = Diagnose deaktiviert

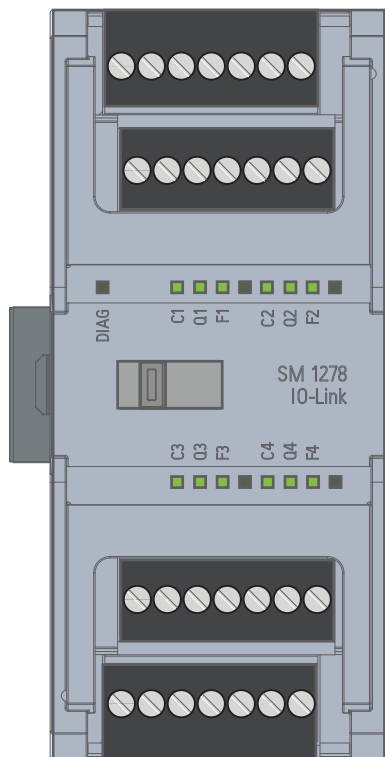
1 = Diagnose aktiviert

Die folgende Tabelle zeigt den Datensatz mit den IOL-Eigenschaften:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserviert							

A.12.1.4 Alarm-, Fehler- und Systemmeldungen

LED-Anzeige



Bedeutung der LED-Anzeigen

Die folgende Tabelle erläutert die Bedeutung der Zustands- und Fehleranzeigen. Abhilfemaßnahmen bei Diagnosealarmen finden Sie im Kapitel "Diagnosemeldungen".

LED DIAG

DIAG	Bedeutung
□ Aus	Rückwandbus-Versorgung der S7-1200 nicht OK
⚡ Blinkt	Modul ist nicht konfiguriert
■ Ein	Modul parametriert und keine Moduldiagnose
⚡ Blinkt	Modul parametriert und Moduldiagnose ODER L+ nicht angeschlossen

LED Port-Status

Gültig für einen IO-Link-Port im IO-Link-Portmodus.

COM/1 ... COM/4	Bedeutung
□ Aus	Port deaktiviert
☀ Blinkt	Port aktiviert, Gerät nicht angeschlossen oder Port nicht an das konfigurierte Gerät angeschlossen
■ Ein	Port aktiviert, Gerät angeschlossen

LED Kanalstatus

Gültig für einen IO-Link-Port in DI/O-Modus.

DI/O1 ... DI/O4	Bedeutung
□ Aus	Prozesssignal = 0
■ Ein	Prozesssignal = 1

LED Portfehler

F1...F4	Bedeutung
□ Aus	Kein Fehler
■ Ein	Fehler

Modulfehler werden nur im IO-Link-Modus als Diagnose (Modulstatus) angezeigt.

Diagnose- alarm	Fehler- code (de- zimal)	STATUS (W#16#...)	Bedeutung (IO-Link-Fehlercode)	IO-Link- Master	IO-Link- Gerät
Kurzschluss	1	1804	Kurzschluss an den Prozesskabeln am IO-Link-Gerät	X	
		7710	Kurzschluss am IO-Gerät		X
Unterspan- nung	2	5111 5112	Versorgungsspannung zu gering		X
Überspan- nung	3	5110	Versorgungsspannung zu hoch		X
Überhitzung	5	1805	Temperatur am Master überschritten	X	
		4000	Temperatur am Gerät überschritten		X
		4210			

Diagnose- alarm	Fehler- code (de- zimal)	STATUS (W#16#...)	Bedeutung (IO-Link-Fehlercode)	IO-Link- Master	IO-Link- Gerät
Drahtbruch	6	1800	<ul style="list-style-type: none"> Kein IO-Link-Gerät angeschlossen Drahtbruch in der Signalleitung zum IO-Link-Gerät IO-Link-Gerät kann wegen eines anderen Fehlers nicht kommunizieren 	X	
Überlauf	7	8C10 8C20	Prozessvariablenbereich überschritten		X
		8C20	Messbereich überschritten		
Unterlauf	8	8C30	Prozessvariablenbereich zu gering		X
Fehler	9	---	Alle hier nicht aufgeführten IO-Link-Fehlercodes werden diesem PROFIBUS DP-Fehler zugeordnet.		X
Parametrie- rungsfehler	16	1882 1883	IO-Link-Master konnte nicht konfiguriert werden	X	
		1802	Falsches Gerät		
		1886	Speicherfehler		
		6320 6321 6350	Gerät wurde nicht richtig konfiguriert		X
Versorgungs- spannung fehlt	17	1806	L+-Versorgungsspannung für das Gerät fehlt	X	
		1807	L+-Versorgungsspannung für das Gerät zu gering (< 20 V)		
Defekte Si- cherung	18	5101	Sicherung am Gerät ist defekt		X
Sicherheits- abschaltung	25	1880	Schwerer Fehler (Master muss ersetzt werden)	X	
Externe Stö- rung	26	1809 180A 180B 180C 180D	Fehler im Datenspeicher	X	
		1808	Am IO-Link-Gerät stehen mehr als 6 Fehler gleichzeitig an		

A.13 Digitale Signalboards (SBs)

A.13.1 Technische Daten des SB 1221 200 kHz Digitaleingabe

Tabelle A-189 Allgemeine technische Daten

Technische Daten	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Artikelnummer	6ES7221-3BD30-0XB0	6ES7221-3AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	
Gewicht	35 Gramm	
Leistungsverlust	1,5 W	1,0 W
Stromaufnahme (SM-Bus)	40 mA	
Stromaufnahme (24 V DC)	7 mA / Eingang + 20 mA	15 mA / Eingang + 15 mA

Tabelle A-190 Digitale Eingänge

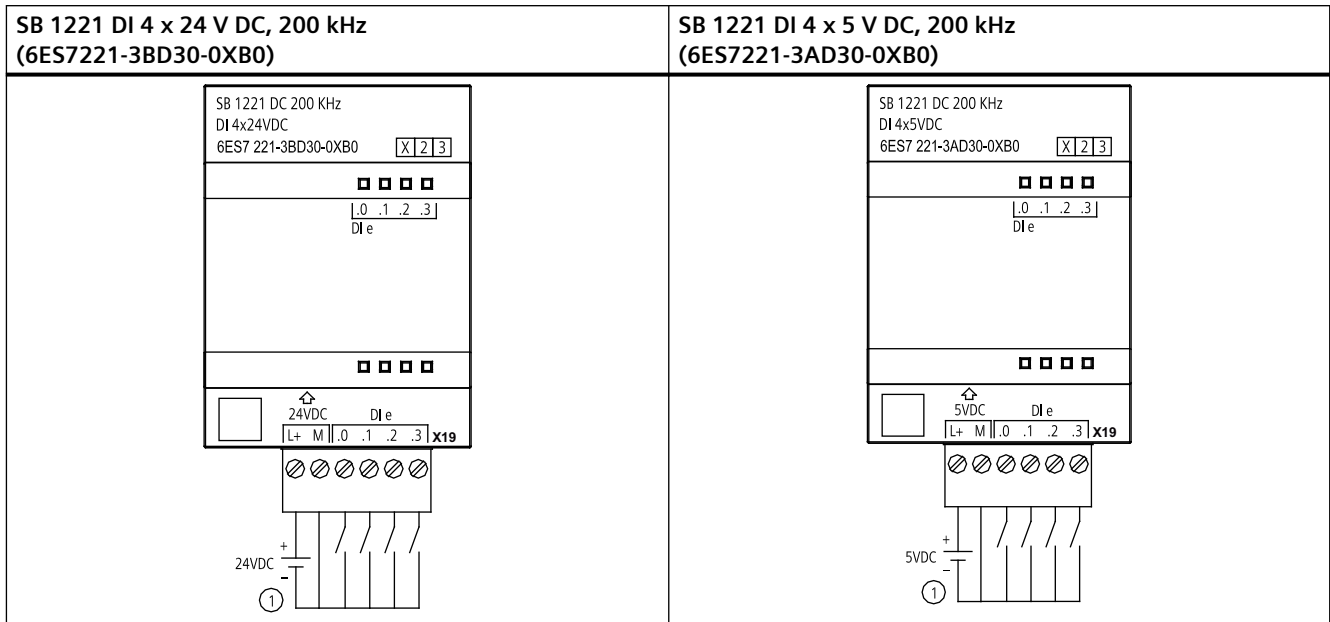
Technische Daten	SB 1221 DI 4 x 24 V DC, 200 kHz	SB 1221 DI 4 x 5 V DC, 200 kHz
Anzahl der Eingänge	4	
Typ	Stromliefernd	
Nennspannung	24 V DC bei 7 mA, nominal	5 V DC bei 15 mA, nominal
Zulässige Dauerspannung	28,8 V DC	6 V DC
Stoßspannung	35 V DC für 0,5 s	6 V
Signal logisch 1	0 V (10 mA) an L+ minus 10 V (2,9 mA)	0 V (20 mA) an L+ minus 2,0 V (5,1 mA)
Signal logisch 0	L+ minus 5 V (1,4 mA) an L+ (0 mA)	L+ minus 1,0 V (2,2 mA) an L+ (0 mA)
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	1	
Filterzeiten	µs-Einstellungen:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
	ms-Einstellungen	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Anzahl gleichzeitig eingeschalteter Eingänge	<ul style="list-style-type: none"> 2 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal 4 bei 55 °C horizontal oder 45 °C vertikal 	4
Leitungslänge (Meter)	50 m, geschirmtes, verdrehtes Leiterpaar	

Hinweis

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
- Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/ stromliefernden (P-M-schaltenden) Treiber.
- Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
- Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
- Fügen Sie am Eingang eine externe Last hinzu.

Tabelle A-191 Schaltpläne der Digitaleingangs-SBs (200 kHz)



① Unterstützt nur stromliefernde Eingänge

Tabelle A-192 Anschlussbelegung für das 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

Tabelle A-193 Anschlussbelegung für das 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 Technische Daten des SB 1222 200 kHz Digitalausgabe

Tabelle A-194 Allgemeine technische Daten

Technische Daten	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Artikelnummer	6ES7222-1BD30-0XB0	6ES7222-1AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	
Gewicht	35 Gramm	
Leistungsverlust	0,5 W	
Stromaufnahme (SM-Bus)	35 mA	
Stromaufnahme (24 V DC)	15 mA	

Tabelle A-195 Digitale Ausgänge

Technische Daten	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Ausgänge	4	
Ausgangstyp	MOSFET, elektronisch (stromziehend/stromliefernd) ¹	
Spannungsbereich	20,4 bis 28,8 V DC	4,25 bis 6,0 V DC
Signal logisch 1 bei max. Strom	L+ minus 1,5 V	L+ minus 0,7 V
Signal logisch 0 bei max. Strom	max. 1,0 V DC	max. 0,2 V DC
Strom (max.)	0,1 A	
Lampenlast	--	
Kontaktwiderstand bei EIN	max. 11 Ω	max. 7 Ω
Widerstand bei AUS	max. 6 Ω	max. 0,2 Ω
Kriechstrom pro Ausgang	--	
Frequenz Impulsgenerator	max. 200 kHz, min. 2 Hz	
Einschaltstrom	0,11 A	
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	1	
Ströme je Leiter	0,4 A	
Induktive Klemmspannung	Nein	

Technische Daten	SB 1222 DQ 4 x 24 V DC, 200 kHz	SB 1222 DQ 4 x 5 V DC, 200 kHz
Schaltverzögerung	1,5 µs + 300 ns steigend 1,5 µs + 300 ns fallend	200 ns + 300 ns steigend 200 ns + 300 ns fallend
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Nein	
Parallele Ausgänge für erhöhte Last	Nein	
Anzahl gleichzeitig eingeschalteter Ausgänge	<ul style="list-style-type: none"> • 2 (keine benachbarten Punkte) bei 60 °C horizontal oder 50 °C vertikal • 4 bei 55 °C horizontal oder 45 °C vertikal 	4
Leitungslänge (Meter)	50 m, geschirmtes, verdrehtes Leiterpaar	

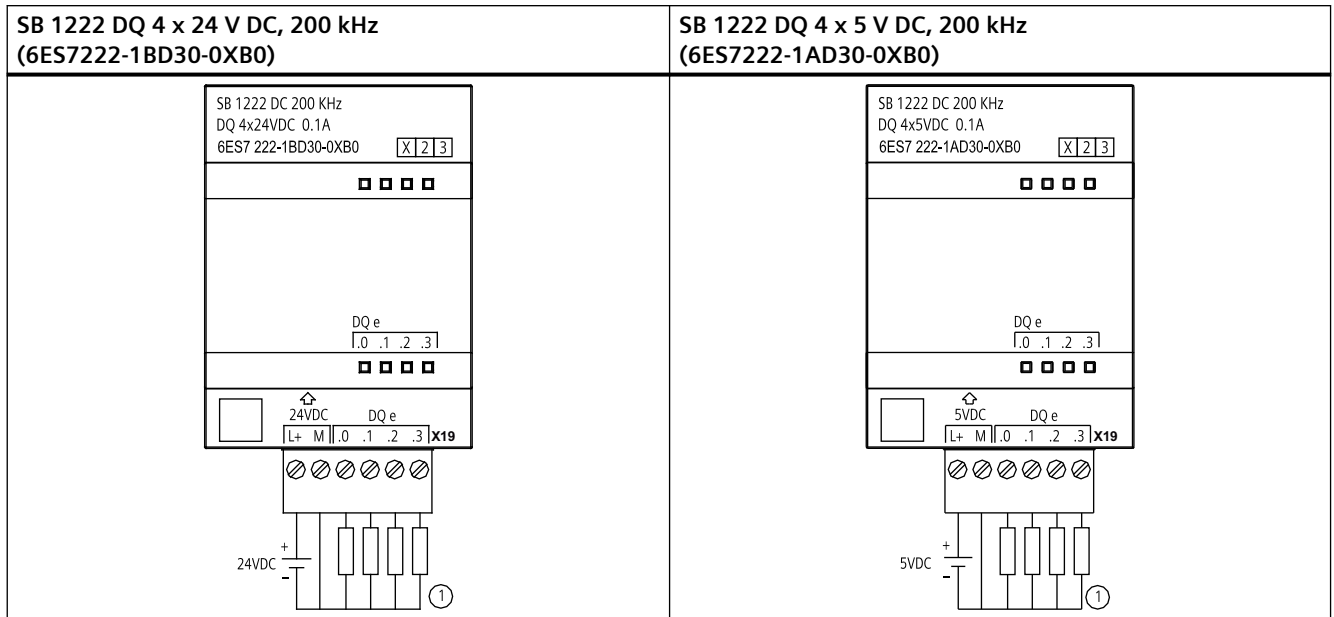
¹ Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Hinweis

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
 - Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/stromliefernden (P-M-schaltenden) Treiber.
 - Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
 - Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
 - Fügen Sie am Eingang eine externe Last hinzu.
-

Tabelle A-196 Schaltpläne der Digitalausgangs-SBs (200 kHz)



① Bei stromliefernden Ausgängen "Load" an "-" anschließen (s. Abbildung). Bei stromziehenden Ausgängen "Load" an "+" anschließen. Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Hinweis

Achten Sie darauf, dass der Draht der Masseverbindung sicher geerdet ist. Durch den Verlust des Erdanschlusses der Hochgeschwindigkeits-DO-SBs kann es möglicherweise zu genügend Kriechstrom kommen, um eine DC-Last zu aktivieren. Werden die Ausgänge für wichtige DC-Lastanwendungen genutzt, sollte als zusätzliche Sicherheitsvorkehrung eine redundante Erdungsleitung an das SB angeschlossen werden.

Tabelle A-197 Anschlussbelegung für das 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	DO e.0
4	DO e.1
5	DO e.2
6	DO e.3

A.13 Digitale Signalboards (SBs)

Tabelle A-198 Anschlussbelegung für das 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	DO e.0
4	DO e.1
5	DO e.2
6	DO e.3

A.13.3 Technische Daten des SB 1223 200 kHz Digitalein-/Digitalausgabe

Tabelle A-199 Allgemeine technische Daten

Technische Daten	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
Artikelnummer	6ES7223-3BD30-0XB0	6ES7223-3AD30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21	
Gewicht	35 Gramm	
Leistungsverlust	1,0 W	0,5 W
Stromaufnahme (SM-Bus)	35 mA	
Stromaufnahme (24 V DC)	7 mA / Eingang + 30 mA	15 mA / Eingang + 15 mA

Tabelle A-200 Digitale Eingänge

Technische Daten	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
Anzahl der Eingänge	2	
Typ	Stromliefernd	
Nennspannung	24 V DC bei 7 mA, nominal	5 V DC bei 15 mA, nominal
Zulässige Dauerspannung	28,8 V DC	
Stoßspannung	35 V DC für 0,5 s	
Signal logisch 1	0 V (10 mA) an L+ minus 10 V (2,9 mA)	0 V (20 mA) an L+ minus 2,0 V (5,1 mA)
Signal logisch 0	L+ minus 5 V (1,4 mA) an L+ (0 mA)	L+ minus 1,0 V (2,2 mA) an L+ (0 mA)
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 200 kHz A/B-Zähler: 160 kHz	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	1 (keine Trennung gegen Ausgang)	
Filterzeiten	µs-Einstellungen:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
	ms-Einstellungen:	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Anzahl gleichzeitig eingeschalteter Eingänge	2	
Leitungslänge (Meter)	50 m, geschirmtes, verdrilltes Leiterpaar	

Tabelle A-201 Digitale Ausgänge

Technische Daten	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
Ausgänge	2	
Ausgangstyp	MOSFET, elektronisch (stromziehend/stromliefernd) ¹	
Spannungsbereich	20,4 bis 28,8 V DC	4,25 bis 6,0 V DC
Nennwert	24 V DC	5 V DC
Signal logisch 1 bei max. Strom	L+ minus 1,5 V	L+ minus 0,7 V
Signal logisch 0 bei max. Strom	max. 1,0 V DC	max. 0,2 V DC
Strom (max.)	0,1 A	
Lampenlast	--	
Kontaktwiderstand bei EIN	max. 11 Ω	max. 7 Ω
Widerstand bei AUS	max. 6 Ω	max. 0,2 Ω
Kriechstrom pro Ausgang	--	
Frequenz Impulsgenerator	max. 200 kHz, min. 2 Hz	
Einschaltstrom	0,11 A	
Überlastschutz	Nein	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	1 (keine Trennung gegen Eingang)	
Ströme je Leiter	0,2 A	
Induktive Klemmspannung	Nein	
Schaltverzögerung	1,5 μ s + 300 ns steigend 1,5 μ s + 300 ns fallend	200 ns + 300 ns steigend 200 ns + 300 ns fallend
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Ansteuerung eines Digitaleingangs	Ja	
Parallele Ausgänge für redundante Laststeuerung	Nein	
Parallele Ausgänge für erhöhte Last	Nein	
Anzahl gleichzeitig eingeschalteter Ausgänge	2	
Leitungslänge (Meter)	50 m, geschirmtes, verdrilltes Leiterpaar	

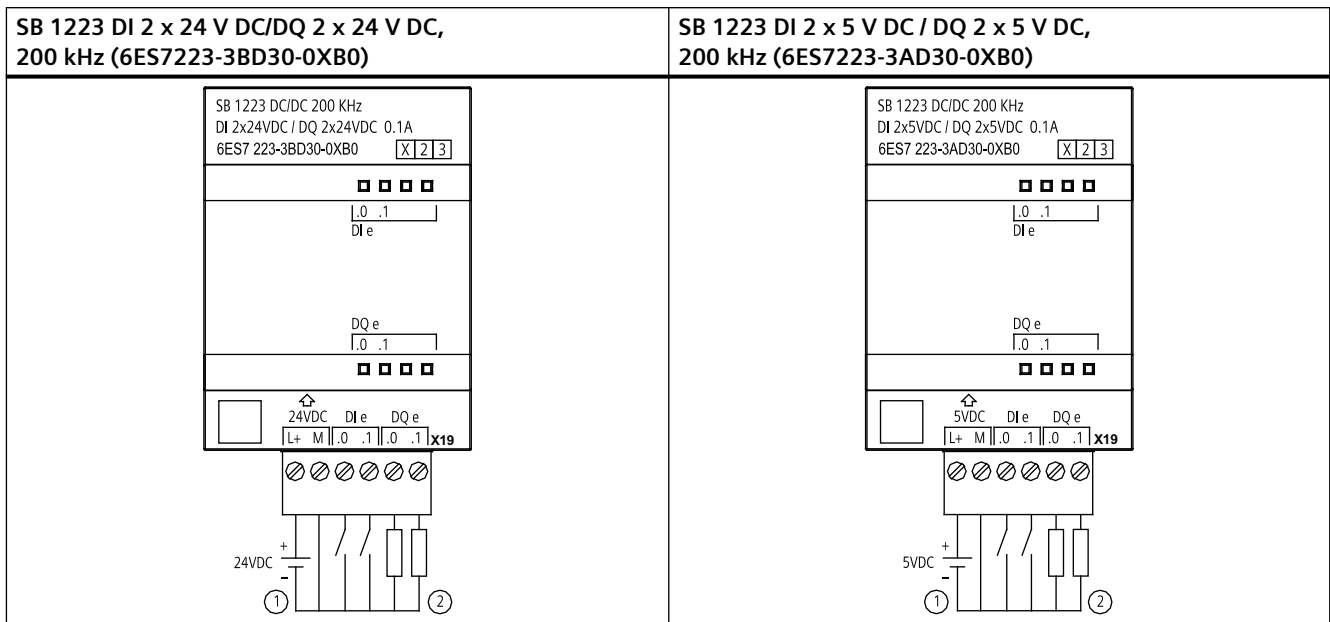
¹ Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Hinweis

Beim Umschalten von Frequenzen über 20 kHz ist es wichtig, dass die Digitaleingänge Rechtecksignale empfangen. Sie haben folgende Möglichkeiten, die Qualität des Eingangssignals zu verbessern:

- Verkürzen Sie die Leitung auf die Mindestlänge.
- Verwenden Sie statt eines nur stromziehenden Treibers einen stromziehenden/ stromliefernden (P-M-schaltenden) Treiber.
- Tauschen Sie das Leitungskabel gegen ein höherwertiges Kabel aus.
- Verringern Sie die Spannung der Schaltkreise/Bauteile von 24 V auf 5 V.
- Fügen Sie am Eingang eine externe Last hinzu.

Tabelle A-202 Schaltpläne der digitalen Eingangs-/Ausgangs-SBs (200 kHz)



- ① Unterstützt nur stromliefernde Eingänge
- ② Bei stromliefernden Ausgängen "Load" an "-" anschließen (s. Abbildung). Bei stromziehenden Ausgängen "Load" an "+" anschließen. ¹ Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben Schaltungen unterstützt werden, ist der aktive Zustand einer stromliefernden Last das Gegenteil von dem einer stromziehenden Last. Ein stromliefernder Ausgang zeigt positive Logik (Q-Bit und LED sind EIN, wenn die Last Stromfluss hat), während ein stromziehender Ausgang negative Logik aufweist (Q-Bit und LED sind AUS, wenn die Last Stromfluss hat). Wenn das Modul ohne Anwenderprogramm gesteckt ist, beträgt der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Hinweis

Achten Sie darauf, dass der Draht der Masseverbindung sicher geerdet ist. Durch den Verlust des Erdanschlusses der Hochgeschwindigkeits-DO-SBs kann es möglicherweise zu genügend Kriechstrom kommen, um eine DC-Last zu aktivieren. Werden die Ausgänge für wichtige DC-Lastanwendungen genutzt, sollte als zusätzliche Sicherheitsvorkehrung eine redundante Erdungsleitung an das SB angeschlossen werden.

Tabelle A-203 Anschlussbelegung für das 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	DO e.0
6	DO e.1

Tabelle A-204 Anschlussbelegung für das 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	DO e.0
6	DO e.1

A.13.4 Technische Daten für das SB 1223 mit 2 x 24-V-DC-Eingang / 2 x 24-V-DC-Ausgang

Tabelle A-205 Allgemeine technische Daten

Technische Daten	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Artikelnummer	6ES7223-0BD30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	40 Gramm
Leistungsverlust	1,0 W

A.13 Digitale Signalboards (SBs)

Technische Daten	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Stromaufnahme (SM-Bus)	50 mA
Stromaufnahme (24 V DC)	4 mA / Eingang

Tabelle A-206 Digitale Eingänge

Technische Daten	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC	
Anzahl der Eingänge	2	
Typ	IEC Typ 1 stromziehend	
Nennspannung	24 V DC bei 4 mA, nominal	
Zulässige Dauerspannung	max. 30 V DC	
Stoßspannung	35 V DC für 0,5 s	
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	
Signal logisch 0 (max.)	5 V DC bei 1 mA	
HSC Eingangstaktfrequenzen (max.)	Einphasenzähler: 30 kHz (15 bis 26 V DC) A/B-Zähler: 20 kHz (15 bis 26 V DC)	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Potentialgetrennte Gruppen	1	
Filterzeiten	µs-Einstellungen:	0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
	ms-Einstellungen	0,05 / 0,1 / 0,2 / 0,4 / 0,8 / 1,6 / 3,2 / 6,4 / 10,0 / 12,8 / 20,0
Anzahl gleichzeitig eingeschalteter Eingänge	2	
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt	

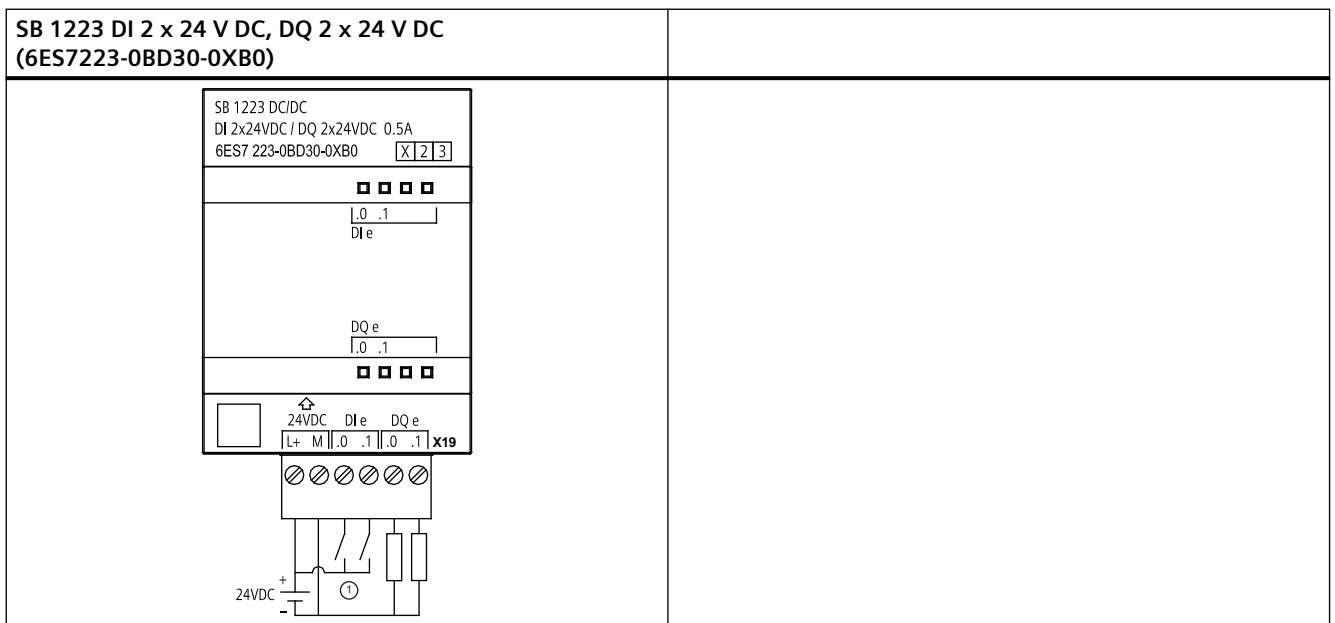
Tabelle A-207 Digitale Ausgänge

Technische Daten	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Ausgänge	2
Ausgangstyp	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	max. 0,1 V DC
Strom (max.)	0,5 A
Lampenlast	5 W
Kontaktwiderstand bei EIN	max. 0,6 Ω
Kriechstrom pro Ausgang	max. 10 µA
Frequenz Impulsfolgeausgang (PTO)	max. 20 kHz, min. 2 Hz ¹
Einschaltstrom	5 A für max. 100 ms
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Potentialgetrennte Gruppen	1
Ströme je Leiter	1 A
Induktive Klemmspannung	L+ minus 48 V, 1 W Verlustleistung

Technische Daten	SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC
Schaltverzögerung	max. 2 µs von Aus nach Ein max. 10 µs von Ein nach Aus
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Ansteuerung eines Digitaleingangs	Ja
Parallele Ausgänge für redundante Laststeuerung	Nein
Parallele Ausgänge für erhöhte Last	Nein
Anzahl gleichzeitig eingeschalteter Ausgänge	2
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt

¹ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

Tabelle A-208 Schaltplan des digitalen Eingangs-/Ausgangs-SBs



① Unterstützt nur stromziehende Eingänge

Tabelle A-209 Anschlussbelegung für das SB 1223 DI mit 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	DO e.0
6	DO e.1

A.14 Analoge Signalboards (SBs)

A.14.1 Technische Daten des SB 1231 1 Analogeingang

Hinweis

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A-210 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 12 Bit
Artikelnummer	6ES7231-4HA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	35 Gramm
Leistungsverlust	0,4 W
Stromaufnahme (SM-Bus)	55 mA
Stromaufnahme (24 V DC)	Keine

Tabelle A-211 Analoge Eingänge

Technische Daten	SB 1231 AI 1x12 Bit
Anzahl der Eingänge	1
Typ	Spannung oder Strom (differential)
Bereich	± 10 V, ± 5 V, $\pm 2,5$ oder 0 bis 20 mA
Auflösung	11 Bit + Vorzeichenbit
Vollausschlag (Datenwort)	-27648 bis 27648
Überbereich/Unterbereich (Datenwort)	Spannung: 32511 bis 27649 / -27649 bis -32512 Strom: 32511 bis 27649 / 0 bis -4864 (siehe Darstellung Analogeingang für Spannung und Darstellung Analogeingang für Strom (Seite 1400)).
Überlauf/Unterlauf (Datenwort)	Spannung: 32767 bis 32512 / -32513 bis -32768 Strom: 32767 bis 32512 / -4865 bis -32768 (siehe Darstellung Analogeingang für Spannung und Darstellung Analogeingang für Strom (Seite 1400)).
Maximale Stehspannung/-strom	± 35 V / ± 40 mA
Glättung	Keine, schwach, mittel oder stark (siehe Antwortzeiten Analogeingang für Schrittantwortzeit (Seite 1400)).
Rauschunterdrückung	400, 60, 50 oder 10 Hz (siehe Ansprechzeiten Analogeingang für Abtastraten (Seite 1400)).
Genauigkeit (25 °C / -20 bis 60 °C)	$\pm 0,3\%$ / $\pm 0,6\%$ des Vollausschlags
Eingangsimpedanz	Spannung: 150 k Ω ; Strom: 250 Ω
Messprinzip	Istwertumwandlung
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz

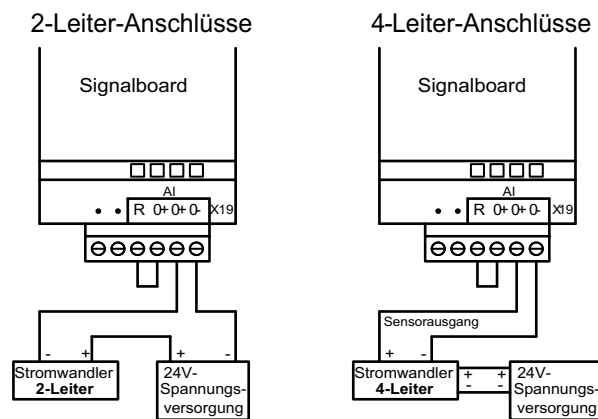
Technische Daten	SB 1231 AI 1x12 Bit
Betriebssignalbereich	Signal- plus Gleichtaktspannung muss kleiner als +35 V und größer als -35 V sein
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt

Tabelle A-212 Diagnose

Technische Daten	SB 1231 AI 1 x 12 Bit
Überlauf/Unterlauf	Ja
24-V-DC-Niederspannung	Nein

Messumformer für SB 1231

Messumformer sind als 2-Draht-Messumformer und als 4-Draht-Messumformer wie nachstehend abgebildet verfügbar.



A.14 Analoge Signalboards (SBs)

Tabelle A-213 Schaltplan des Analogeingangs-SBs

SB 1231 AI x 12 Bit (6ES7231-4HA30-0XB0)	
	<p>① "R" und "0+" für elektrischen Strom anschließen.</p> <p>Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.</p>

Tabelle A-214 Anschlussbelegung für das SB 1231 AI x 12 Bit (6ES7231-4HA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	AI R
4	AI 0+
5	AI 0+
6	AI 0-

A.14.2 Technische Daten des SB 1232 1 Analogausgabe

Tabelle A-215 Allgemeine technische Daten

Technische Daten	SB 1232 AO 1 x 12 Bit
Artikelnummer	6ES7232-4HA30-0XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21
Gewicht	40 Gramm
Leistungsverlust	1,5 W
Stromaufnahme (SM-Bus)	15 mA
Stromaufnahme (24 V DC)	40 mA (ohne Last)

Tabelle A-216 Analoge Ausgänge

Technische Daten	SB 1232 AO 1 x 12 Bit
Ausgänge	1
Typ	Spannung oder Strom
Bereich	± 10 V oder 0 bis 20 mA
Auflösung	Spannung: 12 Bits Strom: 11 Bits
Vollausschlag (Datenwort) Siehe Abschnitt der Ausgangsbereiche für Spannung und Strom (Seite 1401).	Spannung: -27.648 bis 27.648 Strom: 0 bis 27648
Genauigkeit (25 °C / -20 bis 60 °C)	$\pm 0,5\%$ / $\pm 1\%$ des Vollausschlags
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 μ s (R), 750 μ s (1 μ F) Strom: 600 μ s (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: $\geq 1000 \Omega$ Strom: $\leq 600 \Omega$
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt

Tabelle A-217 Diagnose

Technische Daten	SB 1232 AO 1 x 12 Bit
Überlauf/Unterlauf	Ja
Erdschluss (nur Spannungsmodus)	Ja
Drahtbruch (nur Strommodus)	Ja

Tabelle A-218 Schaltplan für das SB 1232 AO 1 x 12 Bit

SB 1232 AQ 1 x 12 Bit (6ES7232-4HA30-0XB0)	
<p>SB 1232 AQ AQ 1x12 BIT +/- 10VDC 0-20mA 6ES7 232-4HA30-0XB0 [X123]</p> <p>AQ 0</p> <p>AQ 0M 0 • • • • X19</p>	
Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.	

A.14 Analoge Signalboards (SBs)

Tabelle A-219 Anschlussbelegung für das SB 1232 AQ 1 x 12 Bit (6ES7232-4HA30-0XB0)

Pin	X19 (vergoldet)
1	AO 0M
2	AO 0
3	Funktionserde
4	Kein Anschluss
5	Kein Anschluss
6	Kein Anschluss

A.14.3 Messbereiche der analogen Eingänge und Ausgänge

A.14.3.1 Schrittantwort der analogen Eingänge

Tabelle A-220 Schrittantwort (ms), 0 V bis 10 V gemessen bei 95 %

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)	Auswahl Integrationszeit			
	400 Hz (2,5 ms)	60 Hz (16,6 ms)	50 Hz (20 ms)	10 Hz (100 ms)
Keine (1 Zyklus): Keine Mittelwertbildung	4,5 ms	18,7 ms	22,0 ms	102 ms
Schwach (4 Zyklen): 4 Abtastwerte	10,6 ms	59,3 ms	70,8 ms	346 ms
Mittel (16 Zyklen): 16 Abtastwerte	33,0 ms	208 ms	250 ms	1240 ms
Stark (32 Zyklen): 32 Abtastwerte	63,0 ms	408 ms	490 ms	2440 ms
Abtastzeit	0,156 ms	1,042 ms	1,250 ms	6,250 ms

A.14.3.2 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A-221 Abtastzeit und Aktualisierungszeit

Auswahl	Abtastzeit	Aktualisierungszeit 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 Messbereiche der analogen Eingänge für Spannung und Strom (SB und SM)

Tabelle A-222 Darstellung Analogeingang für Spannung (SB und SM)

System		Messbereich Spannung				Überlauf
Dezimal	Hexadezimal	±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	Überlauf
32512	7F00					

System		Messbereich Spannung				
Dezimal	Hexadezimal	±10 V	±5 V	±2,5 V	±1,25 V	
32511	7EFF	11,759 V	5,879 V	2,940 V	1,470 V	Überschwingbereich
27649	6C01					
27648	6C00	10 V	5 V	2,5 V	1,250 V	Bemessungsbereich
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	Unterschwingbereich
-27649	93FF					
-32512	8100	-11,759 V	-5,879 V	-2,940 V	-1,470 V	Unterlauf
-32513	80FF					
-32768	8000	-11,851 V	-5,926 V	-2,963 V	-1,481 V	

¹ 7FFF kann aus einem der folgenden Gründe zurückgegeben werden: Überlauf (siehe Tabelle), bevor gültige Werte vorliegen (zum Beispiel unmittelbar beim Hochfahren) oder wenn ein Drahtbruch erkannt wird.

Tabelle A-223 Darstellung Analogeingang für Strom (SB und SM)

System		Messbereich Strom		
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 mA	
32767	7FFF	> 23,52 mA	> 22,81 mA	Überlauf
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Nennbereich
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF			Unterschwingbereich
-4864	ED00	-3,52 mA	1,185 mA	
32767 ¹	7FFF		< 1,185 mA	Drahtbruch (4 bis 20 mA)
-32768	8000	< -3,52 mA		Unterlauf (0 bis 20 mA)

¹ Der Drahtbruchwert von 32767 (16#7FFF) wird unabhängig vom Zustand des Drahtbruchalarms immer zurückgegeben.

A.14.3.4 Messbereiche der analogen Ausgänge für Spannung und Strom (SB und SM)

Tabelle A-224 Darstellung Analogausgang für Spannung (SB und SM)

System		Spannungsausgangsbereich	
Dezimal	Hexadezimal	±10 V	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	

A.14 Analoge Signalboards (SBs)

System		Spannungsausgangsbereich	
Dezimal	Hexadezimal	±10 V	
32511	7EFF	11,76 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Unterschwingbereich
-32512	8100	-11,76 V	
-32513	80FF	Siehe Hinweis 1	Unterlauf
-32768	8000	Siehe Hinweis 1	

¹ Bei Überlauf oder Unterlauf nehmen die Analogausgänge den Ersatzwert für den Betriebszustand STOP an.

Tabelle A-225 Darstellung Analogausgang für Strom (SB und SM)

System		Stromausgangsbereich		
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 mA	
32767	7FFF	Siehe Hinweis 1	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	Siehe Hinweis 1	
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Bemessungsbereich
20736	5100	15 mA	16 mA	
1	1	723,4 nA	4 mA + 578,7 nA	
0	0	0 mA	4 mA	
-1	FFFF		4 mA bis 578,7 nA	
-6912	E500		0 mA	Unterschwingbereich
-6913	E4FF			
-32512	8100			
-32513	80FF	Siehe Hinweis 1	Siehe Hinweis 1	Unterlauf
-32768	8000	Siehe Hinweis 1	Siehe Hinweis 1	

¹ Bei Überlauf oder Unterlauf nehmen die Analogausgänge den Ersatzwert für den Betriebszustand STOP an.

A.14.4 Thermoelement-Signalboards (SBs)

A.14.4.1 Technische Daten des SB 1231 1 Analogeingang Thermoelement

Hinweis

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A-226 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 16 Bit Thermoelement
Artikelnummer	6ES7231-5QA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	35 Gramm
Leistungsverlust	0,5 W
Stromaufnahme (SM-Bus)	5 mA
Stromaufnahme (24 V DC)	20 mA

Tabelle A-227 Analoge Eingänge

Technische Daten	SB 1231 AI 1x16 Bit Thermoelement	
Anzahl der Eingänge	1	
Typ	Potentialfrei, TC und mV	
Bereich	Siehe Filterauswahltabelle Thermoelement (Seite 1404).	
<ul style="list-style-type: none"> • Nennbereich (Datenwort) • Überlauf/Unterlauf (Datenwort) • Überlauf/Unterlauf (Datenwort) 		
Auflösung	Temperaturbereich	0,1 °C/0,1 °F
	Spannung	15 Bit plus Vorzeichen
Max. Stehspannung	±35 V	
Rauschunterdrückung	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz, 400 Hz)	
Gleichtaktunterdrückung	> 120 dB bei 120 V AC	
Impedanz	≥ 10 MΩ	
Genauigkeit	Siehe Thermoelement-Auswahltabelle (Seite 1404).	
Wiederholgenauigkeit	±0,05 % Vollausschlag	
Messprinzip	Integrierend	
Aktualisierungszeit Modul	Siehe Filterauswahltabelle Thermoelement (Seite 1404).	
Fehler kalte Verbindungsstelle	±1,5 °C	
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)	
Leitungslänge (Meter)	Max. 100 m zum Geber	
Leitungswiderstand	max. 100 Ω	

Tabelle A-228 Diagnose

Technische Daten	SB 1231 AI 1 x 16 Bit Thermoelement
Überlauf/Unterlauf ¹	Ja
Drahtbruch ^{2, 3}	Ja

- ¹ Die Informationen der Diagnosealarme "Überlauf" und "Unterlauf" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.
- ² Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.
- ³ Das Modul führt die Drahtbruchprüfung alle 6 Sekunden durch, wodurch die Aktualisierungszeit alle 6 Sekunden für jeden aktivierten Kanal um 9 ms verlängert wird.

Das analoge Thermoelement-Signalmodul SM 1231 TC misst den Wert der an die Moduleingänge angeschlossenen Spannung.

Das analoge Thermoelement-Signalboard SB 1231 misst den Wert der an die Signalboardeingänge angeschlossenen Spannung. Als Temperaturmessart sind entweder "Thermoelement" oder "Spannung" möglich.

- "Thermoelement": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).
- "Spannung": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.

A.14.4.2 Grundlegende Funktionsweise eines Thermoelements

Thermoelemente entstehen, wenn zwei unterschiedliche Metalle elektrisch miteinander verbunden werden. Dadurch wird eine Spannung erzeugt, die proportional zu der Temperatur der Verbindungsstelle ist. Es handelt sich um eine geringe Spannung. Ein Mikrovolt kann viele Grade darstellen. Grundlage für die Temperaturmessung mit Thermoelementen sind das Messen der Spannung eines Thermoelements, das Kompensieren von zusätzlichen Verbindungsstellen und das Linearisieren der Ergebnisse.

Wenn Sie ein Thermoelement an das SM 1231 Thermoelementmodul anschließen, werden die beiden Leitungen der unterschiedlichen Metalle am Signalanschluss des Moduls angeschlossen. Die Stelle, an der die beiden unterschiedlichen Leitungen miteinander verbunden werden, bildet den Sensor des Thermoelements.

Zwei weitere Thermoelemente entstehen an der Stelle, an der die unterschiedlichen Leitungen an den Signalanschluss angeschlossen werden. Die Temperatur des Anschlusses erzeugt eine Spannung, die zu der Spannung des Thermoelementsensors addiert wird. Wird diese Spannung nicht ausgeglichen, weicht die ausgegebene Temperatur von der Temperatur des Sensors ab.

Durch die Kompensation der kalten Verbindungsstelle wird das Thermoelement am Anschluss ausgeglichen. Tabellen für Thermoelemente basieren auf einer Bezugstemperatur an der Verbindungsstelle, üblicherweise Null Grad Celsius. Durch die Kompensation der kalten Verbindungsstelle wird der Anschluss auf Null Grad Celsius kompensiert. Die Spannung, die durch das Thermoelement des Anschlusses addiert wird, wird durch die Kompensation der kalten Verbindungsstelle korrigiert. Die Temperatur des Moduls wird innen gemessen und dann in einen Wert umgewandelt, der zur Sensorumsetzung addiert wird. Die korrigierte Sensorumsetzung wird dann mittels der Thermoelement-Tabellen linearisiert.

Für die optimale Funktionsweise der Kompensation der kalten Verbindungsstelle muss sich das Thermoelementmodul in einer thermisch stabilen Umgebung befinden. Ein langsame Veränderung (weniger als 0,1 °C/Minute) der Temperatur in der Umgebung des Moduls wird

innerhalb der Modulspezifikation korrekt ausgeglichen. Auch Luftbewegungen am Modul verursachen Fehler bei der Kompensation der kalten Verbindungsstelle.

Ist eine bessere Kompensation von Fehlern an der kalten Verbindungsstelle erforderlich, kann eine externe isothermische Klemmenleiste verwendet werden. Mit dem Thermoelementmodul kann eine auf 0 °C bezogene oder eine auf 50 °C bezogene Klemmenleiste eingesetzt werden.

Auswahltabelle für das SB 1231 Thermoelement

Die Bereiche und Genauigkeit der verschiedenen vom SB 1231 Thermoelement-Signalboard unterstützten Thermoelementtypen entnehmen Sie bitte der unten stehenden Tabelle.

Tabelle A-229 Thermoelement-Auswahltabelle

Typ	Minimum für den unteren Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum für den oberen Bereich ²	Normalbereich ^{3,4} Genauigkeit @ 25 °C	Normalbereich ^{1,2,6} Genauigkeit -20 °C bis 60 °C
J	-210,0 °C	-150,0 °C	1.200,0 °C	1.450,0 °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	1.372,0 °C	1.622,0 °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	1.000,0 °C	1.200,0 °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	1.768,0 °C	2.019,0 °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	1.820,0 °C	1.820,0 °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	1.300,0 °C	1.550,0 °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	2.315,0 °C	2.500,0 °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

A.14 Analoge Signalboards (SBs)

Typ	Minimum für den unteren Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum für den oberen Bereich ²	Normalbereich ^{3,4} Genauigkeit @ 25 °C	Normalbereich ^{1,2,6} Genauigkeit -20 °C bis 60 °C
TXK/XK(L)	-200,0 °C	-150,0 °C	800,0 °C	1.050,0 °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
Spannung	-32512	-27648 -80 mV	27648 80 mV	32511	±0,05 %	±0,1 %

- ¹ Die Thermoelementwerte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.
- ² Die Thermoelementwerte oberhalb des Maximums für den oberen Bereich werden als 32767 ausgegeben.
- ³ Der interne Fehler an der kalten Verbindungsstelle beträgt ±1,5 °C für alle Bereiche. Dieser Wert ist zum in dieser Tabelle aufgeführten Fehler zu addieren. Das Modul benötigt eine Aufwärmzeit von mindestens 30 Minuten, bis es spezifikationsgerecht funktioniert.
- ⁴ Bei einer abgestrahlten Funkfrequenz von 970 MHz bis 990 MHz kann sich die Genauigkeit des SM 1231 AI 4 x 16 Bit TC verschlechtern.
- ⁵ Unterer Grenzwert 3276,6 °F mit Meldung in °F
- ⁶ Der Fehler bei der Kompensation an der kalten Verbindungsstelle wurde für Umgebungstemperaturen unter 0 °C nicht charakterisiert und kann den angegebenen Wert überschreiten.

Tabelle A-230 Filterauswahltabelle für das SB 1231 Thermoelement

Unterdrückungsfrequenz (Hz)	Integrationszeit (ms)	Aktualisierungszeit Signalboard (Sekunden)
10	100	0,306
50	20	0,066
60	16,67	0,056
400 ¹	10	0,036

¹ Um die Auflösung und Genauigkeit des Moduls beizubehalten, wenn die 400 Hz Unterdrückung ausgewählt ist, beträgt die Integrationszeit 10 ms. Durch diese Auswahl wird auch Rauschen mit 100 Hz und 200 Hz unterdrückt.

Für die Messung von Thermoelementen wird eine Integrationszeit von 100 ms empfohlen. Niedriger eingestellte Integrationszeiten führen zu einem höheren Wiederholgenauigkeitsfehler der Temperaturmessungen.

Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Gegebenenfalls können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

Sie können diese Logik mit einer Leseabfrage im "Anlauf-OB" implementieren, wodurch der Betrieb gesperrt wird, bis die Initialisierung abgeschlossen ist. Sie müssen die Leseabfrage mit sofortigem Zugriff implementieren. Wenn der Wert der Leseabfrage des Thermoelements 32767 beträgt, muss das Lesen wiederholt werden, bis sich der Wert ändert. Diese Abfrage braucht bei jedem Modul nur für den Eingang mit der größten Nummer des Moduls ausgeführt zu werden (die Moduleingänge werden der Reihe nach von 0 bis 7 initialisiert).

Tabelle A-231 Schaltplan des SB 1231 AI 1 x 16 Thermoelement

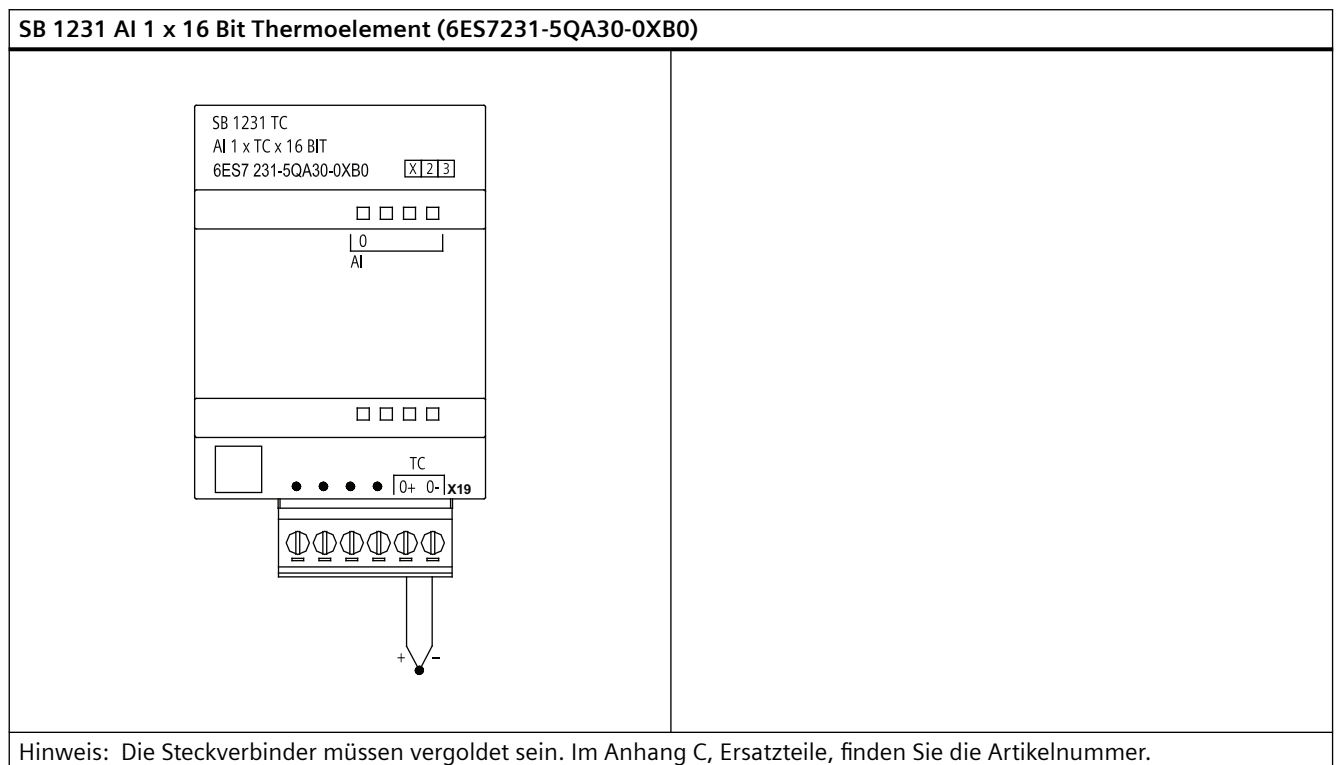


Tabelle A-232 Anschlussbelegung für das SB 1231 AI 1 x 16 Bit Thermoelement (6ES7231-5QA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	Kein Anschluss
4	Kein Anschluss
5	AI 0- /TC
6	AI 0+ /TC

A.14.5 RTD-Signalboards (SBs)

A.14.5.1 Technische Daten des SB 1231 1 Analogeingang RTD

Hinweis

Um diesen SB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A-233 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Artikelnummer	6ES7231-5PA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 2
Gewicht	35 Gramm
Leistungsverlust	0,7 W
Stromaufnahme (SM-Bus)	5 mA
Stromaufnahme (24 V DC)	25 mA

Tabelle A-234 Analoge Eingänge

Technische Daten	SB 1231 AI 1 x 16 Bit RTD	
Anzahl der Eingänge	1	
Typ	Modulreferenz RTD und Ohm	
Bereich	Siehe Auswahltabellen (Seite 1411).	
<ul style="list-style-type: none"> • Nennbereich (Datenwort) • Überlauf/Unterlauf (Datenwort) • Überlauf/Unterlauf (Datenwort) 		
Auflösung	Temperatur	0,1 °C / 0,1 °F
	Spannung	15 Bit plus Vorzeichen

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Max. Stehspannung	±35 V
Rauschunterdrückung	85 dB (10 Hz, 50 Hz, 60 Hz, 400 Hz)
Gleichtaktunterdrückung	> 120 dB
Impedanz	≥ 10 MΩ
Genauigkeit	Siehe Auswahltabellen (Seite 1411).
Wiederholgenauigkeit	±0,05 % Vollausschlag
Maximale Verlustleistung Geber	0.5 m W
Messprinzip	Integrierend
Aktualisierungszeit Modul	Siehe Auswahltable (Seite 1411).
Elektrische Trennung (Feld zu Logik)	707 V DC (Typprüfung)
Leitungslänge (Meter)	Max. 100 m zum Geber
Leitungswiderstand	20 Ω, 2,7 für max. 10 Ω RTD

Tabelle A-235 Diagnose

Technische Daten	SB 1231 AI 1 x 16 Bit RTD
Überlauf/Unterlauf ^{1, 2}	Ja
Drahtbruch ³	Ja

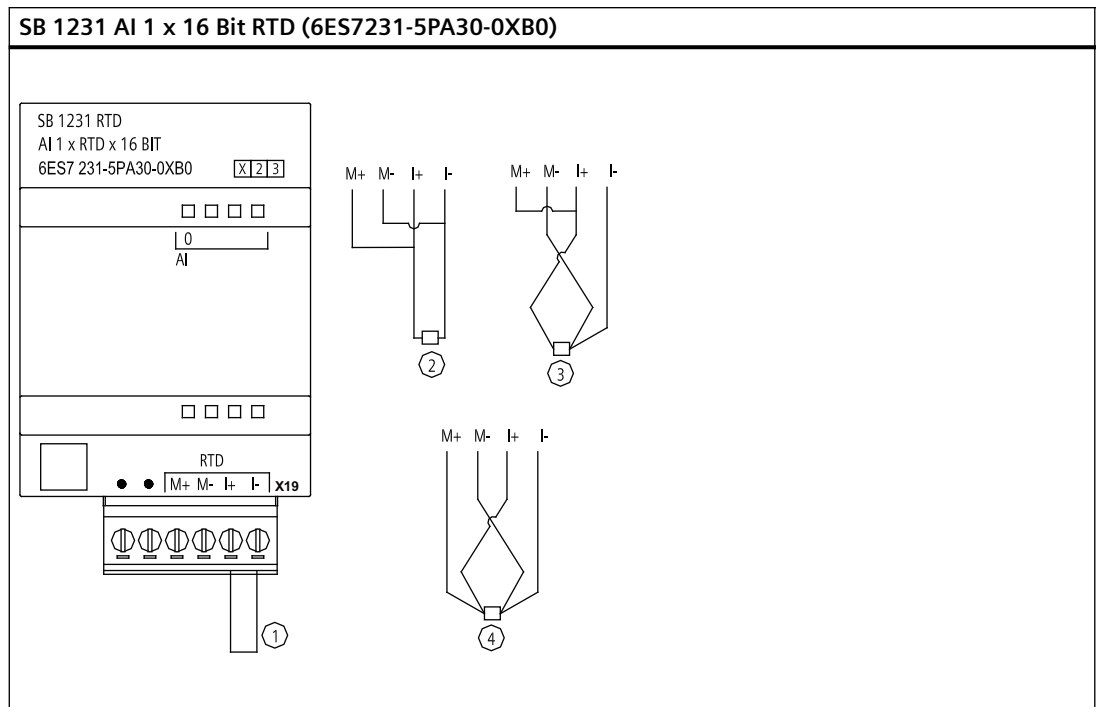
- ¹ Die Informationen der Diagnosealarme "Überlauf" und "Unterlauf" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.
- ² Für Widerstandsmessbereiche ist die Prüfung auf Unterlauf grundsätzlich nicht aktiviert.
- ³ Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Das analoge Thermoelement-Signalboard SM 1231 RTD misst den Wert des an die Signalboardeingänge angeschlossenen Widerstands. Als Messart kann entweder "Widerstand" oder "Thermischer Widerstand" ausgewählt werden.

- "Widerstand": Der Messbereichsendwert im Nennbereich beträgt 27648 dezimal.
- "Thermischer Widerstand": Der Messwert wird in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt). Die Werte des klimatischen Bereichs werden in Grad, multipliziert mit Hundert, ausgegeben (Beispiel: 25,34 Grad werden als Dezimalwert 2534 dargestellt).

Das Signalboard SB 1231 RTD unterstützt Messungen über 2-Leiter-, 3-Leiter- und 4-Leiter-Anschlüsse zum Geberwiderstand.

Tabelle A-236 Schaltplan des SB 1231 AI 1 x 16 Bit RTD



① Nicht belegten RTD-Eingang zurückschleifen

② 2Leiter-RTD

③ 3Leiter-RTD

④ 4Leiter-RTD

Hinweis: Die Steckverbinder müssen vergoldet sein. Im Anhang C, Ersatzteile, finden Sie die Artikelnummer.

Tabelle A-237 Anschlussbelegung für das SB 1231 AI 1 x 16 Bit RTD (6ES7231-5PA30-0XB0)

Pin	X19 (vergoldet)
1	Kein Anschluss
2	Kein Anschluss
3	AI 0 M+ /RTD
4	AI 0 M- /RTD
5	AI 0 I+ /RTD
6	AI 0 I- /RTD

A.14.5.2 Auswahltabellen für das SB 1231 RTD

Tabelle A-238 Bereiche und Genauigkeit für die verschiedenen Geber, die von den RTD-Modulen unterstützt werden

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 100 Klima	-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	1.000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1.000,0 °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	1.000,0 °C	±0,5 °C	±1,0 °C
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1.100,0 °C	1.295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1.100,0 °C	1.295 °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 Die RTD-Werte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.

² Die RTD-Werte oberhalb des Maximums für den oberen Bereich werden als +32768 ausgegeben.

Tabelle A-239 Beständigkeit gegen:

Bereich	Minimum unterer Bereich	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ¹	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich -20 °C bis 60 °C
150 Ω	nicht zutreffend	0 (0 Ω)	27648 (150 Ω)	176,383 Ω	±0,05 %	±0,1 %
300 Ω	nicht zutreffend	0 (0 Ω)	27648 (300 Ω)	352,767 Ω	±0,05 %	±0,1 %
600 Ω	nicht zutreffend	0 (0 Ω)	27648 (600 Ω)	705,534 Ω	±0,05 %	±0,1 %

¹ Die Widerstandswerte oberhalb des Maximums für den oberen Bereich werden als 32767 ausgegeben.

Hinweis

Für alle aktivierte Kanäle ohne angeschlossenen Geber meldet das Modul 32767. Wenn außerdem die Prüfung auf offene Leitungen aktiviert ist, blinken am Modul die entsprechenden roten LED.

Optimale Genauigkeit für die 10 Ω-RTD-Bereiche ermöglichen 4-Leiter-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Leiter-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

Tabelle A-240 Rauschminderung und Aktualisierungszeiten für die RTD-Module

Auswahl Unterdrückungsfrequenz	Integrationszeit	1-kanaliges 4/2-Draht-Modul Aktualisierungszeit (Sekunden)	1-kanaliges 3-Draht-Modul Aktualisierungszeit (Sekunden)
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 Um bei ausgewähltem 400-Hz-Filter die Auflösung und Genauigkeit des Moduls beizubehalten, beträgt die Integrationszeit 10 ms. Durch diese Auswahl wird auch Rauschen mit 100 Hz und 200 Hz unterdrückt.

Hinweis

Nach dem Einschalten führt das Modul die interne Kalibrierung für den A/D-Wandler durch. In diesem Zeitraum meldet das Modul auf jedem Kanal den Wert 32767, bis für den jeweiligen Kanal gültige Daten vorliegen. Diese Initialisierungszeit muss im Anwenderprogramm ggf. berücksichtigt werden. Weil sich die Konfiguration des Moduls auf die Dauer der Initialisierungszeit auswirken kann, prüfen Sie das Verhalten des Moduls in Ihrer Konfiguration. Ggf. können Sie Logik in Ihr Anwenderprogramm aufnehmen, um die Initialisierungszeit des Moduls unterzubringen.

Sie können diese Logik mit einer Leseabfrage im "Anlauf-OB" implementieren, wodurch der Betrieb gesperrt wird, bis die Initialisierung abgeschlossen ist. Sie müssen die Leseabfrage mit sofortigem Zugriff implementieren. Wenn der Wert der Leseabfrage des Widerstandstemperatursensors 32767 beträgt, muss das Lesen wiederholt werden, bis sich der Wert ändert. Diese Abfrage braucht bei jedem Modul nur für den Eingang mit der größten Nummer des Moduls ausgeführt zu werden (die Moduleingänge werden der Reihe nach von 0 bis 7 initialisiert).

A.15 BB 1297 Batterieboard

BB 1297 Batterieboard

Das S7-1200 BB 1297 Batterieboard dient der langfristigen Pufferung der Echtzeituhr. Es kann in den Steckplatz des Signalboards der S7-1200 CPU (ab Firmware 3.0) gesteckt werden. Sie müssen das BB 1297 in die Gerätekonfiguration aufnehmen und die Hardwarekonfiguration in die CPU laden, damit das BB funktionsfähig ist.

Die Batterie (Typ CR1025) ist nicht im Lieferumfang des BB 1297 enthalten und ist separat zu erwerben.

Hinweis

Das BB 1297 ist mechanisch für die CPUs mit Firmware ab Version 3.0 ausgelegt.

Verwenden Sie das BB 1297 nicht mit CPUs früherer Versionen, weil sich der Stecker des BB 1297 nicht in die CPU stecken lässt.


 WARNUNG
Die Installation einer nicht spezifizierten Batterie im BB 1297 oder der Anschluss einer nicht spezifizierten Batterie an den Schaltkreis auf andere Weise kann zu Brand oder Bauteilbeschädigung und unvorhersehbarem Betrieb von Maschinen führen.
Fire or unpredictable operation of machinery can result in death, severe personal injury, or property damage.
Verwenden Sie für die Pufferung der Echtzeituhr nur die spezifizierte Batterie CR1025.

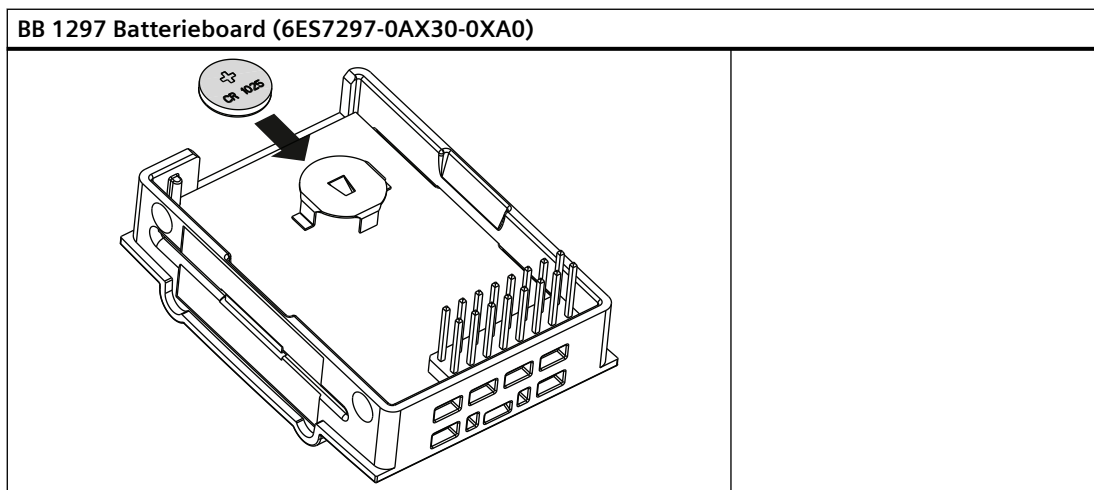
Tabelle A-241 Allgemeine technische Daten

Technische Daten	BB 1297 Batterieboard
Artikelnummer	6ES7297-0AX30-0XA0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	28 Gramm
Leistungsverlust	0,5 W
Stromaufnahme (SM-Bus)	11 mA
Stromaufnahme (24 V DC)	Keine

Batterie (nicht enthalten)	BB 1297 Batterieboard
Netzausfallüberbrückung	Ca. 1 Jahr
Batterietyp	CR1025 Siehe Einsetzen oder Austauschen der Batterie des Batterieboards BB 1297 (Seite 52)
Nennspannung	3 V
Nennkapazität	Mindestens 30 mAH

Diagnose	BB 1297 Batterieboard
Kritischer Batteriezustand	< 2,5 V
Batteriediagnose	Anzeige bei geringer Spannung: <ul style="list-style-type: none"> • Geringe Batteriespannung bewirkt, dass die MAINT-LED der CPU ständig bernsteinfarben leuchtet. • Diagnosepufferereignis: 16#06:2700 "Submodulwartung erforderlich: mindestens eine Batterie ist erschöpft (BATTF)"
Batteriezustand	Batteriezustandsbit 0 = Batterie OK 1 = Batterie schwach
Aktualisierung des Batteriezustands	Der Batteriezustand wird beim Anlauf und dann einmal täglich aktualisiert, wenn sich die CPU im Betriebszustand RUN befindet.

Tabelle A-242 Einbauabbildung für das Batterieboard BB 1297



A.16 Kommunikationsschnittstellen

A.16.1 PROFIBUS

A.16.1.1 CM 1242-5 PROFIBUS DP-SLAVE

Tabelle A-243 Technische Daten des CM 1242-5

Technische Daten	
Artikelnummer	6GK7242-5DX30-0XE0
Schnittstellen	

Technische Daten	
Anschluss an PROFIBUS	9-polige Sub-D-Buchse
Maximale Stromaufnahme an der PROFIBUS-Schnittstelle beim Anschluss von Netzkomponenten (beispielsweise optische Netzkomponenten)	15 mA bei 5 V (nur für den Busabschluss) *)
Zulässige Umgebungsbedingungen	
Umgebungstemperatur <ul style="list-style-type: none"> während der Lagerung während des Transports während des Betriebs bei vertikalem Einbau (Hutschiene horizontal) während des Betriebs bei horizontalem Einbau (Hutschiene vertikal) 	<ul style="list-style-type: none"> -40 °C bis 70 °C -40 °C bis 70 °C 0 °C bis 55 °C 0 °C bis 45 °C
Relative Luftfeuchtigkeit bei 25 °C im Betrieb, ohne Kondensation, max.	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme und Verlustleistung	
Art der Spannungsversorgung	DC
Spannungsversorgung aus dem Rückwandbus	5 V
Stromaufnahme (typisch)	150 mA
Effektive Verlustleistung (typisch)	0,75 W
Potentialtrennung <ul style="list-style-type: none"> PROFIBUS-Schnittstelle zur Masse PROFIBUS-Schnittstelle zur internen Schaltung 	710 V AC für 1 Minute
Abmessungen und Gewichte	
<ul style="list-style-type: none"> Breite Höhe Tiefe 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm
Gewicht <ul style="list-style-type: none"> Nettogewicht Gewicht mit Verpackung 	<ul style="list-style-type: none"> 115 g 152 g

*)Die Strombelastung durch einen externen Verbraucher, der zwischen VP (Pin 6) und DGND (Pin 5) angeschlossen wird, darf für den Busabschluss maximal 15 mA betragen (kurzschlussfest).

A.16.1.2 Anschlussbelegung der Sub-D-Buchse des CM 1242-5

PROFIBUS-Schnittstelle

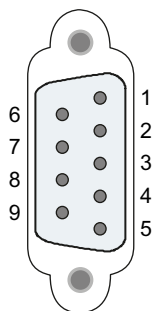


Tabelle A-244 Anschlussbelegung der Sub-D-Buchse

Pin	Beschreibung	Pin	Beschreibung
1	- nicht belegt -	6	P5V2: Spannungsversorgung +5V
2	- nicht belegt -	7	- nicht belegt -
3	RxD/TxD-P: Datenleitung B	8	RxD/TxD-N: Datenleitung A
4	RTS	9	- nicht belegt -
5	M5V2: Datenbezugspotential (Masse DGND)	Gehäuse	Masseanschluss

A.16.1.3 CM 1243-5 PROFIBUS DP-Master

Tabelle A-245 Technische Daten des CM 1243-5

Technische Daten	
Artikelnummer	6GK7243-5DX30-0XE0
Schnittstellen	
Anschluss an PROFIBUS	9-polige Sub-D-Buchse
Maximale Stromaufnahme an der PROFIBUS-Schnittstelle beim Anschluss von Netzkomponenten (beispielsweise optische Netzkomponenten)	15 mA bei 5 V (nur für den Busabschluss) *)
Zulässige Umgebungsbedingungen	
Umgebungstemperatur	<ul style="list-style-type: none"> • während der Lagerung • während des Transports • während des Betriebs bei vertikalem Einbau (Hutschiene horizontal) • während des Betriebs bei horizontalem Einbau (Hutschiene vertikal)
	<ul style="list-style-type: none"> • -40 °C bis 70 °C • -40 °C bis 70 °C • 0 °C bis 55 °C • 0 °C bis 45 °C

Technische Daten	
Relative Luftfeuchtigkeit bei 25 °C im Betrieb, ohne Kondensation, max.	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme und Verlustleistung	
Art der Spannungsversorgung	DC
Spannungsversorgung / extern	24 V
<ul style="list-style-type: none"> • minimal • maximal 	<ul style="list-style-type: none"> • 19,2 V • 28,8 V
Stromaufnahme (typisch)	
<ul style="list-style-type: none"> • aus 24 V DC • aus dem S7-1200 Rückwandbus 	<ul style="list-style-type: none"> • 100 mA • 0 mA
Effektive Verlustleistung (typisch)	
<ul style="list-style-type: none"> • aus 24 V DC • aus dem S7-1200 Rückwandbus 	<ul style="list-style-type: none"> • 2,4 W • 0 W
Fremdeinspeisung 24 V DC	
<ul style="list-style-type: none"> • Min. Leitungsquerschnitt • Max. Leitungsquerschnitt • Anzugsmoment der Schraubklemmen 	<ul style="list-style-type: none"> • min.: 0,14 mm² (AWG 25) • max.: 1,5 mm² (AWG 15) • 0,45 Nm (4 lb-in)
Potentialtrennung	710 V AC für 1 Minute
<ul style="list-style-type: none"> • PROFIBUS-Schnittstelle zur Masse • PROFIBUS-Schnittstelle zur internen Schaltung 	
Abmessungen und Gewichte	
<ul style="list-style-type: none"> • Breite • Höhe • Tiefe 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm
Gewicht	
<ul style="list-style-type: none"> • Nettogewicht • Gewicht mit Verpackung 	<ul style="list-style-type: none"> • 134 g • 171 g

*)Die Strombelastung durch einen externen Verbraucher, der zwischen VP (Pin 6) und DGND (Pin 5) angeschlossen wird, darf für den Busabschluss maximal 15 mA betragen (kurzschlussfest).

Hinweis

Das CM 1243-5 (PROFIBUS-Mastermodul) muss über die 24-V-DC-Geberversorgung der CPU gespeist werden.

A.16.1.4 Anschlussbelegung der Sub-D-Buchse des CM 1243-5

PROFIBUS-Schnittstelle

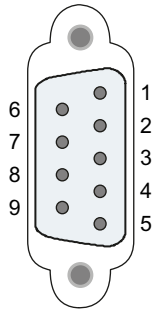


Tabelle A-246 Anschlussbelegung der Sub-D-Buchse

Pin	Beschreibung	Pin	Beschreibung
1	- nicht belegt -	6	VP: Spannungsversorgung +5 V nur für Busabschlusswiderstände; nicht für die Versorgung externer Geräte
2	- nicht belegt -	7	- nicht belegt -
3	RxD/TxD-P: Datenleitung B	8	RxD/TxD-N: Datenleitung A
4	CNTR-P: RTS	9	- nicht belegt -
5	DGND: Masse für Datensignale und VP	Gehäuse	Masseanschluss

PROFIBUS-Kabel

Hinweis

Auflegen der Schirmung des PROFIBUS-Kabels

Der Schirm des PROFIBUS-Kabels muss aufgelegt werden.

Isolieren Sie hierzu das PROFIBUS-Kabel am Ende ein Stück ab und verbinden Sie den Schirm mit der Funktionserde.

A.16.2 CP 1242-7

Hinweis

Der CP 1242-7 hat keine Zulassung für Anwendungen für das Seewesen

Der CP 1242-7 hat keine Zulassung für das Seewesen.

Hinweis

Um diese Module nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

A.16.2.1 CP 1242-7 GPRS

Tabelle A-247 Technische Daten des CP 1242-7 GPRS V2

Technische Daten	
Artikelnummer	6GK7242-7KX3-0XE0
Funkschnittstelle	
Antennenanschluss	SMA-Buchse
Nennimpedanz	50 Ohm
Funkverbindung	
Maximale Sendeleistung	<ul style="list-style-type: none"> • GSM 850, Class 4: +33 dBm ±2dBm • GSM 900, Class 4: +33 dBm ±2dBm • GSM 1800, Class 1: +30 dBm ±2dBm • GSM 1900, Class 1: +30 dBm ±2dBm
GPRS	Multislot-Klasse 10 Endgerätekategorie B Codierungsschema 1...4 (GMSK)
SMS	Betriebsmodus abgehend: MO Dienst: Punkt zu Punkt
Zulässige Umgebungsbedingungen	
Umgebungstemperatur	<ul style="list-style-type: none"> • während der Lagerung • während des Transports • während des Betriebs bei vertikalem Einbau (Hutschiene horizontal) • während des Betriebs bei horizontalem Einbau (Hutschiene vertikal)
	<ul style="list-style-type: none"> • -40 °C bis 70 °C • -40 °C bis 70 °C • 0 °C bis 55 °C • 0 °C bis 45 °C
Relative Luftfeuchtigkeit bei 25 °C im Betrieb, ohne Kondensation, max.	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme und Verlustleistung	
Art der Spannungsversorgung	DC
Spannungsversorgung / extern	24 V
<ul style="list-style-type: none"> • minimal • maximal 	<ul style="list-style-type: none"> • 19,2 V • 28,8 V
Stromaufnahme (typisch)	<ul style="list-style-type: none"> • aus 24 V DC • aus dem S7-1200 Rückwandbus
	<ul style="list-style-type: none"> • 100 mA • 0 mA

Technische Daten	
Effektive Verlustleistung (typisch)	<ul style="list-style-type: none"> aus 24 V DC aus dem S7-1200 Rückwandbus
Stromversorgung 24 V DC	<ul style="list-style-type: none"> Min. Leitungsquerschnitt Max. Leitungsquerschnitt Anzugsmoment der Schraubklemmen
Potentialtrennung Netzteil gegen interne Schaltung	710 V AC für 1 Minute
Abmessungen und Gewichte	
<ul style="list-style-type: none"> Breite Höhe Tiefe 	<ul style="list-style-type: none"> 30 mm 100 mm 75 mm
Gewicht	
<ul style="list-style-type: none"> Nettogewicht Gewicht mit Verpackung 	<ul style="list-style-type: none"> 133 g 170 g

Hinweis

Verhindern von Störung der CPU durch Antennen

Es kann zur Störung der CPU kommen, wenn sich eine Antenne in zu großer Nähe befindet oder wenn Sie nicht empfohlene Antennen verwenden. Empfohlene Antennen finden Sie unter Kompaktbetriebsanleitung Antenne ANT794-4MR für LTE/UMTS/GSM (<https://support.industry.siemens.com/cs/ww/de/view/23119005>) (nur in Deutsch und Englisch verfügbar).

A.16.2.2 GSM/GPRS-Antenne ANT794-4MR

Technische Daten der GSM/GPRS-Antenne ANT794-4MR

ANT794-4MR	
Artikelnummer	6NH9860-1AA00
Mobilfunknetze	GSM/GPRS
Frequenzbereiche	<ul style="list-style-type: none"> 824 bis 960 MHz (GSM 850, 900) 1710 bis 1880 MHz (GSM 1800, 900) 1900 bis 2200 MHz (GSM / UMTS)
Charakteristik	omnidirektional
Antennengewinn	0 dB
Impedanz	50 Ohm
Stehwellenverhältnis (SWR)	< 2,0
Max. Leistung	20 W
Polarität	linear vertikal

ANT794-4MR	
Stecker	SMA
Länge Antennenkabel	5 m
Außenmaterial	Hart-PVC, UV-beständig
Schutzart	IP20
Zulässige Umgebungsbedingungen	<ul style="list-style-type: none"> • Betriebstemperatur • Transport-/Lagertemperatur • Relative Luftfeuchtigkeit
	<ul style="list-style-type: none"> • -40 °C bis +70 °C • -40 °C bis +70 °C • 100 %
Außenmaterial	Hart-PVC, UV-beständig
Konstruktiver Aufbau	Antenne mit 5 m fest verbundenem Kabel und SMA-Stecker
Maße (T x H) in mm	25 x 193
Gewicht	
<ul style="list-style-type: none"> • Antenne mit Kabel • Montageteile 	<ul style="list-style-type: none"> • 310 g • 54 g
Montage	Mit mitgeliefertem Haltewinkel

A.16.2.3 Flachantenne ANT794-3M

Technische Daten der Flachantenne ANT794-3M

ANT794-3M		
Artikelnummer	6NH9870-1AA00	
Mobilfunknetze	GSM 900	GSM 1800/1900
Frequenzbereiche	890 - 960 MHz	1710 - 1990 MHz
Stehwellenverhältnis (VSWR)	≤ 2:1	≤ 1,5:1
Rückflussdämpfung (Tx)	≈ 10 dB	≈ 14 dB
Antennengewinn	0 dB	
Impedanz	50 Ohm	
Max. Leistung	10 W	
Antennenkabel	HF-Kabel RG 174 (fest angeschlossen) mit SMA-Stecker	
Kabellänge	1,2 m	
Schutzart	IP64	
Zulässiger Temperaturbereich	-40 °C bis +75 °C	
Entflammbarkeit	UL 94 V2	
Außenmaterial	ABS Polylac PA-765, lichtgrau (RAL 7035)	
Maße (B x L x H) in mm	70,5 x 146,5 x 20,5	
Gewicht	130 g	

A.16.3 CM 1243-2 AS-i-Master**A.16.3.1 Technische Daten des AS-i Master CM 1243-2**

Tabelle A-248 Technische Daten des AS-i Master CM 1243-2

Technische Daten	
Artikelnummer	3RK7243-2AA30-0XB0
Firmwareversion	V1.0
Datum	01.12.2011
Schnittstellen	
Maximale Stromaufnahme aus dem S7-1200 Rückwandbus aus der AS-i Leitung	max. 250 mA, Versorgungsspannung S7-1200 Kommunikationsbus 5 V DC max. 100 mA
Maximale Strombelastbarkeit zwischen ASI+/ASI-Klemmen	8 A
Anschlussbelegung	Siehe Kapitel Elektrischer Anschluss des AS-i-Masters (Seite 1423)
Leiterquerschnitt	0,2 mm ² (AWG 24) ... 3,3 mm ² (AWG 12)
Anzugsmoment ASI-Stecker	0,56 Nm
Zulässige Umgebungsbedingungen	
Umgebungstemperatur während der Lagerung	-40 °C ... 70 °C
während des Transports	-40 °C ... 70 °C
während des Betriebs bei vertikalem Einbau (Standard-Hutschiene horizontal)	0 °C ... 55 °C
während des Betriebs bei horizontalem Einbau (Standard-Hutschiene vertikal)	0 °C ... 45 °C
Relative Luftfeuchtigkeit bei 25 °C im Betrieb, ohne Kondensation, max.	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme, Verlustleistung	
Art der Spannungsversorgung	DC
Stromaufnahme (typisch) aus dem S7-1200 Rückwandbus	200 mA
Gesamtverlustleistung (typisch):	
• aus dem S7-1200 Rückwandbus	1 W
• aus der AS-i Leitung	2,4 W
Abmessungen und Gewichte	

Technische Daten	
Breite	30 mm
Höhe	100 mm
Tiefe	75 mm
Gewicht	
Nettogewicht	122 g
Gewicht mit Verpackung	159 g

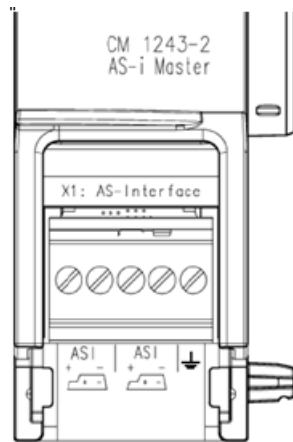
A.16.3.2 Elektrischer Anschluss des AS-i-Masters

Spannungsversorgung des AS-i-Masters CM 1243-2

Der AS-i-Master CM 1243-2 wird über den Kommunikationsbus der S7-1200 mit Spannung versorgt. Somit kann auch bei Ausfall der AS-i-Versorgungsspannung eine Diagnosemeldung an die S7-1200 gesendet werden. Der Anschluss an den Kommunikationsbus befindet sich rechts am AS-i-Master CM 1243-2.

Anschlussklemmen AS-Interface

Die abnehmbare Klemme zum Anschluss der AS-i-Leitung befindet sich hinter der unteren Abdeckklappe auf der Vorderseite des AS-i-Masters CM 1243-2.



Bei Verwendung der AS-i-Profilleitung ist die richtige Polarität der Anschlussleitung durch das Symbol ersichtlich



Informationen zum Ausbau und Wiedereinbau des Klemmenblocks finden Sie im Kapitel "Einbau" (Seite 57).

Hinweis


Maximale Belastbarkeit der Anschlusskontakte

Die Belastbarkeit der Anschlusskontakte beträgt max. 8 A. Falls dieser Wert auf der AS-i-Leitung überschritten wird, darf der AS-i-Master CM 1243-2 nicht in die AS-i-Leitung "eingeschleift" werden, sondern muss über eine Stichleitung angeschlossen werden (nur ein Anschlusspaar des AS-i-Masters CM 1243-2 belegt).

Bitte beachten Sie zudem, dass bei Stromführung durch den AS-i-Master und Strömen über 4 Ampère die verwendeten Leitungen für eine Einsatztemperatur von mindestens 75 °C geeignet sein müssen.

Weitere Hinweise zum Anschluss der AS-i-Leitung finden Sie im Kapitel "Montage, Anschluss und Inbetriebnahme der Baugruppen" im Handbuch "AS-i Master CM 1243-2 und AS-i Datenentkopplungsmodul DCM 1271 für SIMATIC S7-1200".

Anschlussbelegung

Beschriftung	Bedeutung
ASI+	AS-i-Anschluss – positive Polarität
ASI-	AS-i-Anschluss – negative Polarität
	Funktionserde

A.16.4 RS232, RS422 und RS485

A.16.4.1 Technische Daten des CB 1241 RS485

Hinweis

Um dieses CB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A-249 Allgemeine technische Daten

Technische Daten	CB 1241 RS485
Artikelnummer	6ES7241-1CH30-1XB0
Abmessungen (B x H x T) (mm)	38 x 62 x 21
Gewicht	40 Gramm

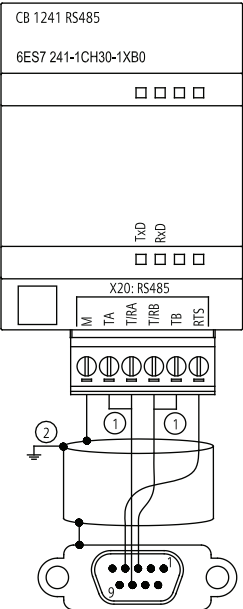
Tabelle A-250 Sender und Empfänger

Technische Daten	CB 1241 RS485
Typ	RS485 (2-adrig halbduplex)
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert
Differentialausgangsspannung Sender	min. 2 V bei $R_L = 100 \Omega$ min. 1,5 V bei $R_L = 54 \Omega$
Abschluss und Bias	10 k zu +5 V an B, RS485 Pin 3 10 k zu GND an A, RS485 Pin 4
Optionaler Abschluss	Pin TB an Pin T/RB kurzschließen, effektive Abschlussimpedanz beträgt 127 Ω , Anschluss an RS485 Pin 3 Pin TA an Pin T/RA kurzschließen, effektive Abschlussimpedanz beträgt 127 Ω , Anschluss an RS485 Pin 4
Eingangsimpedanz Empfänger	min. 5,4 k Ω einschließlich Abschluss
Ansprechgrenze/Sensibilität Empfänger	min. +/- 0,2 V, 60 mV typ. Hysterese
Potentialtrennung RS485-Signal zu Masse RS485-Signal zu CPU-Logik	707 V DC (Typprüfung)
Leitungslänge (geschirmt)	max. 1000 m
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	Nicht unterstützt
Wartezeit	0 bis 65535 ms

Tabelle A-251 Stromversorgung

Technische Daten	CB 1241 RS485
Verlustleistung	1,5 W
Stromaufnahme (SM-Bus), max.	50 mA
Max. Stromaufnahme (24 V DC)	80 mA

A.16 Kommunikationsschnittstellen

<p>CB 1241 RS485 (6ES7241-1CH30-1XB0)</p> 	
<p>① "TA" und TB" wie gezeigt anschließen, um das Netzwerk abzuschließen. (Nur die Endgeräte im RS485-Netz abschließen.)</p> <p>② Verwenden Sie geschirmte, verdrehte Leiterpaare und schließen Sie den Kabelschirm an Erde an.</p>	

Sie schließen nur die zwei Enden des RS485-Netzes ab. Die Geräte zwischen den beiden Endgeräten werden nicht abgeschlossen. Weitere Informationen finden Sie unter "Abschließen eines RS485-Busanschlusssteckers" (Seite 942).

Tabelle A-252 Anschlussbelegung für das CB 1241 RS485 (6ES7241-1CH30-1XB0)

Pin	9-poliger Steckverbinder	X20
1	RS485 / logische Erde	--
2	RS485 / frei	--
3	RS485 / TxD+	4 - T/RB
4	RS485 / RTS	6 - RTS
5	RS485 / logische Erde	--
6	RS485 / 5-V-Leistung	--
7	RS485 / frei	--
8	RS485 / TxD-	3 - T/RA
9	RS485 / frei	--
Hülse		1 - M

A.16.4.2 Technische Daten des CM 1241 RS232

Tabelle A-253 Allgemeine technische Daten

Technische Daten	CM 1241 RS232
Artikelnummer	6ES7241-1AH32-0XB0
Abmessungen (mm)	30 x 100 x 75
Gewicht	150 Gramm

Tabelle A-254 Sender und Empfänger

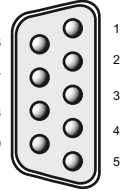
Technische Daten	CM 1241 RS232
Typ	RS232 (Vollduplex)
Ausgangsspannung Sender	min. +/- 5 V bei $R_L = 3 \text{ k}\Omega$
Ausgangsspannung Empfänger	max. +/- 15 V DC
Eingangsimpedanz Empfänger	min. $3 \text{ k}\Omega$
Ansprechgrenze/Sensibilität Empfänger	min. 0,8 V Low, max. 2,4 V High, typ. Hysterese 0,5 V
Empfänger-Eingangsspannung	max. +/- 30 V DC
Potentialtrennung RS-232-Signal zu Masse RS-232-Signal zu CPU-Logik	707 V DC (Typprüfung)
Leitungslänge (geschirmt)	max. 10 m
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	Hardware, Software
Wartezeit	0 bis 65535 ms

Tabelle A-255 Stromversorgung

Technische Daten	CM 1241 RS232
Verlustleistung	1 W
Aus +5 V DC	200 mA

A.16 Kommunikationsschnittstellen

Tabelle A-256 RS232-Steckverbinder (Stecker)

Pin	Beschreibung	Steckverbinder (Stecker)	Pin	Beschreibung
1 DCD	Datenträgererkennung: Eingang		6 DSR	Datensatz bereit: Eingang
2 RxD	Daten von DCE empfangen: Eingang		7 RTS	Sendeanforderung: Ausgang
3 TxD	Daten an DCE gesendet: Ausgang		8 CTS	Bereit zum Senden: Eingang
4 DTR	Datenterminal bereit: Ausgang		9 RI	Rufanzeige (nicht verwendet)
5 GND	Logikmasse		SHELL	Erdungsanschluss

A.16.4.3 Technische Daten des CM 1241 RS422/485

Technische Daten des CM 1241 RS422/485

Tabelle A-257 Allgemeine technische Daten

Technische Daten	CM 1241 RS422/485
Artikelnummer	6ES7241-1CH32-0XB0
Abmessungen B x H x T (mm)	30 x 100 x 75
Gewicht	155 Gramm

Tabelle A-258 Sender und Empfänger

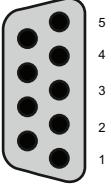
Technische Daten	CM 1241 RS422/485
Typ	RS422 oder RS485, 9-polige Sub-D-Buchse
Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert
Differentialausgangsspannung Sender	min. 2 V bei $R_L = 100 \Omega$ min. 1,5 V bei $R_L = 54 \Omega$
Abschluss und Bias	10 k Ω zu +5 V an B, PROFIBUS Pin 3 10 k Ω zu GND an A, PROFIBUS Pin 8 Interne Bias-Optionen vorhanden oder kein interner Bias. In allen Fällen ist der externe Abschluss erforderlich, siehe Abschließen eines RS485-Busanschlusses (Seite 942) und Konfigurieren von RS422 und RS485 im Systemhandbuch S7-1200 Automatisierungssystem (Seite 993).
Eingangsimpedanz Empfänger	min. 5,4 k Ω einschließlich Abschluss
Ansprechgrenze/Sensibilität Empfänger	min. +/- 0,2 V, 60 mV typ. Hysterese
Potentialtrennung RS485-Signal zu Masse RS485-Signal zu CPU-Logik	707 V DC (Typprüfung)
Leitungslänge (geschirmt)	max. 1000 m (abhängig von der Baudrate)
Baudrate	300 Baud, 600 Baud, 1,2 kBit/s, 2,4 kBit/s, 4,8 kBit/s, 9,6 kBit/s (Standard), 19,2 kBit/s, 38,4 kBit/s, 57,6 kBit/s, 76,8 kBit/s, 115,2 kBit/s
Parität	Keine Parität (Standard), gerade, ungerade, Mark (Paritätsbit immer auf 1), Space (Paritätsbit immer auf 0)

Technische Daten	CM 1241 RS422/485
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	XON/XOFF wird im RS422-Modus unterstützt
Wartezeit	0 bis 65535 ms

Tabelle A-259 Stromversorgung

Technische Daten	CM 1241 RS422/485
Verlustleistung	1,1 W
Aus +5 V DC	220 mA

Tabelle A-260 RS485- oder RS422-Steckverbinder (Buchse)

Pin	Beschreibung	Steckverbinder (Buchse)	Pin	Beschreibung
1	Logik- oder Kommunikationsmasse		6 PWR	+5 V mit 100 Ohm Reihenwiderstand: Ausgang
2 TxD+ ¹	Angeschlossen für RS422 Nicht belegt für RS485: Ausgang		7	Nicht angeschlossen
3 TxD+ ²	Signal B (RxD/TxD+): Eingang/ Ausgang		8 TXD- ²	Signal A (RxD/TxD-): Eingang/Ausgang
4 RTS ³	Ausgang Sendeanforderung/RTS (TTL-Pegel)		9 TXD- ¹	Angeschlossen für RS422 Nicht belegt für RS485: Ausgang
5 GND	Logik- oder Kommunikationsmasse		SHELL	Erdungsanschluss

¹ Pin 2 (TxD+) und Pin 9 (TxD-) sind die RS422-Sendesignale.

² Pin 3 (RxD/Tx+) und Pin 8 (RxD/TxD-) sind die RS485-Sendesignale und -Empfangssignale. Bei RS422 entspricht Pin 3 dem Signal RxD+ und Pin 8 dem Signal RxD-.

³ RTS ist ein Signal auf TTL-Ebene und kann dazu dienen, ein weiteres Halbduplex-Gerät mit Hilfe dieses Signals zu steuern. Es ist beim Senden aktiv und ansonsten inaktiv.

A.17 TeleService (TS-Adapter und TS-Adaptermodul)

Die folgenden Handbücher enthalten die technischen Daten des TS-Adapters IE Basic und des TS-Adaptermoduls:

- Industrie-Software Engineering Tools
Modularer TS-Adapter
- Industrie-Software Engineering Tools
TS-Adapter IE Basic

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Produktkatalog-Website für den TS-Adapter (<https://eb.automation.siemens.com/mall/de/de/Catalog/Search?searchTerm=TS%20Adapter%20IE%20basic&tab=>).

A.18 SIMATIC Memory Cards

Kapazität	Artikelnummer
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 Eingangssimulatoren

Tabelle A-261 Allgemeine technische Daten

Technische Daten	Simulator mit 8 Anschlussklemmen	Simulator mit 14 Anschlussklemmen	CPU 1217C Simulator
Artikelnummer	6ES7274-1XF30-0XA0	6ES7274-1XH30-0XA0	6ES7274-1XK30-0XA0
Abmessungen (B x H x T) (mm)	43 x 35 x 23	67 x 35 x 23	93 x 40 x 23
Gewicht	20 Gramm	30 Gramm	43 Gramm
Ein- und Ausgänge	8	14	14
Verwendung mit	CPU 1211C, CPU 1212C	CPU 1214C, CPU 1215C	CPU 1217C

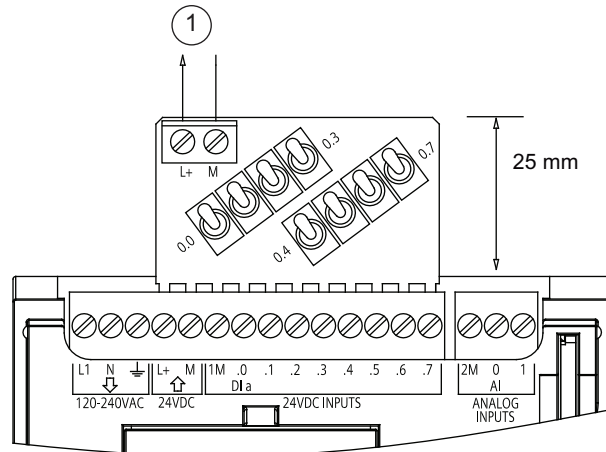
 **WARNUNG**

Sichere Verwendung von Eingangssimulatoren

Diese Eingangssimulatoren sind nicht geeignet für die Nutzung in explosionsgefährdeten Betriebsstätten der Gefahrengruppen Klasse I DIV 2 und Klasse I Zone 2. Die Schalter können bei Einsatz an Standorten der Klasse I DIV2 bzw. Klasse I Zone 2 zu Funkenbildung/Explosion führen. Nicht zugelassene Verwendung kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

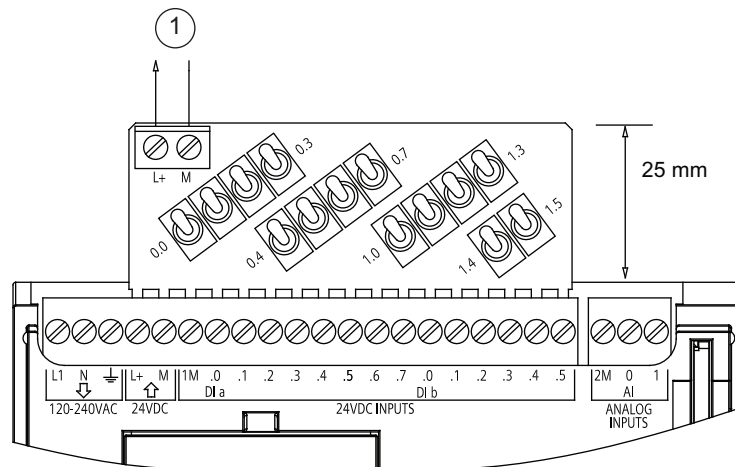
Verwenden Sie diese Eingangssimulatoren nur in Nicht-Ex-Bereichen. Verwenden Sie sie nicht in explosionsgefährdeten Betriebsstätten der Gefahrengruppen Klasse I DIV 2 und Klasse I Zone 2.

Simulator mit 8 Anschlussklemmen (6ES7274-1XF30-0XA0)



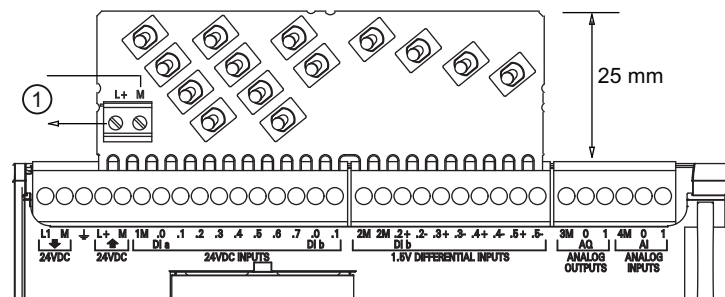
① Geberversorgung
24 V DC

Simulator mit 14 Anschlussklemmen (6ES7274-1XF30-0XA0)



① Geberversorgung
24 V DC

CPU 1217C Simulator (6ES7274-1XK30-0XA0)



① Geberversorgung
24 V DC

A.20 S7-1200 Potentiometermodul

Das S7-1200 Potentiometermodul gehört zum Zubehör für die S7-1200 CPU. Jedes Potentiometer erzeugt eine Ausgangsspannung proportional zur Position des Potentiometers, um jeden der beiden Analogeingänge der CPU mit 0 V DC bis 10 V DC zu speisen. So installieren Sie das Potentiometer:

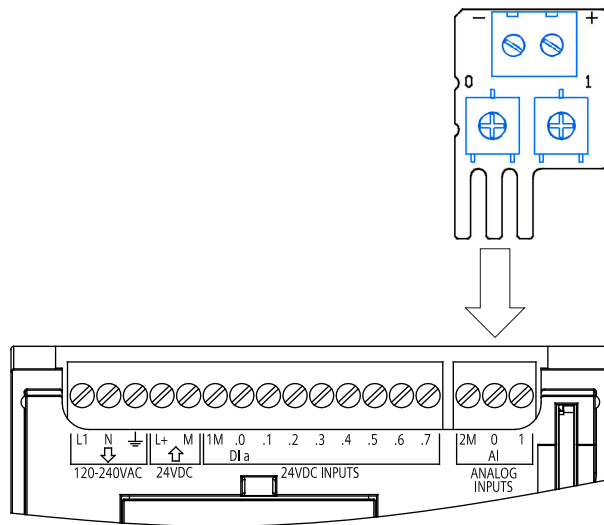
1. Stecken Sie die "Finger" der Leiterplatte in einen beliebigen Analogeingangsklemmenblock einer S7-1200 CPU und schließen Sie eine externe DC-Spannungsversorgung an den 2-fachen Anschluss am Potentiometermodul an.
2. Mit einem kleinen Schraubendreher stellen Sie das Potentiometer ein: drehen Sie rechts herum, wenn Sie den Spannungsausgang erhöhen möchten, drehen Sie links herum, um den Spannungsausgang zu verringern.

Hinweis

Befolgen Sie beim Arbeiten mit dem S7-1200 Potentiometermodul die EGB-Schutzmaßnahmen.

Technische Daten	S7-1200 Potentiometermodul
Artikelnummer	6ES7274-1XA30-0XA0
Verwendung mit	Allen S71200 CPUs
Anzahl der Potentiometer	2
Abmessungen (B x H x T) (mm)	20 x 33 x 14
Gewicht	26 Gramm
Vom Anwender bereitgestellter Spannungseingang am 2-fachen Anschluss ¹ (Klasse 2, leistungsbegrenzt oder Geberspannung aus PLC)	16,4 V DC bis 28,8 V DC
Leitungslänge (Meter)/Typ	< 30 m, geschirmtes, verdrehtes Leiterpaar
Eingangstromaufnahme bei	max. 10 mA
Potentiometer-Spannungsausgabe an S7-1200 CPU-Analogeingänge ¹	Min. 0 V DC bis 10,5 V DC
Potentialtrennung	Nicht elektrisch getrennt
Umgebungstemperaturbereich	-20 °C bis 60 °C

¹ Die Stabilität der Ausgangsspannung des Potentiometermoduls hängt von der Qualität des vom Anwender bereitgestellten Spannungseingangs am 2-fachen Anschluss ab - kann als Analogeingangsspannung betrachtet werden.

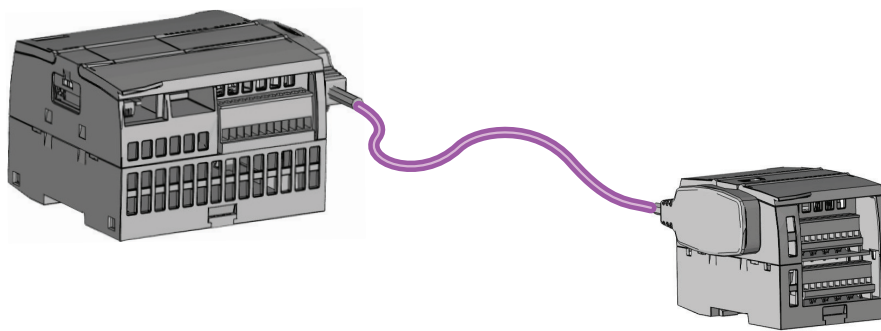


A.21 Steckleitung für Erweiterungsmodule

Tabelle A-262 Erweiterungskabel

Technische Daten	
Artikelnummer	6ES7290-6AA30-0XA0
Leitungslänge	2 m
Gewicht	200 g

Weitere Informationen zum Einbauen und Ausbauen des S7-1200 Erweiterungskabels finden Sie im Abschnitt "Einbau" (Seite 58).



A.22 Zugehörige Produkte

A.22.1 PM 1207 Stromversorgungsmodul

Das PM 1207 ist ein Stromversorgungsmodul für die SIMATIC S7-1200. Das Modul bietet die folgenden Leistungsmerkmale:

- Eingang 120/230 V AC, Ausgang 24 V DC/2,5 A

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Website mit dem Produktkatalog zum PM 1207 (<https://mall.industry.siemens.com/mall/de/de/Catalog/Product/6EP1332-1SH71>).

A.22.2 CSM 1277 Compact Switch Module

Das CSM1277 ist ein Compact Switch Module für Industrial Ethernet. Es kann zur Multiplikation der Ethernet-Schnittstelle der S7-1200 eingesetzt werden, um die gleichzeitige Kommunikation mit Bedienpanels, Programmiergeräten oder anderen Steuerungen zu ermöglichen. Das Modul bietet die folgenden Leistungsmerkmale:

- 4 x RJ45-Buchsen für den Anschluss an Industrial Ethernet
- 3-polige Klemmenleiste für den Anschluss der externen 24-V-DC-Versorgung von oben
- LEDs für Diagnose- und Statusanzeige von Industrial Ethernet-Anschlüssen
- Artikelnummer 6GK7277-1AA00-0AA0

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Website mit dem Produktkatalog zum CSM 1277 (<https://eb.automation.siemens.com/mall/de/de/Catalog/Search?searchTerm=csm%201277&tab=>).

A.22.3 CM CANopen-Modul

Das CM CANopen -Modul ist ein steckbares Modul zwischen dem SIMATIC S7-1200 PLC und einem Gerät, auf dem CANopen ausgeführt wird. Das CM CANopen -Modul kann als Master oder als Slave konfiguriert werden. Es gibt zwei CM CANopen modules: das CANopen-Modul (Artikelnummer 021620-B) und das für höhere Belastung ausgelegte CANopen (Ruggedized) -Modul (Artikelnummer 021730-B).

Das CANopen-Modul bietet die folgenden Leistungsmerkmale:

- Anschlussmöglichkeit von 3 Modulen je CPU
- Anschluss von max. 16 CANopen-Slaveknoten
- 256 Byte Eingang und 256 Byte Ausgang je Modul
- 3 LEDs liefern Diagnoseinformationen zu Modul, Netzwerk und E/A-Zustand
- Unterstützt die Speicherung der CANopen-Netzwerkconfiguration im PLC
- Das Modul ist in den Hardwarekatalog der Konfigurationssuite im TIA Portal integrierbar

- CANopen-Konfiguration über CANopen Configuration Studio (im Lieferumfang enthalten) oder jedes andere externe CANopen -Konfigurationswerkzeug
- Erfüllt die CANopen -Kommunikationsprofile CiA 301 Rev. 4.2 und CiA 302 Rev. 4.1
- Unterstützt transparentes CAN 2.0A für die Verarbeitung benutzerdefinierter Protokolle
- Vorgefertigte Funktionsbausteine für jede PLC-Programmierung im TIA Portal verfügbar
- Im Lieferumfang der CM CANopen -Module enthalten: Sub-D-Steckverbinder mit Schraubklemmen für Subnetze, CM CANopen Configuration Studio CD und USB-Konfigurationskabel.

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Website mit dem Produktkatalog zum CM CANopen.

A.22.4 Kommunikationsmodul RF120C

Das Kommunikationsmodul RF10C ermöglicht es, die von Siemens angebotenen RFID-Systeme und Codeleser direkt und einfach mit einer S7-1200 zu verbinden. Der Leser wird per Punkt-zu-Punkt-Verbindung mit dem RF120C verbunden. An eine S7-1200 können links von der CPU bis zu drei Kommunikationsmodule angeschlossen werden. Das Kommunikationsmodul RF120C wird über das TIA Portal konfiguriert. Die Artikelnummer für das Kommunikationsmodul RF120C lautet 6GT2002-OLA00.

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Webseite mit dem Produktkatalog zum RF120C.

A.22.5 SM 1238 Energy Meter

Das SM 1238 Energy Meter 480 V AC ist für die Bereitstellung in einem System S7-1200 vorgesehen. Das Modul erfasst über 200 verschiedene elektrische Mess- und Energiewerte. So schaffen Sie bereits im maschinennahen Bereich Transparenz über den Energiebedarf einzelner Komponenten einer Produktionsanlage. Anhand der vom SM 1238 Energy Meter gelieferten Messwerte können Sie Energieverbrauch und Leistungsaufnahme ermitteln.

Weitere Informationen zu diesem Produkt und die Produktdokumentation sowie die technischen Daten finden Sie auf der Website mit dem Produktkatalog zum SM 1238 Energy Meter (<https://support.industry.siemens.com/cs/ww/de/view/109483435>).

A.22.6 SIWAREX Wägeelektronik

SIWAREX WP231, WP241 und WP251

In der S7-1200 können die Wägeelektroniksysteme SIWAREX WP231, WP241 und WP251 verwendet werden. Diese Module nutzen alle Funktionen eines modernen Automatisierungssystems wie die integrierte Kommunikation, Bedienen und Beobachten, das Diagnosesystem und die Projektierungswerkzeuge im TIA Portal.

- Die Wägeelektronik SIWAREX WP231 (<https://support.industry.siemens.com/cs/ww/de/view/90229056>) mit Kalibrierung (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 MV/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA)
- Die Wägeelektronik SIWAREX WP241 (<https://support.industry.siemens.com/cs/cn/zh/view/90229063>) für den Betrieb als Bandwaage (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 M/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA)
- Die Wägeelektronik SIWAREX WP251 für Chargen- und Abfüllprozesse (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 MV/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA),

Siehe auch

SIWAREX WP251 (<https://support.industry.siemens.com/cs/ww/de/view/109481751>)

Berechnung der Leistungsbilanz

Die CPU besitzt eine interne Spannungsversorgung, die neben der CPU selbst die Erweiterungsmodule und andere 24-V-DC-Verbraucher versorgt.

Es gibt vier Arten von Erweiterungsmodulen:

- Signalmodule (SM) werden an der rechten Seite der CPU angeschlossen. Für jede CPU gibt es eine maximal anschließbare Anzahl Signalmodule, unabhängig von der Leistungsbilanz.
 - An die CPU 1214C, CPU 1215C und CPU 1217C können 8 Signalmodule angeschlossen werden.
 - An die CPU 1212C können 2 Signalmodule angeschlossen werden.
 - An die CPU 1211C können keine Signalmodule angeschlossen werden.
- Kommunikationsmodule (CM) werden an der linken Seite der CPU angeschlossen. An jede CPU können maximal 3 Kommunikationsmodule angeschlossen werden, unabhängig von der Leistungsbilanz.
- Signalboards (SB), Kommunikationsboards (CB) und Batterieboards (BB) werden oben auf der CPU angeschlossen. An jede CPU kann maximal 1 Signalboard, Kommunikationsboard oder Batterieboard angeschlossen werden.

Mit Hilfe der folgenden Informationen können Sie berechnen, wieviel Leistung die CPU für Ihre Konfiguration zur Verfügung stellen kann.

Jede CPU liefert Gleichspannung von 5 V und 24 V:

- Die CPU liefert 5-V-Gleichspannung für eventuell angeschlossene Erweiterungsmodule. Ist der Leistungsbedarf der Erweiterungsmodule an 5-V-Gleichspannung höher, als die interne Spannungsversorgung liefern kann, müssen Sie auf Erweiterungsmodule verzichten, bis die Leistungsbilanz der CPU wieder eingehalten wird.
- Jede CPU besitzt eine 24-V-DC-Geberversorgung, die die integrierten Eingänge und die Relaispulen der Erweiterungsmodule mit 24-V-Gleichspannung versorgt. Ist der Leistungsbedarf der CPU an 24-V-Gleichspannung höher als die interne Spannungsversorgung liefern kann, können Sie zusätzlich eine externe 24-V-DC-Versorgung zum Speisen der Erweiterungsmodule anschließen. Sie müssen die 24-V-DC-Versorgung manuell an die Eingänge und die Relaispulen anschließen.

WARNUNG

Wenn Sie parallel zur 24-V-DC-Geberversorgung eine externe 24-V-DC-Spannungsquelle anschließen, kann es sein, dass die beiden Spannungsquellen sich beim Aufbauen der geeigneten Ausgangsspannung gegenseitig beeinträchtigen.

Als Folge kann sich die Lebensdauer verkürzen bzw. eine oder beide Spannungsquellen können sofort ausfallen. Unvorhersehbarer Betrieb kann zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Die DC-Geberversorgung der CPU und eine externe Spannungsquelle müssen die Spannung an unterschiedlichen Punkten liefern. Dabei darf maximal eine Verbindung zwischen den beiden Spannungsquellen bestehen.

Einige der 24-V-Eingangsports des PLC-Systems sind miteinander verbunden, wobei ein logischer Bezugsleiter mehrere M-Klemmen verbindet. Der 24-V-DC-Versorgungseingang der CPU, der Relaispuleneingang des SM und ein nicht potentialgetrennter analoger Versorgungseingang sind Beispiele für Stromkreise, die miteinander verbunden sind, sofern sie in den Datenblättern als nicht potentialgetrennt angegeben sind. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.



WARNUNG

Wenn Sie nicht potentialgetrennte M-Klemmen an verschiedene Bezugspotentiale anschließen, verursacht dies unbeabsichtigten Stromfluss, der zu Beschädigung oder unvorhersehbarem Betrieb des Zielsystems und angeschlossener Geräte führen kann.

Beschädigung oder unvorhersehbarer Betrieb können zu lebensgefährlichen Verletzungen und/oder Sachschäden führen.

Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem PLC-System an dasselbe Bezugspotential an.

Informationen zu den Leistungsbilanzen der CPUs und dem Leistungsbedarf der Signalmodule finden Sie in den technischen Daten (Seite 1253).

Hinweis

Wird die Leistungsbilanz der CPU überschritten, können Sie evtl. nicht die maximale Anzahl Module an Ihre CPU anschließen.

Beispiel für eine Leistungsbilanz

Das folgende Beispiel zeigt eine Musterberechnung des Leistungsbedarfs für eine Konfiguration mit einer CPU 1214C AC/DC/Relais, einem SB 1223 2 x 24-V-DC-Eingang / 2 x 24-V-DC-Ausgang, einem CM 1241, drei SM 1223 8 DC-Eingänge / 8 Relaisausgänge und einem SM 1221 8 DC Eingang. Dieses Beispiel verfügt insgesamt über 48 Eingänge und 36 Ausgänge.

Hinweis

Die CPU hat den für die internen Relaispulen benötigten Leistungsbedarf bereits zugeordnet. Sie brauchen den Leistungsbedarf der internen Relaispulen nicht in die Berechnung der Leistungsbilanz aufzunehmen.

Die CPU in diesem Beispiel liefert genügend 5-V-DC-Spannung für die Signalmodule, doch die Geberversorgung liefert nicht genügend 24-V-DC-Spannung für alle Eingänge und Erweiterungsrelais. Die Ein- und Ausgänge benötigen 456 mA und die CPU liefert nur 400 mA. Dieser Aufbau benötigt zusätzlich mindestens 56 mA an 24-V-DC-Spannung, um alle 24-V-DC-Eingänge und -Ausgänge zu versorgen.

Tabelle B-1 Beispiel für eine Leistungsbilanz

Leistungsbilanz der CPU	5 V DC	24 V DC
CPU 1214C AC/DC/Relais	1600 mA	400 mA
<i>Minus</i>		

Leistungsbilanz der CPU	5 V DC	24 V DC
Systemvoraussetzungen	5 V DC	24 V DC
CPU 1214C, 14 Eingänge	-	14 * 4 mA = 56 mA
1 SB 1223 2 x 24-V-DC-Eingang / 2 x 24-V-DC-Ausgang	50 mA	2 * 4 mA = 8 mA
1 CM 1241 RS422/485, 5 V	220 mA	
3 SM 1223, 5-V-Spannung	3 * 145 mA = 435 mA	-
1 SM 1221, 5-V-Spannung	1 * 105 mA = 105 mA	-
3 SM 1223, je 8 Eingänge	-	3 * 8 * 4 mA = 96 mA
3 SM 1223, je 8 Relaisausgänge	-	3 * 8 * 11 mA = 264 mA
1 SM 1221, je 8 Eingänge	-	8 * 4 mA = 32 mA
Gesamtbedarf	810 mA	456 mA
<i>Gleich</i>		
Leistungsbilanz	5 V DC	24 V DC
GesamtLeistungsbilanz	790 mA	(56 mA)

Formular zum Berechnen der Leistungsbilanz

Mit Hilfe dieser Tabelle können Sie berechnen, wieviel Leistung die S7-1200 CPU für Ihre Konfiguration liefern kann. Die technischen Daten (Seite 1253) bieten Informationen zu der verfügbaren Leistung der einzelnen Ausführungen der CPUs und zum Leistungsbedarf Ihrer Signalmodule.

Tabelle B-2 Berechnen der Leistungsbilanz

Leistungsbilanz der CPU	5 V DC	24 V DC
<i>Minus</i>		
Systemvoraussetzungen	5 V DC	24 V DC
Gesamtbedarf		
<i>Gleich</i>		
Leistungsbilanz	5 V DC	24 V DC
GesamtLeistungsbilanz		

Bestellinformationen

C.1 CPU-Module

Tabelle C-1 S71200 CPUs

CPU-Varianten		Artikelnummer
CPU 1211C	CPU 1211C DC/DC/DC	6ES7211-1AE40-0XB0
	CPU 1211C AC/DC/Relais	6ES7211-1BE40-0XB0
	CPU 1211C DC/DC/Relais	6ES7211-1HE40-0XB0
CPU 1212C	CPU 1212C DC/DC/DC	6ES7212-1AE40-0XB0
	CPU 1212C AC/DC/Relais	6ES7212-1BE40-0XB0
	CPU 1212C DC/DC/Relais	6ES7212-1HE40-0XB0
CPU 1214C	CPU 1214C DC/DC/DC	6ES7214-1AG40-0XB0
	CPU 1214C AC/DC/Relais	6ES7214-1BG40-0XB0
	CPU 1214C DC/DC/Relais	6ES7214-1HG40-0XB0
CPU 1215C	CPU 1215C DC/DC/DC	6ES7215-1AG40-0XB0
	CPU 1215C AC/DC/Relais	6ES7215-1BG40-0XB0
	CPU 1215C DC/DC/Relais	6ES7215-1HG40-0XB0
CPU 1217C	CPU 1217C DC/DC/DC	6ES7217-1AG40-0XB0

C.2 Signalmodule (SMs), Signalboards (SBs) und Batterieboards (BBs)

Tabelle C-2 Signalmodule (SMs)

Signalmodule		Artikelnummer
Digitaleingang	SM 1221 8 x 24-V-DC-Eingang (stromziehend/stromliefernd)	6ES7221-1BF32-0XB0
	SM 1221 16 x 24-V-DC-Eingang (stromziehend/stromliefernd)	6ES7221-1BH32-0XB0
Digitalausgang	SM 1222 8 x 24-V-DC-Ausgang (Quelle)	6ES7222-1BF32-0XB0
	SM 1222 16 x 24-V-DC-Ausgang (Quelle)	6ES7222-1BH32-0XB0
	SM 1222 16 x 24-V-DC-Ausgang (stromziehend)	6ES7222-1BH32-1XB0
	SM 1222 8 x Relaisausgang	6ES7222-1HF32-0XB0
	SM 1222 8 x Relaisausgang (Umschaltung)	6ES7222-1XF32-0XB0
	SM 1222 16 x Relaisausgang	6ES7222-1HH32-0XB0

Signalmodule		Artikelnummer
Digitaleingänge/-ausgänge	SB 1223 8 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 8 x 24-V-DC-Ausgang (stromliefernd)	6ES7223-1BH32-0XB0
	SM 1223 16 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 16 x 24-V-DC-Ausgang (stromliefernd)	6ES7223-1BL32-0XB0
	SM 1223 16 x 24-V-DC-Eingang / 16 x 24-V-DC-Ausgang (stromziehend)	6ES7223-1BL32-1XB0
	SM 1223 8 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 8 x Relaisausgang	6ES7223-1PH32-0XB0
	SM 1223 16 x 24-V-DC-Eingang (stromziehend/stromliefernd) / 16 x Relaisausgang	6ES7223-1PL32-0XB0
	SM 1223 8 x 120/230-V-AC-Eingang (stromziehend/stromliefernd) / 8 x Relaisausgang	6ES7223-1QH32-0XB0
Analogeingang	SM 1231 4 x Analogeingang	6ES7231-4HD32-0XB0
	SM 1231 8 x Analogeingang	6ES7231-4HF32-0XB0
	SM 1231 4 x Analogeingang x 16 Bit (hochfunktionell)	6ES7231-5ND32-0XB0
	SM 1238 Energy Meter 480 V AC	6ES7238-5XA32-0XB0
Analogausgang	SM 1232 2 x Analogausgang	6ES7232-4HB32-0XB0
	SM 1232 4 x Analogausgang	6ES7232-4HD32-0XB0
Analogeingänge/-ausgänge	SM 1234 4 x Analogeingang / 2 x Analogausgang	6ES7234-4HE32-0XB0
RTD und Thermo-element	SM 1231 TC 4 x 16 Bit	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 Bit	6ES7231-5PF32-0XB0
Technologiemodule	SM 1278 4xIO-Link-Master	6ES7278-4BD32-0XB0
	Wägeelektronik SIWAREX WP231 mit Kalibrierung (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 MV/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA)	7MH4960-2AA01
	Wägeelektronik SIWAREX WP241 für den Betrieb als Bandwaage (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 MV/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA)	7MH4960-4AA01
	Wägeelektronik SIWAREX WP251 für Chargen- und Abfüllprozesse (1 Kanal) für DMS-Wägezellen/-Vollbrücken (1-4 MV/V) für die SIMATIC S7-1200, Schnittstelle RS-485 und Ethernet, integrierte E/A: 4 DI / 4 DO, 1 AO (0/4...20 MA),	7MH4960-6AA01

Tabelle C-3 Signalboards (SBs) und Batterieboards (BBs)

Signal- und Batterieboards		Artikelnummer
Digitaleingang	SB 1221 200 kHz, 4 x 24-V-DC-Eingang (stromliefernd)	6ES7221-3BD30-0XB0
	SB 1221 200 kHz, 4 x 5-V-DC-Eingang (stromliefernd)	6ES7221-3AD30-0XB0

Signal- und Batterieboards		Artikelnummer
Digitalausgang	SB 1222 200 kHz, 4 x 24-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7222-1BD30-0XB0
	SB 1222 200 kHz, 4 x 5-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7222-1AD30-0XB0
Digitaleingänge/-ausgänge	SB 1223 2 x 24-V-DC-Eingang (stromziehend) / 2 x 24-V-DC-Ausgang (stromliefernd)	6ES7223-0BD30-0XB0
	SB 1223 200 kHz 2 x 24-V-DC-Eingang (stromliefernd) / 2 x 24-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7223-3BD30-0XB0
	SB 1223 200 kHz 2 x 5-V-DC-Eingang (stromliefernd) / 2 x 5-V-DC-Ausgang (stromziehend/stromliefernd)	6ES7223-3AD30-0XB0
Analog	SB 1232 1 Analogausgang	6ES7232-4HA30-0XB0
	SB 1231 1 Analogeingang	6ES7231-4HA30-0XB0
	SB 1231 1 Analogeingang Thermoelement	6ES7231-5QA30-0XB0
	SB 1231 1 Analogeingang RTD	6ES7231-5PA30-0XB0
Batterie	BB 1297 Batterieboard (Batterietyp CR1025 nicht enthalten)	6ES7297-0AX30-0XA0

C.3 Kommunikation

Tabelle C-4 Kommunikationsmodul (CM)

Kommunikationsmodul (CM)			Artikelnummer
RS232, RS422 und RS485	CM 1241 RS232	RS232	6ES7241-1AH32-0XB0
	CM 1241 RS422/485	RS422/485	6ES7241-1CH32-0XB0
PROFIBUS	CM 1243-5	PROFIBUS-Master	6GK7243-5DX30-0XE0
	CM 1242-5	PROFIBUS-Slave	6GK7242-5DX30-0XE0
AS-i-Master	CM 1243-2	AS-i-Master	3RK7243-2AA30-0XB0

Tabelle C-5 Kommunikationsboard (CB)

Kommunikationsboard (CB)			Artikelnummer
RS485	CB 1241 RS485	RS485	6ES7241-1CH30-1XB0

Tabelle C-6 Kommunikationsprozessor (CP)

CP	Schnittstelle	Artikelnummer
CP 1242-7 GPRS V2	GPRS	6GK7242-7KX31-0XE0
CP 1243-7 LTE-US	LTE	6GK7243-7SX30-0XE0
CP 1243-7 LTE-EU	LTE	6GK7243-7KX30-0XE0
CP 1243-1	IE-Schnittstelle	6GK7243-1BX30-0XE0
CP 1243-8 IRC	IE- und serielle Schnittstelle	6GK7243-8RX30-0XE0

Tabelle C-7 TeleService

TS-Adapter	Artikelnummer
TS-Modul RS232	6ES7972-0MS00-0XA0
TS-Modul Modem	6ES7972-0MM00-0XA0

Tabelle C-8 Zubehör

Zubehör	Artikelnummer
Antenne	ANT794-4MR GSM/GPRS-Antenne 6NH9860-1AA00
	ANT794-3M Flachantenne 6NH9870-1AA00

Tabelle C-9 Steckverbinder

Art des Steckverbinders	Artikelnummer
RS485	Schraubklemmenanschluss, 35-Grad-Kabelabgang 6ES7972-0BA42-0XA0
	FastConnect-Anschluss, 35-Grad-Kabelabgang 6ES7972-0BA60-0XA0

C.4 Fehlersichere CPUs und Signalmodule

Tabelle C-10 Fehlersichere CPUs

Fehlersichere CPU-Modelle	Artikelnummer
CPU 1212FC	CPU 1212FC DC/DC/DC 6ES7212-1AF40-0XB0
	CPU 1212FC DC/DC/Relais 6ES7212-1HF40-0XB0
CPU 1214FC	CPU 1214FC DC/DC/DC 6ES7214-1AF40-0XB0
	CPU 1214FC DC/DC/Relais 6ES7214-1HF40-0XB0
CPU 1215FC	CPU 1215FC DC/DC/DC 6ES7215-1AF40-0XB0
	CPU 1215FC DC/DC/Relais 6ES7215-1HF40-0XB0

Tabelle C-11 Fehlersichere Signalmodule

Signalmodule mit funktionaler Sicherheit	Artikelnummer
Digitaleingang	SM 1226 F-DI 16 x 24 V DC 6ES7226-6BA32-0XB0
Digitalausgang	SM 1226 F-DQ 4 x 24 V DC 6ES7226-6DA32-0XB0
	SM 1226 F-DQ 2 x Relay 6ES7226-6RA32-0XB0

C.5 Sonstige Module

Tabelle C-12 Zugehörige Produkte

Beschreibung		Artikelnummer
Stromversorgung	PM 1207 Stromversorgung	6EP1332-1SH71
Ethernet-Switch	CSM 1277 Ethernet-Switch - 4 Ports	6GK7277-1AA10-0AA0
CM CANopen	CANopen für SIMATIC S7-1200	021620-B
	CANopen (Ruggedized) für SIMATIC S7-1200	021730-B
RF120C	Kommunikationsmodul RF120C	6GT2002-0LA00

C.6 Memory Cards

Tabelle C-13 Memory Cards

SIMATIC Memory Cards	Artikelnummer
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 Grundlegende HMI-Geräte

Tabelle C-14 HMI-Geräte

Grundlegende HMI-Panels	Artikelnummer
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 Ersatzteile und sonstige Hardware

Tabelle C-15 Erweiterungskabel, Simulatoren und Endhalter

Beschreibung		Artikelnummer
Steckleitung für Erweiterungsmodule	Steckleitung für Erweiterungsmodule, 2 m	6ES7290-6AA30-0XA0
E/A-Simulator	Simulator (1211C/1212C - 8 E/A)	6ES7274-1XF30-0XA0
	Simulator (1214C/1215C - 14 E/A)	6ES7274-1XH30-0XA0
	Simulator CPU 1217C	6ES7274-1XK30-0XA0
Potentiometermodul	S71200 Potentiometermodul	6ES7274-1XA30-0XA0
Ethernet-Zugentlastung	Zugentlastung ein RJ45-Anschluss, 10/100 Mbit/s	6ES7290-3AA30-0XA0
	Zugentlastung zwei RJ45-Anschlüsse, 10/100 Mbit/s	6ES7290-3AB30-0XA0
Ersatzabdeckklappe	CPU 1211C/1212C	6ES7291-1AA30-0XA0
	CPU 1214C	6ES7291-1AB30-0XA0
	CPU 1215C	6ES7291-1AC30-0XA0
	CPU 1217C	6ES7291-1AD30-0XA0
	Signalmodul (SM), 45 mm	6ES7291-1BA30-0XA0
	Signalmodul (SM), 70 mm	6ES7291-1BB30-0XA0
	Kommunikationsmodul (zur Verwendung mit den Modulen 6ES72xx-xxx32-0XB0 und 6ES72xx-xxx30-0XB0)	6ES7291-1CC30-0XA0
Endhalter	Endhalter Thermoplastik, 10 mm	8WA1808
	Endhalter Stahl, 10,3 mm	8WA1805

Austauschen des Klemmenblocks

Es ist wichtig, dass Sie für Ihr Modul den richtigen Klemmenblock verwenden. Ziehen Sie die nachstehenden Tabellen und die technischen Daten Ihres Moduls hinzu, um den richtigen Klemmenblock für den Austausch zu ermitteln.

Hinweis

Abnehmbare, codierte Klemmenblöcke

PLCs müssen stets richtig verdrahtet sein, um Sicherheit und ordnungsgemäßen Betrieb zu gewährleisten.

Beim Austauschen eines Klemmenblocks in Ihrer CPU oder Ihrem SM ist es wichtig, dass Sie den richtigen Klemmenblock und die richtige Verdrahtungsquelle für Ihr Modul verwenden.

Durch die Codierung ist sichergestellt, dass Sie nicht unbeabsichtigt einen mit Hochspannung verdrahteten Klemmenblock in ein Niederspannungsmodul einsetzen oder ein mit einer Sonderspannung verdrahteten Klemmenblock in ein Modul mit normaler Spannung. Einige Klemmenblöcke sind links, rechts oder in der Mitte codiert.

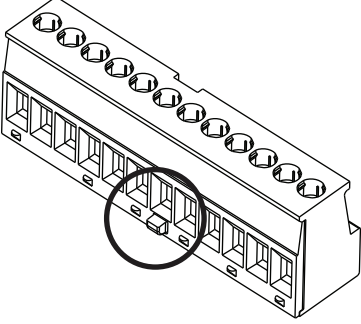
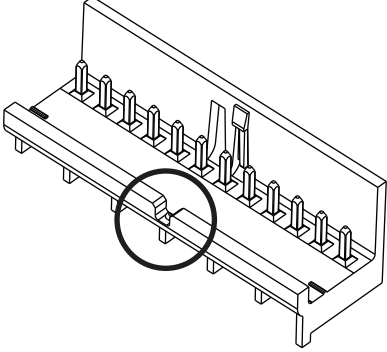
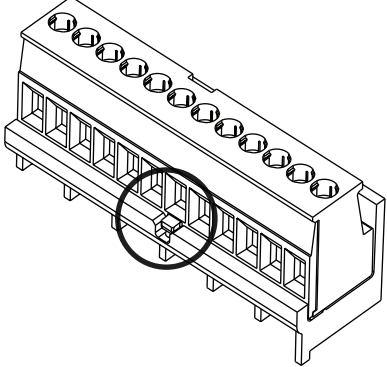
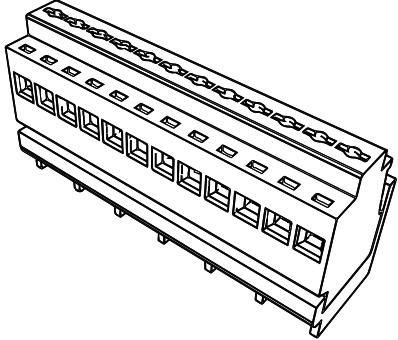
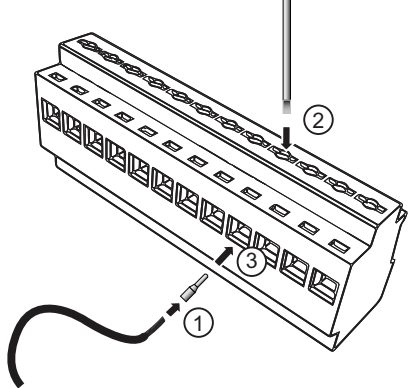
		
Abnehmbarer Klemmenblock (codiertes Beispiel dargestellt)	Codierung am Gerät	Codierung am Gerät passt nur zum entsprechenden abnehmbaren Klemmenblock
		
Push-in-Klemmenblock	Gehen Sie wie folgt vor: 1. Führen Sie die Ader und den Crimpanschluss ein. 2. Drücken Sie mit einem kleinen Schraubendreher die Zunge nieder, um den Schlitz zu öffnen. 3. Führen Sie den Crimpanschluss in den geöffneten Schlitz am Klemmenblock ein.	

Tabelle C-16 S7-1200 CPU ab V4 - Ersatzklemmenblöcke

Wenn Sie eine S7-1200 CPU ab V4.0 haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock (Schraubklemmen) Artikelnummer	Äquivalent Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
CPU 1211C DC/DC/DC (6ES7211-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8-polig, verzinkt
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-polig, verzinkt

Wenn Sie eine S7-1200 CPU ab V4.0 haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock (Schraubklemmen) Artikelnummer	Äquivalent Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
CPU 1211C DC/DC/Relais (6ES7211-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-polig, verzinnt, codiert
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-polig, verzinnt
CPU 1211C AC/DC/Relais (6ES7211-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-polig, verzinnt, codiert
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14-polig, verzinnt, codiert
CPU 1212C DC/DC/DC (6ES7212-1AE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	8-polig, verzinnt
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-polig, verzinnt
CPU 1212C DC/DC/Relais (6ES7212-1HE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-polig, verzinnt, codiert
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	14-polig, verzinnt
CPU 1212C AC/DC/Relais (6ES7212-1BE40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	8-polig, verzinnt, codiert
	6ES7292-1AP40-0XA0	6ES7292-2AP40-0XA0	14-polig, verzinnt, codiert
CPU 1214C DC/DC/DC (6ES7214-1AG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-polig, verzinnt
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt
CPU 1214C DC/DC/Relais (6ES7214-1HG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-polig, verzinnt, codiert
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt
CPU 1214C AC/DC/Relais (6ES7214-1BG40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-polig, verzinnt, codiert
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20-polig, verzinnt, codiert
CPU 1215C DC/DC/DC (6ES7215-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-polig, verzinnt
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt
CPU 1215C DC/DC/Relais (6ES7215-1HG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-polig, verzinnt, codiert
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt

Wenn Sie eine S7-1200 CPU ab V4.0 haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock (Schraubklemmen) Artikelnummer	Äquivalent Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
CPU 1215C AC/DC/Relais (6ES7215-1BG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-polig, verzinkt, codiert
	6ES7292-1AV40-0XA0	6ES7292-2AV40-0XA0	20-polig, verzinkt, codiert
CPU 1217C DC/DC/DC (6ES7217-1AG40-0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AK30-0XA0	6ES7292-2AK30-0XA0	10-polig, verzinkt
	6ES7292-1AR30-0XA0	6ES7292-2AR30-0XA0	16-polig, verzinkt
	6ES7292-1AT30-0XA0	6ES7292-2AT30-0XA0	18-polig, verzinkt

Tabelle C-17 S7-1200 SMs ab V3.2 - Ersatzklemmenblöcke

Wenn Sie S7-1200 SMs ab V3.2 haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock (schraubbar) Artikelnummer	Äquivalent Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
SM 1221 DI 8 x DC (6ES7221-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1222 DQ 8 x DC (6ES7222-1BF32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1222 DQ 8 x Relais (6ES7222-1HF32-0XB0)	6ES7292-1AG40-0XA1	6ES7292-2AG40-0XA1	7-polig, verzinkt, links codiert
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) für Spannungseingang (oben)	6ES7292-1AG40-0XA2	6ES7292-2AG40-0XA2	7-polig, verzinkt, in der Mitte codiert
SM 1238 Energy Meter 480 V AC (6ES7238-5XA32-0XB0) für Stromeingang (unten)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1231 AI 4 x 13 Bit (6ES7231-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1232 AQ 2 x 14 Bit (6ES7232-4HB32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1231 AI 4 x TC (6ES7231-5QD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1231 AI 4 x 16 Bit (6ES7231-5ND32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1221 DI 16 x DC (6ES7221-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1222 DQ 16 x DC (6ES7222-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1222 DQ 16 x Relais (6ES7222-1HH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-polig, verzinkt, rechts codiert

Wenn Sie S7-1200 SMs ab V3.2 haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock (schraubbar) Artikelnummer	Äquivalent Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
SM 1223 DI 8 x DC/DQ 8 x DC (6ES7223-1BH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1223 8 x DC/8 x Relais (6ES7223-1PH32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-polig, verzinkt, rechts codiert
SM 1223 8 x AC/8 x Relais (6ES7223-1QH32-0XB0)	6ES7292-1AG40-0XA0	6ES7292-2AG40-0XA0	7-polig, verzinkt, rechts codiert
SM 1234 AI 4 / AQ 2 (6ES7234-4HE32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1231 AI 8 x 13 Bit (6ES7231-4HF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1232 AQ 4 x 14 Bit (6ES7232-4HD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1231 AI 4 x RTD (6ES7231-5PD32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1231 AI 8 x TC (6ES7231-5QF32-0XB0)	6ES7292-1BG30-0XA0	6ES7292-2BG30-0XA0	7-polig, vergoldet
SM 1278 IO LINK (6ES7278-4BD32-0XB0)	6ES7292-1AG30-0XA0	6ES7292-2AG30-0XA0	7-polig, verzinkt
SM 1222 DQ 8 x Relais (Umschaltung) (6ES7222-1XF32-0XB0)	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11-polig, verzinkt, codiert
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-polig, verzinkt
SM 1223 DI 16 x DC/DQ 16 x Relais (6ES7223-1PL32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-polig, verzinkt
	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11-polig, verzinkt, codiert
SM 1231 AI 8 x RTD (6ES7231-5PF32-0XB0)	6ES7292-1BL30-0XA0	6ES7292-2BL30-0XA0	11-polig, vergoldet

Tabelle C-18 S7-1200 SBs, CBs und BBs - Ersatzklemmenblöcke

Wenn Sie ein S7-1200 SB, CB oder BB haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)	
	Klemmenblock (schraubbar) Artikelnummer	Klemmenblock Beschreibung
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)	6ES7292-1BF30-0XA0	6-polig
SB 1221 DI 4 x 5 V DC (6ES7221-3AD30-0XB0)		
SB 1221 DI 4 x 24 V DC (6ES7221-3BD30-0XB0)		
SB 1222 DO 4 x 5 V DC (6ES7222-1AD30-0XB0)		
SB 1222 DO 4 x 24 V DC (6ES7222-1BD30-0XB0)		
SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC (6ES7223-0BD30-0XB0)		
SB 1223 DI 2 x 5 V DC / DQ 2 x 5 V DC (6ES7223-3AD30-0XB0)		
SB 1223 DI 2 x 24 V DC / DQ 2 x 24 V DC (6ES7223-3BD30-0XB0)		
SB 1231 AI 1 x 12 Bit (6ES7231-4HA30-0XB0)		
SB 1231 AI 1 x RTD (6ES7231-5PA30-0XB0)		
SB 1231 AI 1 x TC (6ES7231-5QA30-0XB0)		
SB 1232 AO 1 x 12 Bit (6ES7232-4HA30-0XB0)		
CB 1231 RS-485 (6ES7241-1CH30-1XB0)		
BB 1297 Batterie (6ES7297-0AX30-0XA0)		

Tabelle C-19 Fehlersichere CPUs - Ersatzklemmenblöcke

Wenn Sie eine fehlersichere CPU haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock Artikelnummer	Gleichwertiger Push-in-Klemmenblock Artikelnummer	Klemmenblock Beschreibung
CPU 1212FC DC/DC/DC (6ES7212-1AF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH30-0XA0	6ES7292-2AH30-0XA0	12-polig, verzinnt
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	20-polig, verzinnt
CPU 1212FC DC/DC/Relais (6ES7212-1HF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AH40-0XA0	6ES7292-2AH40-0XA0	3-polig, vergoldet
	6ES7292-1AP30-0XA0	6ES7292-2AP30-0XA0	12-polig, verzinnt, codiert
CPU 1214FC DC/DC/DC (6ES7214-1AF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-polig, verzinnt
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt
CPU 1214FC DC/DC/Relais (6ES7214-1HF40-0XB0)	6ES7292-1BC30-0XA0	6ES7292-2BC30-0XA0	3-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	12-polig, verzinnt, codiert
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt

Wenn Sie eine fehlersichere CPU haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock Artikelnummer	Gleichwertiger Push-in- Klemmenblock Artikel- nummer	Klemmenblock Beschreibung
CPU 1215FC DC/DC/DC (6ES7215-1AF40 0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AM30-0XA0	6ES7292-2AM30-0XA0	12-polig, verzinnt
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt
CPU 1215FC DC/DC/Relais (6ES7215-1HF40 0XB0)	6ES7292-1BF30-0XB0	6ES7292-2BF30-0XB0	6-polig, vergoldet
	6ES7292-1AM40-0XA0	6ES7292-2AM40-0XA0	2-polig, verzinnt, co- diert
	6ES7292-1AV30-0XA0	6ES7292-2AV30-0XA0	20-polig, verzinnt

Tabelle C-20 Fehlersichere Signalmodule - Ersatzklemmenblöcke

Wenn Sie ein fehlersicheres Signalmo- dul haben (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)		
	Klemmenblock Artikelnummer	Gleichwertiger Push-in- Klemmenblock Artikel- nummer	Klemmenblock Beschreibung
SM 1226 F-DI (6ES7226-6BA32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-polig, verzinnt
SM 1226 F-DQ (6ES7226-6DA32-0XB0)	6ES7292-1AL30-0XA0	6ES7292-2AL30-0XA0	11-polig, verzinnt
SM 1226 F-Relais (6ES7226-6RA32-0XB0)	6ES7292-1AL40-0XA0	6ES7292-2AL40-0XA0	11-polig, verzinnt, co- diert

C.9 Programmiersoftware

Tabelle C-21 Programmiersoftware

SIMATIC-Software		Artikelnummer
Programmiersoft- ware	STEP 7 Basic V17	6ES7822-0AA06-0YA5
	STEP 7 Professional V17	6ES7822-1AA06-0YA5
Visualisierungs- software	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 max. PowerTags V17	6AV2103-0XA06-0AA5

C.10 OPC UA-Lizenzen

Tabelle C-22 OPC UA-Lizenzen für S7-1200

OPC UA-Lizenzen		Artikelnummer
SIMATIC-Lizenzen	SIMATIC OPC UA S7-1200 Basic DVD	6ES7823-0BA00-2BA0
	SIMATIC OPC UA S7-1200 Basic DL	6ES7823-0BE00-2BA0

Geräteaustausch und Ersatzteilkompatibilität

D.1 Eine CPU mit Schutz vertraulicher PLC-Konfigurationsdaten ersetzen

Die Zuweisung von Passwörtern zum Schutz vertraulicher PLC-Konfigurationsdaten wirkt sich auch auf den Austausch von Teilen aus.

Regeln für den Austausch von Teilen

Beachten Sie die folgenden Regeln für den Austausch von Teilen:

Konfigurierung der Ersatz-CPU im TIA Portal

- Wenn möglich, verwenden Sie eine CPU, die keine Projektkonfiguration und kein konfiguriertes Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten wie die Ersatz-CPU hat.
Vorteil: Sie können das Projekt ohne weitere Vorbereitung in die Ersatz-CPU laden.
- Wenn die Ersatz-CPU bereits konfiguriert ist, müssen Sie die CPU auf die Werkseinstellungen zurücksetzen (Seite 1214) und diese Optionen wählen:
 - Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten löschen
 - Memory Card formatieren, wenn die CPU eine Memory Card hat

Eine CPU mit Konfigurationsdaten auf einer SIMATIC Memory Card ersetzen

- Wenn Sie einer CPU in Ihrem Projekt **kein** Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten zugewiesen haben, können Sie die Memory Card der zu ersetzenden CPU ohne weitere Aktion in eine neue, noch nicht verwendete CPU stecken.
- Wenn die Ersatz-CPU bereits mit einem Passwort konfiguriert ist, müssen Sie zunächst diese CPU auf die Werkseinstellungen zurücksetzen (Seite 1214) und dabei die Option "Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten löschen" verwenden.
- Wenn Sie das gleiche Passwort zum Schutz vertraulicher PLC-Konfigurationsdaten einer Gruppe von CPUs zugeordnet haben, können Sie der Ersatz-CPU das Gruppenpasswort auch in der Gerätekonfiguration (Seite 161) im TIA Portal zuweisen. In diesem Fall können Sie beispielsweise eine Memory Card mit dem aktuellen Projekt in die CPU stecken und sie ohne weitere Passwortbearbeitung in Betrieb nehmen.
- Wenn Sie jeder CPU in Ihrem Projekt ein anderes Passwort zuweisen, gehen Sie online (Seite 1209) und legen das Passwort für den Schutz vertraulicher Daten für die Ersatz-CPU mit den Online- und Diagnosetools fest. Wählen Sie in den Online-Funktionen die Option "Passwort für den Schutz vertraulicher PLC-Konfigurationsdaten festlegen". (Seite 1213)

D.2 Ersatz einer CPU V3.0 durch eine CPU V4.x

Zum Hochrüsten einer CPU V3.0 auf eine CPU V4.x müssen Sie die CPU-Hardware ersetzen. Eine CPU V3.0 kann nicht mit einem Firmware-Update auf eine CPU V4.x hochgerüstet werden.

Sie können dann in Ihrem STEP 7-Projekt Ihre CPU V3.0 durch eine CPU V4.x ersetzen (Seite 152) und Ihr bestehendes, für die CPU V3.0 entwickeltes STEP 7-Projekt verwenden.

Wenn Sie eine CPU V3.0 durch eine CPU V4.x ersetzen, können Sie gleichzeitig nach Firmware-Updates (Seite 131) für Ihre Signal- und Kommunikationsmodule suchen und diese installieren.

Hinweis

Kein Gerätetausch möglich in STEP 7 von V4.x nach V3.0

Sie können eine CPU V3.0 durch eine CPU V4.x ersetzen, doch Sie können nach dem Laden der Konfiguration eine CPU V4.x nicht durch eine CPU V3.0 ersetzen. Wenn Sie Ihr bestehendes STEP 7 V3.0-Projekt anzeigen oder anderweitig nutzen möchten, legen Sie vor dem Gerätetausch ein Archiv Ihres STEP 7 V3.0-Projekts an.

Sofern Sie die ausgetauschte Gerätekonfiguration noch nicht geladen haben, können Sie sie rückgängig machen. Nach dem Laden können Sie den Ersatz von V3.0 durch V4.x nicht mehr rückgängig machen.

Beachten Sie dabei einige Konfigurations- und Betriebsänderungen zwischen den beiden CPU-Versionen:

STEP 7-Projekt hochrüsten

Sie können STEP 7 V11- oder V12-Projekte nicht direkt in STEP 7 V15 hochrüsten. Sie müssen diese Projekte zunächst auf STEP 7 V13 SP1 oder STEP 7 V13 SP2 hochrüsten. Dann können Sie dieses Projekt als Basis für ein Upgrade auf STEP 7 V15 verwenden.



WARNUNG

Risiken beim Kopieren und Einfügen von Programmlogik aus älteren Versionen von STEP 7

Kopieren von Programmlogik aus einer älteren Version von STEP 7 wie STEP 7 V12 in STEP 7 V15 kann unvorhersehbares Verhalten bei der Programmausführung oder Fehler beim Übersetzen verursachen. In den verschiedenen Versionen von STEP 7 werden Programmelemente unterschiedlich implementiert. Beim Übersetzen werden die Unterschiede nicht immer erkannt, wenn Sie die Änderungen durch Einfügen aus einer Vorgängerversion in STEP 7 V15 vorgenommen haben. Die Ausführung unberechenbarer Programmlogik kann schwere oder lebensgefährliche Verletzungen verursachen, wenn Sie das Programm nicht korrigieren.

Aktualisieren Sie bei Verwendung von Programmlogik aus einer Version von STEP 7 vor STEP 7 V15 immer das gesamte Projekt auf STEP 7 V15. Sie können Programmlogik nach Bedarf kopieren, ausschneiden, einfügen und bearbeiten. In STEP 7 V15 können Sie ein Projekt aus STEP 7 V13 SP1 oder höher öffnen. STEP 7 führt anschließend die notwendigen Kompatibilitätsumwandlungen durch und aktualisiert das Programm korrekt. Solche Umwandlungen und Korrekturen im Zusammenhang mit der Aktualisierung sind für die ordnungsgemäße Übersetzung und Ausführung des Programms notwendig. Wenn Ihr Projekt älter als STEP 7 V13 SP1 ist, müssen Sie das Projekt schrittweise auf STEP 7 V15 aktualisieren.

Organisationsbausteine

Sie können konfigurieren, ob die OB-Ausführung unterbrechbar oder nicht unterbrechbar (Seite 86) sein soll. In Projekten von CPUs der Version V3.0 legt STEP 7 für alle OBs standardmäßig die nicht unterbrechbare Ausführung fest.

STEP 7 legt für alle OB-Prioritäten (Seite 86) die Werte aus dem STEP 7-Projekt der V3.0-CPU fest.

Nachfolgend können Sie bei Bedarf die Unterbrechbarkeits- und Prioritätseinstellungen ändern.

Die Anlaufinformationen des Diagnosefehler-OB (Seite 80) beziehen sich auf das Submodul als Ganzes, wenn kein Diagnoseereignis ansteht.

CPU-Passwortschutz

STEP 7 legt für die CPU V4.x als Passwortschutzstufe (Seite 163) die gleiche Stufe fest, die in der CPU V3.0 eingestellt war, und weist das Passwort der Version 3.0 dem Passwort für "Vollzugriff (kein Schutz)" der CPU V4.x zu:

Schutzstufe in der Version 3.0	Zugriffsstufe in der Version V4.x
Kein Schutz	Vollzugriff (kein Schutz)
Schreibschutz	Lesezugriff
Schreib-/Lesechutz	HMI-Zugriff

Beachten Sie, dass die V4.x-Zugriffsstufe "Kein Zugriff (kompletter Schutz)" in der Version 3.0 nicht vorhanden war.

Webserver

Wenn Sie in Ihrem V3.0-Projekt benutzerdefinierte Webseiten verwenden, speichern Sie diese im Installationsverzeichnis Ihres Projekts in Unterverzeichnis "UserFiles\Webserver", bevor Sie das Upgrade Ihres Projekts starten. Sind die benutzerdefinierten Seiten in diesem Verzeichnis abgelegt, werden beim Speichern des STEP 7-Projekts auch die benutzerdefinierten Webseiten gespeichert.

Wenn Sie eine CPU V3.0 durch eine CPU V4.x ersetzen, sind Ihre Webserver-Projekteinstellungen (Seite 850) zum Aktivieren des Webserver und der HTTPS-Einstellungen die gleichen Einstellungen wie in der Version 3.0. Sie können dann Benutzer, Rechte, Passwörter (Seite 852) und Sprachen (Seite 850) wie für die Arbeit mit dem Webserver benötigt ändern. Wenn Sie keine Benutzer mit zusätzlichen Rechten konfigurieren, sind Sie auf das begrenzt, was auf den Standard-Webseiten (Seite 861) angezeigt wird. Die S7-1200 CPU V4.x unterstützt den bisher vorkonfigurierten Benutzer "admin" und das zugehörige Passwort nicht.

Die Webserverseite für das Datenprotokoll der S7-1200 V3.0 bot bisher die Funktion "Herunterladen und Löschen". Die Dateibrowser-Seite (Seite 893) des Webserver der Version 4.x, über die Sie auf Datenprotokolle zugreifen, bietet diese Funktion nicht mehr an. Stattdessen bietet der Webserver die Möglichkeit zum Herunterladen, Umbenennen und Löschen von Datenprotokolldateien.

Inkompatibilität der Übertragungskarte

Sie können eine Übertragungskarte (Seite 118) der Version 3.0 nicht dazu verwenden, ein Programm der Version 3.0 in eine CPU der Version 4.x zu übertragen. Sie müssen das V3.0-Projekt in STEP 7 öffnen, das Gerät in eine CPU V4.x ändern (Seite 152) und das STEP 7-Projekt dann in Ihre CPU V4.x laden. Nachdem Sie Ihr Projekt in ein Projekt der Version 4.x geändert haben, können Sie eine V4.x-Übertragungskarte für nachfolgende Programmübertragungen einrichten.

GET/PUT-Kommunikation

Standardmäßig ist bei S7-1200 V3.0-CPU's die Kommunikation über GET/PUT aktiviert. Wenn Sie Ihre CPU V3.0 durch eine CPU V4.x ersetzen (Seite 152), wird im Informationsbereich zur Kompatibilität eine Meldung angezeigt, dass GET/PUT aktiviert ist.

Unterstützung der Bewegungssteuerung

S7-1200 CPU's V4.x unterstützen die Bewegungssteuerungsbibliotheken V1.0 und V2.0 nicht. Wenn Sie für ein STEP 7-Projekt mit Bewegungssteuerungsbibliotheken V1.0 oder V2.0 einen Gerätetausch durchführen, werden durch den Gerätetausch die Bewegungssteuerungsbibliotheken V1.0 und V2.0 zum Zeitpunkt des Übersetzens durch kompatible Bewegungssteuerungsanweisungen der Version 3.0 ersetzt.

Wenn Sie für ein STEP 7-Projekt mit zwei verschiedenen Versionen von Bewegungssteuerungsanweisungen (V3.0 und V5.0) einen Gerätetausch von einer CPU V3.0 auf eine CPU V4.x durchführen, werden durch den Gerätetausch zum Zeitpunkt des Übersetzens kompatible Bewegungssteuerungsanweisungen der Version 5.0 ersetzt.

Bei einem Gerätetausch von einer CPU V3.0 auf eine CPU V4.x wird die Version des Technologieobjekts (TO) für die Bewegungssteuerung nicht automatisch von V3.0 in V5.0 geändert. Wenn Sie auf die späteren Versionen aufrüsten möchten, müssen Sie im Anweisungsverzeichnis die erforderliche Version der S7-1200 Bewegungssteuerung für Ihr Projekt auswählen. Dies wird in der nachstehenden Tabelle gezeigt:

CPU-Version	Zulässige Bewegungssteuerungsversionen
V4.3 (Bewegungssteuerung V5.0)	V6.0 oder V5.0 oder V4.0 oder V3.0
V4.2.x (Bewegungssteuerung V5.0)	V6.0 oder V5.0 oder V4.0 oder V3.0
V4.1 (Bewegungssteuerung V5.0)	V5.0 oder V4.0 oder V3.0
V4.0 (Bewegungssteuerung V4.0)	V4.0 oder V3.0
V3.0 (Bewegungssteuerung V3.0)	V3.0

Die TO-Struktur ist bei den Bewegungssteuerungsversionen V3.0 und V5.0 unterschiedlich. Alle zugehörigen Bausteine ändern sich ebenfalls. Bausteinschnittstellen, Beobachtungstabellen und die Trace-Funktion werden auf die neue Struktur der Bewegungssteuerung V5.0 aktualisiert. Die Unterschiede in den Achsenparametern der Bewegungssteuerung zwischen der CPU V3.0 und der CPU V4.x werden in den folgenden zwei Tabellen aufgeführt:

CPU V3.0 (Bewegungssteuerung V3.0)	CPU V4.x (Bewegungssteuerung V5.0)
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution

CPU V3.0 (Bewegungssteuerung V3.0)	CPU V4.x (Bewegungssteuerung V5.0)
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity
MotionStatus.Distance	StatusPositioning.Distance
MotionStatus.TargetPosition	StatusPositioning.TargetPosition
StatusBits.SpeedCommand	StatusBits.VelocityCommand
StatusBits.Homing	StatusBits.HomingCommand

Der einzige "commandtable"-Parameter, der umbenannt wird, ist das Array mit den Befehlen:

V3.0	V4.x
Config.Command[]	Command[]

Hinweis: Das Array "Command[]" ist in Version 3.0 ein UDT des Typs "TO_CmdTab_Config_Command" und in Version 4.x ein UDT des Typs "TO_Struct_Command".

Geänderte Anweisungen

Bei den folgenden Anweisungen ändern sich Parameter oder das Verhalten:

- RDREC und WRREC (Seite 373)
- CONV (Seite 285)

HMI-Panel-Kommunikation

Wenn eines oder mehrere Ihrer HMI-Panels (Seite 32) an Ihre S7-1200 CPU der Version 3.0 angeschlossen waren, hängt die Kommunikation mit der S7-1200 CPU der Version 4.x vom verwendeten Kommunikationstyp und von der Firmware-Version des HMI-Panels ab. Übersetzen Sie Ihr Projekt erneut und laden Sie es in die CPU und HMI und/oder aktualisieren Sie Ihre HMI-Firmware.

Voraussetzung für das Neuübersetzen von Programmbausteinen

Nach dem Ersetzen einer CPU V3.0 durch eine CPU V4.x müssen Sie alle Programmbausteine neu übersetzen, damit Sie sie in die CPU der Version 4.x laden können. Außerdem müssen Sie bei Bausteinen mit Knowhow-Schutz (Seite 167) oder an eine PLC-Seriennummer gebundenem Kopierschutz (Seite 168) den Schutz aufheben, damit Sie die Bausteine übersetzen und laden können. (An eine Memory Card gebundenen Kopierschutz brauchen Sie jedoch nicht aufzuheben.) Nach dem erfolgreichen Übersetzen können Sie den Knowhow-Schutz und/oder den an eine PLC-Seriennummer gebundenen Kopierschutz erneut konfigurieren. Sollte Ihr Projekt von einem OEM (Original Equipment Manufacturer) bereitgestellte Bausteine mit Knowhow-Schutz enthalten, müssen Sie sich an den OEM wenden, damit er Ihnen diese Bausteine für die Version 4.x zur Verfügung stellt.

Im Allgemeinen empfiehlt Siemens, nach dem Gerätetausch die Hardwarekonfiguration und Software in STEP 7 neu zu übersetzen und in alle Geräte Ihres Projekts zu laden. Berichten Sie alle Fehler, die beim Übersetzen des Projekts auftauchen und wiederholen Sie die Übersetzung, bis keine Fehler mehr auftreten. Dann können Sie das Projekt in die CPU V4.x laden.

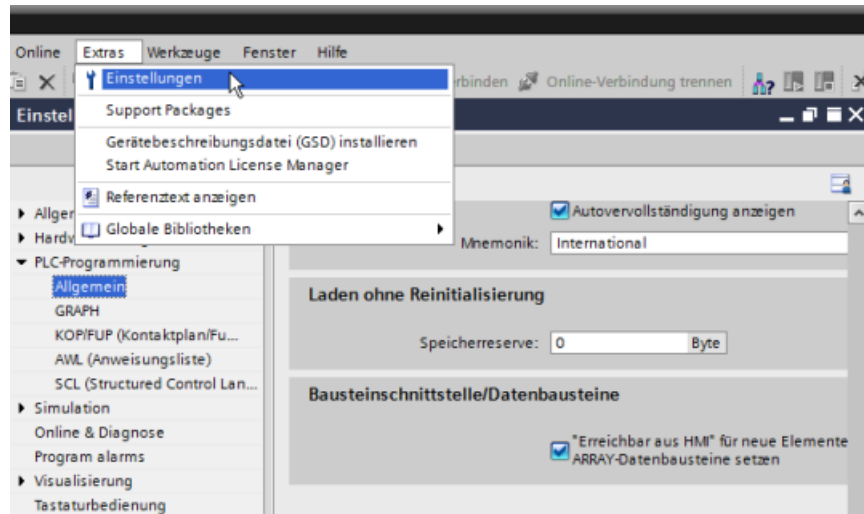
S7-1200 V3.0-Projekte passen möglicherweise nicht in S7-1200 CPUs V4.x.

S7-1200 ab V4.0 ergänzt jeden DB um einen Reservebereich von 100 Bytes, um das Laden ohne Neuinitialisierung zu unterstützen.

Sie können den 100-Byte-Reservebereich vor dem Laden eines V3.0-Projekts in eine CPU V4.x aus den DBs löschen.

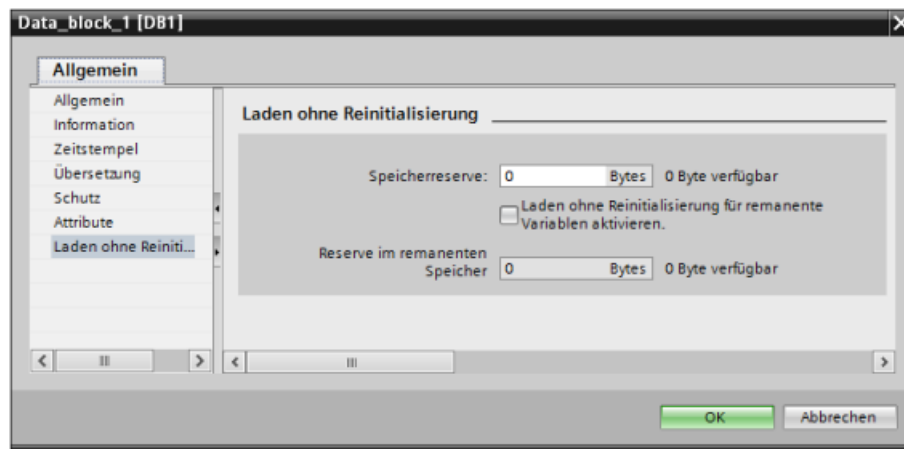
Um den 100-Byte-Reservebereich zu löschen, gehen Sie vor dem Gerätetausch wie folgt vor:

1. Wählen Sie im Hauptmenü im TIA-Portal den Menübefehl "Optionen" > "Einstellungen".
2. Öffnen Sie in der Projektnavigation den Knoten "PLC-Programmierung" > "Allgemein".
3. Legen Sie im Bereich "Laden ohne Reinitialisierung" eine Speicherreserve von 0 Bytes fest.



Wenn Sie den Gerätetausch bereits durchgeführt haben, müssen Sie die 100-Byte-Reserve in jedem Baustein einzeln löschen:

1. Klicken Sie in der Projektnavigation im Ordner der Programmbausteine mit der rechten Maustaste auf einen Datenbaustein und wählen Sie im Kontextmenü "Eigenschaften".
2. Wählen Sie im Eigenschaftsdialog des Datenbausteins den Knoten "Laden ohne Reinitialisierung".
3. Legen Sie eine Speicherreserve von 0 Bytes fest.
4. Wiederholen Sie diesen Vorgang für jeden Datenbaustein in Ihrem Projekt.



D.3 S7-1200 bis V3.0 - Ersatzklemmenblöcke

Tabelle D-1 S7-1200 CPU bis V3.0 - Ersatzklemmenblöcke

Sie haben eine S7-1200 CPU bis V3.0 (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)	
	Artikelnummer Klemmenblöcke	Beschreibung der Klemmenblöcke
CPU 1211C DC/DC/DC (6ES7211-1AE31-0XB0)	6ES7292-1BC30-0XA0	3-polig, vergoldet
CPU 1211C AC/DC/Relais (6ES7211-1BE31-0XB0)	6ES7292-1AH30-0XA0	8-polig, vergoldet
CPU 1211C DC/DC/Relais (6ES7211-1HE31-0XB0)	6ES7292-1AP30-0XA0	14-polig, verzinnt
CPU 1212C DC/DC/DC (6ES7212-1AE31-0XB0)		
CPU 1212C AC/DC/Relais (6ES7212-1BE31-0XB0)		
CPU 1212C DC/DC/Relais (6ES7212-1HE31-0XB0)		
CPU 1214C DC/DC/DC (6ES7214-1AG31-0XB0)	6ES7292-1BC30-0XA0	3-polig, vergoldet
CPU 1214C AC/DC/Relais (6ES7214-1BG31-0XB0)	6ES7292-1AM30-0XA0	12-polig, verzinnt
CPU 1214C DC/DC/Relais (6ES7214-1HG31-0XB0)	6ES7292-1AV30-0XA0	20-polig, verzinnt
CPU 1215C DC/DC/DC (6ES7215-1AG31-0XB0)	6ES7292-1BF30-0XB0	6-polig, vergoldet
CPU 1215C AC/DC/Relais (6ES7215-1BG31-0XB0)	6ES7292-1AM30-0XA0	12-polig, verzinnt
CPU 1215C DC/DC/Relais (6ES7215-1HG31-0XB0)	6ES7292-1AV30-0XA0	20-polig, verzinnt

Tabelle D-2 S7-1200 SMs bis V3.0 - Ersatzklemmenblöcke

Sie haben ein S7-1200 SM bis V3.0 (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)	
	Artikelnummer Klemmenblöcke	Beschreibung der Klemmenblöcke
SM 1221 DI 8 x DC (6ES7221-1BF30-0XB0)	6ES7292-1AG30-0XA0	7-polig, verzinnt
SM 1222 DQ 8 x DC (6ES7222-1BF30-0XB0)		
SM 1222 DQ 8 x Relais (6ES7222-1HF30-0XB0)		
SM 1231 AI 4 x 13 Bit (6ES7231-4HD30-0XB0)	6ES7292-1BG30-0XA0	7-polig, vergoldet
SM 1232 AQ 2 x 14 Bit (6ES7232-4HB30-0XB0)		
SM 1231 AI 4 x TC (6ES7231-5QD30-0XB0)		
SM 1231 AI 4 x 16 Bit (6ES7231-5ND30-0XB0)		
SM 1221 DI 16 x DC (6ES7221-1BH30-0XB0)	6ES7292-1AG30-0XA0	7-polig, verzinnt
SM 1222 DQ 16 x DC (6ES7222-1BH30-0XB0)		
SM 1222 DQ 16 x Relais (6ES7222-1HH30-0XB0)		
SM 1223 DI 8 x DC / DQ 8x DC (6ES7223-1BH30-0XB0)		
SM 1223 8 x DC / 8 x Relais (6ES7223-1PH30-0XB0)		
SM 1223 8 x AC / 8 x Relais (6ES7223-1QH30-0XB0)		
SM 1234 AI 4 / AQ 2 (6ES7234-4HE30-0XB0)	6ES7292-1BG30-0XA0	7-polig, vergoldet
SM 1231 AI 8 x 13 Bit (6ES7231-4HF30-0XB0)		
SM 1232 AQ 4 x 14 Bit (6ES7232-4HD30-0XB0)		
SM 1231 AI 4 x RTD (6ES7231-5PD30-0XB0)		
SM 1231 AI 8 x TC (6ES7231-5QF30-0XB0)		

Sie haben ein S7-1200 SM bis V3.0 (Artikelnummer)	Dann verwenden Sie diese Ersatzklemmenblöcke (4/Packung)	
	Artikelnummer Klemmenblöcke	Beschreibung der Klemmenblöcke
SM 1222 DQ 8 x Relais (Umschaltung) (6ES7222-1XF30-0XB0)	6ES7292-1AL30-0XA0	11-polig, verzinkt
SM 1223 DI 16 x DC/DQ 16 x DC (6ES7223-1BL30-0XB0)		
SM 1223 16 x DC/16 x Relais (6ES7223-1PL30-0XB0)		
SM 1231 AI 8 x RTD (6ES7231-5PF30-0XB0)	6ES7292-1BL30-0XA0	11-polig, vergoldet

Index

&

&-Box (logische UND-Verknüpfung in FUP), 210

/

/=-Box (Zuweisung negieren in FUP), 211

=

=-Box (Zuweisung in FUP), 211

>

>=1-Box (logische ODER-Verknüpfung in FUP), 210

A

Abfragearchitektur, 988

Abfragearchitektur Master, 988

Abfragearchitektur Slave, 989

ABS (Absolutwert bilden), 239

Abstand, Luftströmung und Kühlung, 44

AC

Erdung, 62

Richtlinien für Potentialtrennung, 61

Verdrahtungsrichtlinien, 60, 62

Achssteuerungsbefehle als Bewegungsfolge ausführen (MC_CommandTable), 581

ACOS (Arcuscosinuswert bilden), 242

ACT_TINT (Uhrzeitalarm aktivieren), 413

ADD (Addieren), 236

Ad-hoc-Modus, TCP und ISO-on-TCP, 620

Adressen

MAC-Adresse mit GetStationInfo lesen, 439

Teilnehmeradresse mit GetStationInfo lesen, 439

Adressen im Speicher, 98, 100

Adressierung

Boolesche Werte oder Bitwerte, 99

Einzelne Eingänge (E) oder Ausgänge (A), 99

Prozessabbild, 99

Speicherbereiche, 99

Aktive/passive Kommunikation

Parameter, 624

Partner konfigurieren, 601, 805

Verbindungs-IDs, 621

Aktive/passive Verbindung, 600

Aktualisierungs-OB, 84

Alarmer

Alarmlatenz, 86

ATTACH (OB einem Alarmereignis zuweisen), 404

CAN_DINT (Verzögerungsalarm abbrechen), 414

DETACH (Zuweisung eines OB zu einem

Alarmereignis aufheben), 404

QRY_DINT (Status des Verzögerungsalarms abfragen), 414

SRT_DINT (Verzögerungsalarm starten), 414

Übersicht, 75

Aliasnamen in benutzerdefinierten Webseiten, 904

Alte Anweisung TSEND_C (Daten über Ethernet senden (TCP)), 650

Alte Anweisungen TCON, TDISCON, TSEND und TRCV, 674

Alte Anweisung TRCV_C (Daten über Ethernet empfangen (TCP)), 650

An-/Abmelden, Standard-Webseiten, 864

Analoge E/A

Darstellung Ausgang (Spannung), 1359, 1401

Darstellung Ausgang (Strom), 1359, 1402

Darstellung Eingang (Spannung), 1358, 1400

Darstellung Eingang (Strom), 1358, 1401

Konfiguration, 170

Schrittantwortzeiten (CPU), 1271, 1284, 1296, 1308, 1324

Schrittantwortzeiten (SB), 1400

Schrittantwortzeiten (SM), 1357

Statusanzeigen, 1207, 1209

Umwandlung in physikalische Einheiten, 104, 292

Analoge Signalboards

SB 1231, 1396

SB 1231 RTD, 1408

SB 1231 Thermoelement, 1403

SB 1232, 1398

Analoge Signalmodule

SM 1231, 1347

SM 1231 RTD, 1366

SM 1231 Thermoelement, 1360

SM 1232, 1352

SM 1234, 1354

Analogwerte normieren, 292

Analogwerte skalieren, 292

Ändern

Beobachtungstabelle, 1226

Variablen über Webserver, 879

Zustand im Programmiereditor, 1224

- Anlauf nach NETZ-EIN, 71
 - Anlaufverarbeitung, 74
- Anlauf-OB, 76
- Anlaufparameter, 121
- Anschlüsse
 - Ethernet-Protokolle, 803
 - Kommunikationsarten, 589
 - Konfiguration, 624
 - Partner, 601, 805
 - S7-Verbindung, 803
 - Typen, Verbindungen mit mehreren Teilnehmern, 803
 - Verbindungs-IDs, 621
 - Webservers, 937
- Anschlusskontakte
 - Maximale Belastbarkeit, 1424
- Antriebe, Einrichten des MM4-Antriebs, 1016
- Anweisung DeviceStates (Modulstatus eines E/A-Systems lesen), 449
- Anweisung GEO2LOG (Hardwarekennung über Steckplatz ermitteln), 528
- Anweisung Get_IM_Data (Identifikations- und Wartungsdaten lesen), 431
- Anweisung Modbus_Comm_Load (SIPLUS I/O oder Port am PtP-Modul für Modbus RTU konfigurieren), 1092
- Anweisungen
 - &-Box (logische UND-Verknüpfung in FUP), 210
 - (-) (Schließerspule), 211
 - (/)- (Öffnerspule), 211
 - (N)- (Operand bei negativer Signalflanke setzen), 215
 - (P)- (Operand bei positiver Signalflanke setzen), 215
 - (RESET_BF) (Bitfeld rücksetzen), 213
 - (SET_BF) (Bitfeld setzen), 213
 - /=-Box (Zuweisung negieren in FUP), 211
 - |/|- (Öffnerkontakt), 209, 267, 271, 276, 280
 - ||- (Schließerkontakt), 209
 - |N|- (Operand auf negative Signalflanke abfragen), 214
 - |P|- (Operand auf positive Signalflanke abfragen), 214
 - =-Box (Zuweisung in FUP), 211
 - >=1-Box (logische ODER-Verknüpfung in FUP), 210
 - ABS (Absolutwert bilden), 239
 - ACOS (Arcuscosinuswert bilden), 242
 - ACT_TINT (Uhrzeitalarm aktivieren), 413
 - ADD (Addieren), 236
 - Alte Anweisung TRCV_C (Daten über Ethernet empfangen (TCP)), 650
 - Alte Anweisung TSEND_C (Daten über Ethernet senden (TCP)), 650
 - Alte Anweisungen TCON, TDISCON, TSEND und TRCV, 674
 - Alte USS-Statuscodes, 1145
 - Anweisung DeviceStates (Modulstatus eines E/A-Systems lesen), 449
 - Anweisung GEO2LOG (Hardwarekennung über Steckplatz ermitteln), 528
 - Anweisung Get_IM_Data (Identifikations- und Wartungsdaten lesen), 431
 - Anweisung ModuleStates (Modulstatusinformationen eines Moduls lesen), 454
 - ASIN (Arcussinuswert bilden), 242
 - ATAN (Arcustangenswert bilden), 242
 - ATH (ASCII-Zeichenkette in Hexadezimalzahl umwandeln), 348
 - ATTACH (OB einem Alarmereignis zuweisen), 404
 - ATTR_DB (Attribute eines Datenbausteins lesen), 525
 - Ausgang rücksetzen, 212
 - Ausgang setzen, 212
 - Beobachten, 1224
 - Bewegungssteuerung, 575
 - CALCULATE (Berechnen), 235
 - CAN_DINT (Verzögerungsalarm abbrechen), 414
 - CAN_TINT (Uhrzeitalarm löschen), 412
 - CASE (SCL), 312
 - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 290
 - Chars_TO_Strg (Array aus CHAR in Zeichenkette umwandeln), 346
 - CONCAT (Zeichenketten verketteten), 352
 - CONTINUE (SCL), 316
 - CONV (Wert umwandeln), 285
 - COS (Cosinuswert bilden), 242
 - CountOfElements (Anzahl ARRAY-Elemente abfragen), 264
 - CREATE_DB (Datenbaustein erstellen), 518
 - CTD (Rückwärtszählen), 225
 - CTRL_HSC (Schnellen Zähler steuern), 571
 - CTRL_HSC_EXT (Schnellen Zähler steuern (erweitert)), 544
 - CTRL_PTO (Impulsfolge), 472
 - CTRL_PWM (Impulsdauermodulation), 470
 - CTU (Vorwärtszählen), 225
 - CTUD (Vorwärts- und Rückwärtszählen), 225
 - DataLogClose (Datenprotokoll schließen), 504
 - DataLogCreate (Datenprotokoll erstellen), 495
 - DataLogNewFile (Datenprotokoll in neuer Datei), 507

- DataLogOpen (Datenprotokoll öffnen), 499
 DataLogWrite (Datenprotokoll schreiben), 501
 Datum, 327
 DB_ANY_TO_VARIANT (DB_ANY in VARIANT konvertieren), 295
 DEC (Dekrementieren), 239
 DECO (Decodieren), 320
 DELETE (Zeichen in einer Zeichenkette löschen), 354
 DELETE_DB (Datenbaustein löschen), 527
 DEMUX (Demultiplexen), 323
 Deserialize, 248
 DETACH (Zuweisung eines OB zu einem Alarmereignis aufheben), 404
 Dezentrale Peripherie AS-i, 371
 Dezentrale Peripherie PROFIBUS, 371
 Dezentrale Peripherie PROFINET, 371
 DIS_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen deaktivieren), 417
 DIV (Dividieren), 236
 DPNRM_DG (Diagnosedaten eines DP-Slaves lesen), 401
 DPRD_DAT (Konsistente Daten eines DP-Normslaves lesen), 393
 DPWR_DAT (Konsistente Daten eines DP-Normslaves schreiben), 393
 EN_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen aktivieren), 417
 ENCO (Encodieren), 320
 ENDIS_PW (Passwort-Legitimierung einschränken und freigeben), 301
 EQ_ElemType (Vergleiche, ob Datentyp eines ARRAY-Elements GLEICH dem Datentyp einer Variablen ist), 233
 EQ_Type (Vergleiche, ob Datentyp GLEICH dem Datentyp einer Variablen ist), 233
 EXIT (SCL), 317
 EXP (Exponentialwert bilden), 242
 EXPT (Potenzieren), 242
 F_TRIG (Variable bei negativer Signalflanke setzen), 216
 FeldLesen (Feld lesen), 265
 FeldSchreiben (Feld schreiben), 265
 FileDelete, 540
 FileReadC, 535
 FileWriteC, 538
 FILL_BLK (Speicher mit Bitmuster belegen), 253
 FIND (Zeichen in einer Zeichenkette finden), 357
 FLOOR (Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen), 290
 FOR (SCL), 313
 Forcefunktion, 1230
 FRAC (Nachkommastellen ermitteln), 242
 Gemeinsame Parameter, 727
 Gen_UsrMsg (Anwenderdiagnosemeldungen erzeugen), 418
 GET (Daten aus entfernter CPU auslesen), 799
 GET_DIAG (Diagnoseinformationen lesen), 459
 GET_ERROR (Fehler lokal abrufen), 305
 GET_ERROR_ID (Fehler-ID lokal abrufen), 307
 Get_Features (Erweiterte Funktionen abrufen), 985
 GetBlockName (Name des Bausteins auslesen), 369
 GetInstanceName (Namen der Baustein-Instanz auslesen), 364
 GetInstancePath (Zusammengesetzten globalen Namen der Baustein-Instanz abfragen), 367
 GETSMCInfo, 466
 GetSMCInfo (Informationen über die Memory Card auslesen), 466
 GetSymbolName (Namen einer Variable am Eingangsparameter auslesen), 358
 GetSymbolPath (Zusammengesetzten globalen Namen der Eingangsparameterversorgung abfragen), 361
 GOTO (SCL), 317
 HTA (Hexadezimalzahl in ASCII-Zeichenkette umwandeln), 348
 IF-THEN (SCL), 311
 IN_Range (Wert innerhalb Bereich), 231
 INC (Inkrementieren), 239
 INSERT (Zeichen in einer Zeichenkette einfügen), 355
 INV (Einerkomplement erstellen), 319
 IO2MOD (Aus einer E/A-Adresse die Hardwarekennung ermitteln), 531
 IS_ARRAY (Auf ARRAY prüfen), 234
 IS_NULL (Abfrage nach Pointer GLEICH Null), 234
 JMP (Springen bei VKE = 1), 296
 JMP_LIST (Sprungliste definieren), 297
 JMPN (Springen bei VKE = 0), 296
 Kalender, 327
 Label (Sprungmarke), 296
 LED (LED-Status lesen), 430
 LEFT (linke Zeichen einer Zeichenkette lesen), 353
 LEN (Länge einer Zeichenkette ermitteln), 351
 LIMIT (Grenzwert setzen), 241
 LN (Natürlichen Logarithmus bilden), 242
 LOG2GEO (Steckplatz über Hardwarekennung ermitteln), 530

- LOWER_BOUND (Untere ARRAY-Grenze auslesen), 255
- MAX (Maximum abrufen), 240
- MAX_LEN (Maximale Länge einer Zeichenkette), 351
- MB_CLIENT, 1022
- MB_RED_CLIENT, 1044
- MB_RED_SERVER, 1063
- MC_ChangeDynamic, 582
- MC_CommandTable, 581
- MC_Halt (Achse pausieren), 578
- MC_Home (Referenzpunktfahrt der Achse durchführen), 578
- MC_MoveAbsolute (Achse absolut positionieren), 579
- MC_MoveJog (Achse im Tippbetrieb bewegen), 580
- MC_MoveRelative (Achse relativ positionieren), 579
- MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 580
- MC_Power (Achse freigeben/sperrern), 577
- MC_ReadParam (Parameter eines Technologieobjekts lesen), 582
- MC_Reset (Fehler bestätigen), 577
- MC_WriteParam (Parameter des Technologieobjekts schreiben), 581
- MID (mittlere Zeichen einer Zeichenkette lesen), 353
- MIN (Minimum abrufen), 240
- MOD (Divisionsrest einer Division), 237
- Modbus_Comm_Load (SIPLUS I/O oder Port am PtP-Modul für Modbus RTU konfigurieren), 1092
- Modbus_Master (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Master kommunizieren), 1096
- Modbus_Slave (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Slave kommunizieren), 1103
- MOVE (Wert kopieren), 244
- MOVE_BLK (Bereich kopieren), 244
- MUL (Multiplizieren), 236
- MUX (Multiplexen), 322
- N (Operand auf negative Signalflanke abfragen), 214
- N_TRIG (VKE auf negative Signalflanke abfragen), 215
- N=Box und N-Spule (Operand bei negativer Signalflanke setzen), 215
- NE_ElemType (Vergleiche, ob Datentyp eines ARRAY-Elements UNGLEICH dem Datentyp einer Variablen ist), 233
- NE_Type (Vergleiche, ob Datentyp UNGLEICH dem Datentyp einer Variablen ist), 233
- NEG (Zweierkomplement erstellen), 238
- NORM_X (Normieren), 291
- NOT (VKE invertieren), 211
- NOT_NULL (Abfrage nach Pointer UNGLEICH Null), 234
- NOT_OK (Ungültigkeit prüfen), 232
- ODER (logische Verknüpfung), 318
- OK (Gültigkeit prüfen), 232
- OUT_Range (Wert außerhalb Bereich), 231
- P (Operand auf positive Signalflanke abfragen), 214
- P_TRIG (VKE auf positive Signalflanke abfragen), 215
- P=Box und P-Spule (Operand bei positiver Signalflanke setzen), 215
- P3964_Config (3964(R)-Protokoll konfigurieren), 975
- PEEK- und POKE-Varianten, 195, 258
- PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 585
- PID_Temp (universeller PID-Regler für die Temperaturregelung), 587
- Port_Config (Portkonfiguration), 965
- Programmsteuerung (SCL), 310
- PRVREC (Datensatz bereitstellen), 398
- PUT (Daten in entfernte CPU schreiben), 799
- QRY_CINT (Weckalarmparameter abfragen), 409
- QRY_DINT (Status des Verzögerungsalarms abfragen), 414
- QRY_TINT (Status des Uhrzeitalarms abfragen), 413
- R (Ausgang rücksetzen), 212
- R_TRIG (Variable bei positiver Signalflanke setzen), 216
- RALRM (Alarm empfangen), 380
- RCVREC (Datensatz empfangen), 396
- RD_ADDR (E/A-Adressen über Hardwarekennung ermitteln), 532
- RD_LOC_T (Lokalzeit lesen), 330
- RD_SYS_T (Uhrzeit lesen), 330
- RDREC (Datensatz lesen), 373
- RE_TRIGR, 90
- RE_TRIGR (Zyklusüberwachungszeit neu starten), 304
- READ_BIG (Daten in Big-Endian-Format lesen), 260
- READ_DBL (Aus Datenbaustein im Ladespeicher lesen), 522
- READ_LITTLE (Daten in Little-Endian-Format lesen), 260
- Receive_Config (Empfangskonfiguration), 970

- Receive_P2P (Punkt-zu-Punkt-Daten empfangen), 980
 Receive_Reset (Empfangspuffer löschen), 982
 RecipeExport (Rezeptexport), 487
 RecipelImport (Rezeptimport), 489
 REPEAT (SCL), 315
 REPLACE (Zeichen in einer Zeichenkette ersetzen), 356
 RESET_BF (Bitfeld rücksetzen), 213
 RET (Rückgabewert), 300
 RETURN (SCL), 318
 RIGHT (rechte Zeichen einer Zeichenkette lesen), 353
 ROL (Links rotieren) und ROR (Rechts rotieren), 325
 ROUND (Zahl runden), 289
 RS (Flipflop rücksetzen/setzen), 213
 RT (Zeit zurücksetzen), 217
 RTM (Betriebsstundenzähler), 334
 RUNTIME (Programmlaufzeit messen), 309
 S (Ausgang setzen), 212
 S_CONV (Zeichenkette umwandeln), 337
 S_MOV (Zeichenkette verschieben), 337
 SCALE_X (Skalieren), 291
 SEL (Selektieren), 321
 Send_Config (Sendekonfiguration), 968
 Send_P2P (Punkt-zu-Punkt-Daten senden), 977
 Serialize, 250
 SET_BF (Bitfeld setzen), 213
 SET_CINT (Weckalarm parametrieren), 407
 Set_Features (Erweiterte Funktionen setzen), 986
 SET_TIMEZONE (Zeitzone setzen), 333
 SET_TINTL (Datum und Uhrzeit eines Uhrzeitalarms festlegen), 411
 SGN_GET (RS232-Signale abrufen), 983
 SHL (Links schieben) und SHR (Rechts schieben), 324
 Signal_Set (RS232-Signale setzen), 984
 SIN (Sinuswert bilden), 242
 Spalten und Überschriften, 636, 649, 656, 673, 1000, 1021, 1091, 1133, 1147, 1165
 SQR (Quadrat bilden), 242
 SQRT (Quadratwurzel bilden), 242
 SR (Flipflop setzen/rücksetzen), 213
 SRT_DINT (Verzögerungsalarm starten), 414
 STP (Programm beenden), 305
 Strg_TO_Chars (Zeichenkette in Array aus CHAR umwandeln), 346
 STRG_VAL (Zeichenkette in Zahlenwert umwandeln), 337
 SUB (Subtrahieren), 236
 SWAP (Anordnung ändern), 255
 SWITCH (Sprungverteilung), 298
 T_ADD (Zeiten addieren), 328
 T_COMBINE (Zeiten verknüpfen), 329
 T_CONFIG (Schnittstelle konfigurieren), 717
 T_CONV (Zeiten umwandeln und extrahieren), 327
 T_DIAG, 684
 T_DIFF (Zeitdifferenz), 329
 T_RESET, 682
 T_SUB (Zeiten subtrahieren), 328
 TAN (Tangenswert bilden), 242
 TCON, 657
 TDISCON, 657
 TM_MAIL (E-Mail senden), 1197
 TOF (Ausschaltverzögerung), 217
 TON (Einschaltverzögerung), 217
 TONR (Speichernde Einschaltverzögerung), 217
 TP (Impuls), 217
 TRCV, 657
 TRCV_C, 636, 735
 TRUNC (Ganzzahl erzeugen), 289
 TSEND, 657
 TSEND_C, 636, 734
 TURCV (Daten über Ethernet empfangen (UDP)), 712
 TUSEND (Daten über Ethernet senden (UDP)), 712
 Überwachungsstatus oder -wert, 1223
 UFILL_BLK (Speicher ununterbrechbar mit Bitmuster belegen), 253
 Uhr, 330
 Uhrzeit, 327
 UMOVE_BLK (Bereich ununterbrechbar kopieren), 244
 Umwandlungsanweisungen in SCL, 286
 UND (logische Verknüpfung), 318
 UPPER_BOUND (Obere ARRAY-Grenze auslesen), 257
 USS_Drive_Control (Daten mit Antrieb tauschen), 1006
 USS_Port_Scan (Kommunikation über USS-Netzwerk bearbeiten), 1004
 USS_Read_Param (Parameter aus dem Antrieb auslesen), 1009
 USS_Write_Param (Parameter im Antrieb ändern), 1010
 USS-Statuscodes, 1012
 VAL_STRG (Zahlenwert in Zeichenkette umwandeln), 337
 VARIANT_TO_DB_ANY (VARIANT in DB_ANY konvertieren), 294
 VariantGet (VARIANT Variablenwert lesen), 262

- VariantPut (VARIANT Variablenwert schreiben), 263
- Versionen von Anweisungen, 636, 649, 656, 673, 1000, 1021, 1091, 1133, 1147, 1165
- Werte vergleichen, 230
- WHILE (SCL), 314
- WR_LOC_T (Lokalzeit einstellen), 330
- WR_SYS_T (Uhrzeit einstellen), 330
- WRIT_DBL (In Datenbaustein im Ladespeicher schreiben), 522
- WRITE_BIG (Daten in Big-Endian-Format schreiben), 260
- WRITE_LITTLE (Daten in Little-Endian-Format schreiben), 260
- WRREC (Datensatz schreiben), 373
- WWW (benutzerdefinierte Webseiten synchronisieren), 913
- x-Box (logische Exklusiv-ODER-Verknüpfung in FUP), 210
- XOR (logische Verknüpfung), 318
- Zeit, 217
- Zustand, 1224
- Anweisungen,
 - Ausführungsgeschwindigkeiten, 1265, 1277, 1289, 1302, 1316
 - Anweisungen, veraltet
 - MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren), 1148
 - MB_COMM_LOAD (Port am PtP-Modul für Modbus RTU konfigurieren), 1166
 - MB_MASTER (Über PtP-Port als Modbus-Master kommunizieren), 1169
 - MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren), 1155
 - MB_SLAVE (Über PtP-Port als Modbus-Slave kommunizieren), 1175
 - PORT_CFG (Kommunikationsparameter dynamisch konfigurieren), 1117
 - RCV_CFG (Serielle Empfangsparameter dynamisch konfigurieren), 1120
 - RCV_PTP (Empfangsmeldungen aktivieren), 1128
 - RCV_RST (Empfangspuffer löschen), 1129
 - SEND_CFG (Serielle Übertragungsparameter dynamisch konfigurieren), 1119
 - SEND_PTP (Sendepufferdaten übertragen), 1125
 - SGN_GET (RS232-Signale abfragen), 1130
 - SGN_SET (RS232-Signale festlegen), 1131
 - USS_DRV (Daten mit Antrieb tauschen), 1139
 - USS_PORT (Kommunikation über USS-Netzwerk bearbeiten), 1137
 - USS_RPM (Parameter aus dem Antrieb auslesen), 1142
 - USS_WPM (Parameter im Antrieb ändern), 1143
- Anwenderprogramm
 - Bausteine aus einer Online-CPU kopieren, 205
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - Laden in CPU, 199
 - Lineare und strukturierte Programme, 176
 - Memory Card, 118
 - Organisationsbausteine (OBs), 178
 - Passwortschutz, 167
 - Programmkarte, 118
 - Übertragungskarte, 118
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Anzeige erster Zyklus, 95
- Arbeitsspeicher, 28
 - CPU 1211C, 1263
 - CPU 1212C, 1276
 - CPU 1214C, 1288
 - CPU 1215C, 1300
 - CPU 1217C, 1314
- Arithmetik, 235, 236
- Arrays mit Variablen indexieren, 266
- Arrays, Zugriff auf Elemente, 266
- Artikelnummern
 - CPU 1214FC, CPU 1215FC, 1444
 - CPUs, 1441
 - CSM 1277 Ethernet Switch, 1445
 - Endhalter, 1446
 - Erweiterungskabel, 1446
 - FS Signalmodule, 1444
 - Grundlegende HMI-Panels, 1445
 - Klemmenblöcke, 1446
 - Kommunikationsschnittstellen (CM, CB und CP), 1443
 - Memory Cards, 1445
 - PM 1207 Stromversorgung, 1445
 - Programmiersoftware, 1452
 - Signalboards, Batterieboards, 1442
 - Signalmodule, 1441
 - Simulatoren, 1446
 - Steckverbinder und Klemmenanschlüsse, 1444
 - STEP 7, 1452
 - Visualisierungssoftware, 1452
 - WinCC, 1452
- ASi
 - Adresse, 791
 - Analogwerte übertragen, 795
 - AS-i-Master CM 1243-2, 788
 - AS-i-Mastermodul CM1243-2 hinzufügen, 789

- AS-i-Slave hinzufügen, 790
- Digitalwerte übertragen, 795
- Netzwerkverbindung, 790
- RDREC (Datensatz lesen), 373
- Slavekonfiguration mit STEP 7, 795
- Slavekonfiguration ohne STEP 7, 793
- Systemzuweisung, 793
- Systemzuweisung von Slaveadressen, 793
- WRREC (Datensatz schreiben), 373
- AS-i
 - Anweisungen für die dezentrale Peripherie, 371
- ASIN (Arcussinuswert bilden), 242
- ATEX-Zulassung, 1255
- ATH (ASCII-Zeichenkette in Hexadezimalzahl umwandeln), 348
- ATTACH (OB einem Alarmereignis zuweisen), 404
- ATTR_DB (Attribute eines Datenbausteins lesen), 525
- AT-Variablenüberlagerung, 116
- Auf Werkseinstellungen zurücksetzen, 1214
- Aufrufstruktur, 208
- Ausführungsgeschwindigkeiten der Anweisungen, 1265, 1277, 1289, 1302, 1316
- Ausgangsparameter, 181
 - Impulsausgänge, 475
 - Impulskanäle konfigurieren, 477
- Auslösen
 - Trace, 1242
 - Werte in der Beobachtungstabelle, 1228
- Ausschaltverzögerung (TOF), 217
- Australien und Neuseeland - RCM Mark-Zulassung, 1256
- Autonegotiation, 610
- AWP_Enum_Def, 905
- AWP_Import_Fragment, 908
- AWP_In_Variable, 899, 903
- AWP_Out_Variable, 901
- AWP_Start_Fragment, 907
- AWP-Befehle, 896
 - Alias verwenden, 904
 - Definitionen verbinden, 909
 - Enum-Typ definieren, 905
 - Enum-Typ referenzieren, 905
 - Fragmente generieren, 907
 - Fragmente importieren, 908
 - Sondervariablen lesen, 901
 - Sondervariablen schreiben, 903
 - Variablen schreiben, 899
- B**
 - Batterieboard (BB)
 - Batterie einlegen, 1414
 - BB 1297, 1413
 - Baudrate, 948
 - Baugruppenträger- oder Stationsfehler-OB ("Rack or station failure OB"), 82
 - Bausteine
 - Alarmer, 30, 86, 1266, 1278, 1290, 1303, 1317
 - Anfangswert eines FB, 181
 - Anlauf-OBs, 86
 - Anzahl der Codebausteine, 30, 67, 1265, 1278, 1290, 1302, 1316
 - Anzahl der OBs, 30, 86, 1266, 1278, 1290, 1303, 1317
 - Bausteinaufrufe, 67
 - Bausteine aus einer Online-CPU kopieren, 205
 - Beobachten, 30, 1265, 1278, 1290, 1302, 1316
 - Datenbaustein (DB), 67
 - Einzelinstanz- oder Multiinstanz-DB, 181
 - Ereignisse, 86
 - FB oder FC mit SCL aufrufen, 192
 - Funktion (FC), 67, 180
 - Funktionsbaustein (FB), 67, 181
 - Größe des Anwenderprogramms, 30, 67, 1265, 1278, 1290, 1302, 1316
 - Gültige FC-, FB- und DB-Nummern, 67
 - Instanz-Datenbaustein (DB), 181
 - Konsistenzprüfung, 208
 - Laden, 199
 - Lineare und strukturierte Programme, 176
 - Organisationsbausteine (OBs), 30, 67, 75, 86, 1266, 1278, 1290, 1303, 1317
 - Passwortschutz, 167
 - Schachtelungstiefe, 30, 67, 1265, 1278, 1290, 1302, 1316
 - Typen, 67
 - Zähler (Anzahl und Speicherbedarf), 30, 1266, 1278, 1290, 1303, 1317
 - Zeiten (Anzahl und Speicherbedarf), 30, 1266, 1278, 1290, 1303, 1317
 - Bausteine aus einer Online-CPU kopieren, 205
 - BB 1297, 1413
 - Bearbeiten im Betriebszustand RUN, (Laden im Betriebszustand RUN)
 - Bedienoberfläche
 - STEP 7 Projekt- und Portalansicht, 39
 - Bedienpanel, 32

- Beispiele, Anweisungen
 - ATH (ASCII-Zeichenkette in Hexadezimalzahl wandeln), 349
 - CONTINUE, SCL, 316
 - CTRL_PWM, 479
 - DECO (Decodieren), 321
 - Deserialize, 249
 - DeviceStates, PROFIBUS und PROFINET, 450
 - EXIT, SCL, 317
 - GET_DIAG und Modi, 465
 - GOTO (SCL), 317
 - HTA (Hexadezimalzahl in ASCII-Zeichenkette wandeln), 350
 - LIMIT (Grenzwert setzen), 242
 - ModuleStates, PROFIBUS und PROFINET, 455
 - PEEK- und POKE-Varianten, 195, 258
 - RETURN, SCL, 318
 - ROR (Rechts rotieren), SCL, 326
 - RUNTIME (Programmlaufzeit messen), 309
 - S_CONV (Zeichenkette umwandeln), 344
 - Serialize, 252
 - SET_CINT Weckalarmausführung und Zeitparameter, 408
 - SHL (Links schieben), SCL, 325
 - STRG_VAL (Zeichenkette in Zahlenwert umwandeln), 345
 - SWAP (Anordnung ändern), 255
 - TM_MAIL, 1202
 - VAL_STRG (Zahlenwert in Zeichenkette umwandeln), 346
 - Zeitspulen, 219
- Beispiele, Kommunikation
 - Asi-Slaveadressierung, 791
 - CPU-Kommunikation mit einer gemeinsamen Verbindung zum Senden und Empfangen von Daten, 622
 - CPU-Kommunikation mit getrennten Verbindungen zum Senden und Empfangen von Daten, 621
 - CPU-Kommunikation über TSEND_C- oder TRCV_C-Verbindungen, 623
 - I-Device als IO-Gerät und IO-Controller, 744
 - PROFIBUS S7-Verbindung konfigurieren, 810
 - PROFINET S7-Verbindung konfigurieren, 808
 - PROFINET-Kommunikationsprotokolle, 617
 - Shared Device, 752
 - Shared I-Device, 757
 - T_CONFIG, IP-Parameter ändern, 723
 - T_CONFIG, IP-Parameter der NTP-Server ändern, 725
 - T_CONFIG, IP-Parameter und PROFINET IO-Device-Namen ändern, 724
 - Telecontrol, 1192
- Beispiele, Modbus
 - Beispielprogramm für einen Modbus RTU-Master, 1113
 - Beispielprogramm für einen Modbus RTU-Slave, 1116
 - MB_CLIENT für mehrere Anforderungen mit gemeinsamer TCP-Verbindung, 1085
 - MB_CLIENT für mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen, 1086
 - MB_SERVER für mehrere TCP-Verbindungen, 1084
 - Modbus TCP MB_CLIENT für die Koordination mehrerer Anforderungen, 1088
 - Modbus TCP MB_CLIENT Schreibenanforderung für das Prozessabbild der Ausgänge, 1087
 - Modbus TCP, Beispiele für Parameter von MB_HOLD_REG, 1036
 - Modbus TCP, Halteregeisteradresse, 1039
 - Modbus TCP, MB_CLIENT-Verbindungsparameter, 1028
 - Modbus TCP, MB_SERVER-Verbindungsparameter, 1034
- Beispiele, Modbus CP in älteren Systemen
 - Parameter von MB_HOLD_REG, 1157
- Beispiele, Modbus in älteren Systemen
 - Beispiele für Modbus RTU, Parameter von MB_HOLD_REG in älteren Systemen, 1176
 - Modbus RTU, Halteregeisteradressierung in älteren Systemen, 1179
- Beispiele, Modbus RTU in älteren Systemen
 - Master-Programm, 1181
 - Slave-Programm, 1183
- Beispiele, Modbus TCP in älteren Systemen
 - Halteregeisteradressierung, 1159
 - MB_CLIENT für die Koordination mehrerer Modbus TCP-Anforderungen, 1163
 - MB_CLIENT für mehrere Anforderungen mit unterschiedlichen TCP-Verbindungen, 1162
 - MB_CLIENT Schreibenanforderung für das Prozessabbild der Ausgänge, 1163
 - MB_CLIENT: mehrere Anforderungen mit gemeinsamer Modbus TCP-Verbindung, 1161
 - MB_SERVER für mehrere TCP-Verbindungen, 1160
- Beispiele, PtP-Kommunikation
 - Bedingung für das Meldungsende, 958
 - Bedingung für den Meldungsbeginn, 954
 - Beispiel für den Terminalemulator ausführen, 997
 - Konfiguration, 990
 - Meldungslänge innerhalb der Meldung, 958

- PtP-Kommunikation in älteren Systemen, RCV_CFG, 1123
- Receive_Config, 972
- STEP 7-Programmierung, 995
- Terminalemulator, 989, 997
- Beispiele, USS-Kommunikation
 - Meldung von USS-Kommunikationsfehlern, 1013
 - Meldung von USS-Kommunikationsfehlern in älteren Systemen, 1146
- Beispiele, verschiedene
 - Analogwertverarbeitung, 104, 292
 - AT-Variablenüberlagerung, 116
 - Ausgewählte Bausteine im Betriebszustand RUN laden, 1235
 - Berechnung der Leistungsbilanz, 1438
 - CPU 1217C Differentialausgang und Anwendung, 1329
 - CPU 1217C Differentialeingang und Anwendung, 1328
 - Datenprotokollprogramm, 514
 - ENO-Auswertung in SCL, 198
 - Konfigurationssteuerung (Optionsverwaltung), 148
 - Rezept, 483, 491
 - S7-1200 IO-Link-Master Anschluss, 1377
 - Slice eines Variablentyps, 115
 - Trace- und Logic-Analyzer-Funktion, 1244
 - verschachtelte CASE-Anweisungen, SCL, 313
 - Zugriff auf Array-Elemente, 266
- Beispiele, Webserver
 - Aliasnamen, 898, 904
 - AWP-Deklarationen verbinden, 909
 - benutzerdefinierte Webseite, 917, 922
 - benutzerdefinierte Webseite für die Spracheinstellung, 929
 - DB-Fragmente, 908
 - Enum-Typen, 905, 906, 920
 - Sondervariablen lesen, 902
 - Sondervariablen schreiben, 903, 922
 - Sonderzeichen in AWP-Befehlen, 910
 - STEP 7-Programm für die Prüfung auf Fragmente, 935
 - Variablen lesen, 898, 919
 - Variablen schreiben, 900, 921
 - Zugriff über mobiles Gerät, 857
- Beispiele, Zeichenkettenanweisungen während der Laufzeit
 - GetBlockName, 370
 - GetInstanceName, 366
 - GetInstancePath, 367
 - GetSymbolName, 360
 - GetSymbolPath, 363
- Bemessungsspannungen, 1260
- Benutzerdefinierte Webseiten, 848, 895
 - Aktivieren mit WWW-Anweisung, 913
 - Aktualisieren, 896
 - Anlegen mit HTML-Editor, 895
 - AWP-Befehle für den Zugriff auf S7-1200 Daten, 896
 - Beispiel, 917
 - Einschränkungen Ladespeicher, 916
 - Entsprechende DBs in die CPU laden, 915
 - Fragmente erstellen, 907
 - Fragmente importieren, 908
 - Handhabung von Sonderzeichen, 909
 - HTML-Liste, 922
 - Konfiguration für mehrere Sprachen, 931
 - Konfigurieren, 911
 - Manuelle Steuerung von DB-Fragmenten, 932
 - mehrere Sprachen, 928
 - Programmbausteine generieren, 912
 - Programmbausteine löschen, 913
 - Programmieren in STEP 7, 913
 - Sondervariablen lesen, 901
 - Sondervariablen schreiben, 903
 - Über Steuer-DB aktivieren und deaktivieren, 932
 - Variablen lesen, 898
 - Variablen schreiben, 899
 - Zugriff über PC, 915
- Benutzerdefinierte Webseiten aktualisieren, 896
- Benutzerdefinierte Webseiten anlegen, 895
- Benutzerkonfiguration, Webserver, 852
- Beobachten
 - Beobachtungstabelle, 1226
 - DB-Werte erfassen und zurücksetzen, 1225
 - Forcefunktion, 1230
 - Forcetabelle, 1230
 - KOP-Status und Verwendung der Beobachtungstabelle, 1223
 - KOP-Zustand, 1224
 - Speicherauslastung, 1219
 - Zykluszeit, 1219
- Beobachtungstabelle
 - Ausgänge im Betriebszustand STOP freischalten, 1229
 - Beobachten, 1223
 - Forcen, 206
 - Funktionsweise, 1226
 - Werte auslösen, 1228
- Betriebsart für PG/PC- und HMI-Kommunikation, 160
- Betriebsphase
 - HSC (schneller Zähler), 562
- Betriebsstundenzähler (RTM), 334

- Betriebszustand
 - Betriebszustände der CPU, 71
 - STOP/RUN wechseln, 1218
 - Betriebszustand RUN, 71, 74, 1218
 - Forcefunktion, 1230
 - Betriebszustand STOP, 71, 1218
 - Ausgänge im Betriebszustand STOP freischalten, 1229
 - Forcefunktion, 1230
 - Bewegungsfolge (MC_CommandTable), 581
 - Bewegungssteuerung
 - MC_CommandTable, 581
 - MC_Halt (Achse pausieren), 578
 - MC_Home (Referenzpunktfahrt der Achse durchführen), 578
 - MC_MoveAbsolute (Achse absolut positionieren), 579
 - MC_MoveJog (Achse im Tippbetrieb bewegen), 580
 - MC_MoveRelative (Achse relativ positionieren), 579
 - MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 580
 - MC_Power (Achse freigeben/sperrern), 577
 - MC_ReadParam (Parameter eines Technologieobjekts lesen), 582
 - MC_Reset (Fehler bestätigen), 577
 - MC_WriteParam (Parameter des Technologieobjekts schreiben), 581
 - Übersicht, 574
 - Bewegungssteuerungsanweisungen, 575
 - Bibliothek für das alte USS-Protokoll
 - Statuscodes, 1145
 - Übersicht, 1132
 - USS_DRV (Daten mit Antrieb tauschen), 1139
 - USS_PORT (Kommunikation über USS-Netzwerk bearbeiten), 1137
 - USS_RPM (Parameter aus dem Antrieb auslesen), 1142
 - USS_WPM (Parameter im Antrieb ändern), 1143
 - Voraussetzungen für den Einsatz, 1135
 - Bibliothek für das USS-Protokoll
 - Statuscodes, 1012
 - Übersicht, 998
 - USS_Drive_Control (Daten mit Antrieb tauschen), 1006
 - USS_Port_Scan (Kommunikation über USS-Netzwerk bearbeiten), 1004
 - USS_Read_Param (Parameter aus dem Antrieb auslesen), 1009
 - USS_Write_Param (Parameter im Antrieb ändern), 1010
 - Voraussetzungen für den Einsatz, 1002
 - Bitverknüpfung
 - NOT-Operation (VKE invertieren), 211
 - Operationen AND (UND), OR (ODER) und XOR (EXKLUSIV ODER), 210
 - Operationen Steigende Flanke und Fallende Flanke, 215
 - Schließer- und Öffnerkontakte, 209
 - Schließer- und Öffnerspulen, 211
 - Setz- und Rücksetzoperationen, 212
 - Boolesche Werte oder Bitwerte, 99
 - Busstecker, 32
- ## C
- CALCULATE (Berechnen), 235
 - CAN_DINT (Verzögerungsalarm abbrechen), 414
 - CAN_TINT (Uhrzeitalarm löschen), 412
 - CANopen-Module
 - 021620-B, 021630-B, 1434
 - CB 1241
 - Abschluss, 943
 - CB 1241 RS485, 1425
 - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 290
 - CE-Zulassung, 1253
 - Char (Datentyp Character), 111
 - Chars_TO_Strg (Array aus CHAR in Zeichenkette umwandeln), 346
 - Codebaustein
 - Alarmer, 30, 1266, 1278, 1290, 1303, 1317
 - Anfangswert eines FB, 181
 - Anzahl der Codebausteine, 30, 67, 1265, 1278, 1290, 1302, 1316
 - Anzahl der OBs, 30, 1266, 1278, 1290, 1303, 1317
 - Bausteinaufrufe, 67
 - Beobachten, 1265, 1278, 1290, 1302, 1316
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - DB (Datenbaustein), 67, 182
 - FB (Funktionsbaustein), 67, 181
 - FC (Funktion), 67, 180
 - Größe des Anwenderprogramms, 30, 1265, 1278, 1290, 1302, 1316
 - Gültige FC-, FB- und DB-Nummern, 67
 - Instanz-Datenbaustein (DB), 181
 - Knowhow-Schutz, 167
 - Kopierschutz, 168
 - Lineare und strukturierte Programme, 176

- Organisationsbausteine (OBs), 30, 178, 1266, 1278, 1290, 1303, 1317
- Schachtelungstiefe, 30, 1265, 1278, 1290, 1302, 1316
- Überwachung, 30
- Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Zähler (Anzahl und Speicherbedarf), 30, 1266, 1278, 1290, 1303, 1317
- Zeiten (Anzahl und Speicherbedarf), 30, 1266, 1278, 1290, 1303, 1317
- Codebausteine im Anwenderprogramm aufrufen, 178
- Compact Switch Module, CSM 1277, 1434
- CONCAT (Zeichenketten verketteten), 352
- CONTINUE, SCL, 316
- CONV (Wert umwandeln), 285
- Cookie siemens_automation_language, 929
- Cookie, siemens_automation_language, 929
- Cookie-Einschränkungen, Standard-Webseiten, 938
- COS (Cosinuswert bilden), 242
- CountOfElements (Anzahl ARRAY-Elemente abfragen), 264
- CP-Modul
 - Webserver Startseite, 867
 - Zugriff auf Webserver, 858
- CPU
 - Anlaufparameter, 121
 - Anlaufverarbeitung, 74
 - Anzahl der Kommunikationsverbindungen, 596
 - ASi, 790
 - ASi-Adresse, 791
 - AS-i-Port, 790
 - Auf Werkseinstellungen zurücksetzen, 1214
 - Ausgänge im Betriebszustand STOP freischalten, 1229
 - Bausteine aus einer Online-CPU kopieren, 205
 - Bausteine vergleichen und synchronisieren, 1221
 - Bedienpanel (Online-CPU), 1218
 - Beobachtungstabelle, 1226
 - Betriebsarten RUN/STOP, 1218
 - Betriebszustände, 71
 - DB-Werte erfassen und zurücksetzen, 1225
 - Eigenschaft Uhrzeitsynchronisation, 615
 - Einbau, 48, 49
 - Erdung, 62
 - Erweiterungskabel, 58
 - Ethernet-Port, 608
 - Forcen, 1230
 - Gerätekonfiguration, 135
 - HSC-Konfiguration, 560
 - Impulsausgänge, 475
 - induktive Lasten, 64
 - IP-Adresse, 608
 - Klemmenblock, 57
 - Knowhow-Schutz, 167
 - Kommunikation, 598
 - Kommunikationsarten, 589
 - Kommunikationsboards (CB), 31
 - Kommunikationslast, 91
 - Laden in CPU, 199
 - Laden in Gerät, 613
 - Lampenlasten, 63
 - LED-Anzeigen, 1205
 - Leere Übertragungskarte, 134
 - Leistungsbedarf, 1437
 - Leistungsbilanz, 45
 - MAC- und IP-Adressen anzeigen, 613
 - MAC-Adresse, 608, 613
 - Module hinzufügen, 140
 - Netzwerkverbindung, 600
 - Neues Gerät hinzufügen, 137
 - Nicht spezifizierte CPU, 138
 - OBs verarbeiten, 178
 - Online, 1212
 - Online gehen, 1209
 - PROFIBUS-Adresse, 786
 - PROFINET IO, 737
 - PROFINET-Port, 608
 - Programmausführung, 67
 - Richtlinien für Potentialtrennung, 61
 - RTM (Betriebsstundenzähler), 334
 - Schutzstufen, 163
 - Sichern, 1249
 - Signalboards (SB), 31
 - Thermischer Bereich, 44, 47
 - Überlastverhalten, 88
 - Übersicht, 27
 - Überwachung online, 1223
 - Verdrahtungsrichtlinien, 60, 62
 - Vergleichstabelle, 28
 - Verlorenes Passwort, 134
 - Versionskompatibilität, 40
 - Vorgehensweise bei verlorenem Passwort, 134
 - Wiederherstellen einer Sicherungskopie, 1252
 - Zugriffsschutz durch Passwörter, 163
 - Zuweisen einer IP-Adresse zu einer Online-CPU, 606
 - Zykluszeitkonfiguration, 91
- CPU Memory Card
 - Einfügen, 118
 - Programmkarte, 126
 - Übertragungskarte, 122

- CPU-Eigenschaften, benutzerdefinierte Webseiten
 - Mehrere Sprachen einrichten, 931
 - STEP 7-Konfiguration, 911
 - CPU-Konfiguration
 - Betriebsparameter, 153
 - Impulskanäle, 477
 - Kommunikation mit HMI, 731
 - Mehrere CPUs, 732
 - Moduleigenschaften, 169
 - Zykluszeitüberwachung, 90
 - CPUs
 - CPU 1211C AC/DC/Relais, 1263
 - CPU 1211C DC/DC/DC, 1263
 - CPU 1211C DC/DC/Relais, 1263
 - CPU 1212C AC/DC/Relais, 1276
 - CPU 1212C DC/DC/DC, 1276
 - CPU 1212C DC/DC/Relais, 1276
 - CPU 1214C AC/DC/Relais, 1288
 - CPU 1214C DC/DC/DC, 1288
 - CPU 1214C DC/DC/Relais, 1288
 - CPU 1215C AC/DC/Relais, 1300
 - CPU 1215C DC/DC/DC, 1300
 - CPU 1215C DC/DC/Relais, 1300
 - CPU 1217C DC/DC/DC, 1314
 - Schrittantwortzeiten, 1271, 1284, 1296, 1308, 1324
 - CREATE_DB (Datenbaustein erstellen), 518
 - CSM 1277 Compact Switch Module, 1434
 - CTD (Rückwärtszählen), 225
 - CTRL_HSC (Schnellen Zähler steuern), 571
 - CTRL_HSC_EXT (Schnellen Zähler steuern (erweitert)), 544
 - CTS (Hardware-Flusskontrolle, PtP), 949
 - CTU (Vorwärtszählen), 225
 - CTUD (Vorwärts- und Rückwärtszählen), 225
 - cULus-Zulassung, 1254
- D**
- D_ACT_DP, 383
 - DataLogClear, 503
 - DataLogDelete, 505
 - Date
 - Datentyp Date, 109
 - DTL (Datentyp Date and Time long), 110
 - Datenaustausch zwischen E/A-Systemen, 744
 - Datenbaustein
 - Attribute mit ATTR_DB lesen, 525
 - CONF_DATA, 720
 - Einzelner FB mit Multiinstanz-DBs, 182
 - Fragmente in benutzerdefinierten Webseiten importieren, 908
 - Globaler Datenbaustein, 98, 182
 - Instanz-Datenbaustein, 98
 - Mit CREATE_DB erstellen, 518
 - Mit DELETE_DB löschen, 527
 - Optimierter Zugriff, 183
 - Organisationsbausteine (OBs), 178
 - READ_DBL (Aus Datenbaustein im Ladespeicher lesen), 522
 - Standardzugriff, 183
 - Struktur, 67
 - Synchronisieren der Startwerte von Online- und Offline-CPU's, 203
 - Übersicht, 67, 182
 - Werte erfassen und zurücksetzen, 1225
 - WRIT_DBL (In Datenbaustein im Ladespeicher schreiben), 522
 - Datenhantierungsbausteine (DHBs), 182
 - Datenprotokoll
 - Beispielprogramm, 514
 - DataLogClose (Datenprotokoll schließen), 504
 - DataLogCreate (Datenprotokoll erstellen), 495
 - DataLogNewFile (Datenprotokoll in neuer Datei), 507
 - DataLogOpen (Datenprotokoll öffnen), 499
 - DataLogWrite (Datenprotokoll schreiben), 501
 - Datenprotokolle anzeigen, 509
 - Datensatzstruktur, 494
 - Größenbegrenzung und Größenberechnung, 511
 - Mit DataLogClear leeren, 503
 - Mit DataLogDelete löschen, 505
 - Überblick über Datenprotokolle, 493
 - Datentypen, 105
 - Arrays, 112
 - Bool, Byte, Wort und Doppelwort, 107
 - PLC-Datentypeditor, 114
 - Real, LReal (Gleitpunkt), 108
 - Struc, 113
 - Time, Date, TOD (Uhrzeit), DTL (Date and Time long), 109
 - USInt, SInt, UInt, Int, UDInt, Dint (Ganzzahl), 108
 - Variant (Pointer), 114
 - Zeichen und Zeichenfolgen, 111
 - Datenübertragung, anstoßen, 977, 1125
 - Datum
 - SET_TIMEZONE (Zeitzone setzen), 333
 - T_ADD (Zeiten addieren), 328
 - T_COMBINE (Zeiten verknüpfen), 329
 - T_CONV (Zeiten umwandeln und extrahieren), 327
 - T_DIFF (Zeitdifferenz), 329
 - T_SUB (Zeiten subtrahieren), 328
 - DB (Datenbaustein), (Datenbaustein)

- DB_ANY_TO_VARIANT (DB_ANY in VARIANT konvertieren), 295
 - DB-Fragmente (benutzerdefinierte Webseiten)
 - Generieren, 912
 - Mit AWP-Befehl importieren, 908
 - Mit AWP-Befehlen erstellen, 907
 - DBs für benutzerdefinierte Webseiten erstellen, 912
 - DBs für benutzerdefinierte Webseiten generieren, 912
 - DC
 - Ausgänge, 1260
 - Erdung, 62
 - induktive Lasten, 64
 - Richtlinien für Potentialtrennung, 61
 - Verdrahtungsrichtlinien, 60, 62
 - DEC (Dekrementieren), 239
 - DECO (Decodieren), 320
 - DELETE (Zeichen in einer Zeichenkette löschen), 354
 - DELETE_DB (Datenbaustein löschen), 527
 - DEMUX (Demultiplexen), 323
 - Deserialize, 248
 - DETACH (Zuweisung eines OB zu einem Alarmereignis aufheben), 404
 - DeviceStates, Beispiel, 450
 - Diagnose
 - Anweisung DeviceStates (Modulstatus eines E/A-Systems lesen), 449
 - Anweisung Get_IM_Data (Identifikations- und Wartungsdaten lesen), 431
 - Anweisung ModuleStates (Modulstatusinformationen eines Moduls lesen), 454
 - Beobachtungstabelle, 1226
 - Diagnosepuffer, 1219
 - GET_DIAG (Diagnoseinformationen lesen), 459
 - LED (LED-Status lesen), 430
 - LED-Anzeigen, 1205
 - Puffer, 96
 - Speicherauslastung, 1219
 - Statusanzeige, 95
 - Zykluszeit, 1219
 - Diagnose, Reduzieren von Sicherheitsereignissen, 96
 - Diagnosefehler-OB, 80
 - Digitale E/A
 - Impulsabgriff, 170
 - Konfiguration, 170
 - Statusanzeigen, 1206
 - Digitale Signalboards
 - SB 1221, 1385
 - SB 1222, 1387
 - SB 1223, 1390, 1393
 - Digitale Signalmodule
 - SM 1221, 1330
 - SM 1222, 1332, 1333
 - SM 1223, 1339, 1345
 - DigitaleingangsfILTERzeit, 155
 - DIS_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen deaktivieren), 417
 - DIV (Dividieren), 236
 - Dokumentation, 4
 - DP-Normslaves
 - Alle Ausgänge mit SETIO schreiben, 377
 - Alle Eingänge mit GETIO lesen, 376
 - Teil der Ausgänge mit SETIO_PART schreiben, 379
 - Teil der Eingänge mit GETIO_PART lesen, 378
 - DPNRM_DG, 401
 - DPRD_DAT (Konsistente Daten eines DP-Normslaves lesen), 393
 - DPWR_DAT (Konsistente Daten eines DP-Normslaves schreiben), 393
 - Dynamische Verknüpfung, 168
- E**
- E/A
 - Adressierung, 103
 - Analoge Statusanzeigen, 1207, 1209
 - Darstellung Analogausgang (Spannung), 1359, 1401
 - Darstellung Analogausgang (Strom), 1359, 1402
 - Darstellung Analogeingang (Spannung), 1358, 1400
 - Darstellung Analogeingang (Strom), 1358, 1401
 - Digitale Statusanzeigen, 1206
 - Forcefunktion, 1230
 - In einer Beobachtungstabelle beobachten, 1226
 - induktive Lasten, 64
 - Schrittantwortzeiten (CPU), 1271, 1284, 1296, 1308, 1324
 - Schrittantwortzeiten (SB), 1400
 - Schrittantwortzeiten (SM), 1357
 - Zustand in KOP beobachten, 1224
 - E/A-System, Datenaustausch, 744
 - Eigenschaft Uhrzeitsynchronisation, 615
 - Ein- und Ausgänge
 - Beobachten, 1223
 - Einbau
 - Abstand, 44
 - CPU, 49
 - Erdung, 62
 - Erweiterungskabel, 58
 - induktive Lasten, 64

- Klemmenblock, 57
- Kommunikationsboard (CB), 52
- Kommunikationsmodul (CM), 56
- Kühlung, 44
- Lampenlasten, 63
- Luftströmung, 44
- Montageabmessungen, 47
- Richtlinien, 43
- Richtlinien für Potentialtrennung, 61
- Signalboard (SB), 52
- Signalmodul (SM), 54
- Signalmodule (SM), 32
- Thermischer Bereich, 44, 47
- Überblick, 48
- Übersicht, 43
- Verdrahtungsrichtlinien, 60, 62
- Voraussetzungen, 37
- Einfügen eines Geräts
 - Nicht spezifizierte CPU, 138
- Eingänge
 - Impulsabgriffbits, 156
- Eingangsfiterzeit, 155
- Eingangssimulatoren, 1430
- Eingeschränkte TSAPs und Portnummern, 728
- Einschaltverzögerung (TON), 217
- Einschränkungen
 - Benutzerdefinierte Webseiten, 916
 - Webserver, 937
- Elektromagnetische Verträglichkeit (EMV), 1257
- E-Mail, Senden mit TMAIL_C, 688
- EN und ENO (Signalfluss), 197
- EN_AIRT (Ausführung von Alarmen höherer Priorität und asynchronen Fehlerereignissen aktivieren), 417
- ENCO (Encodieren), 320
- Endebedingungen, 955
- ENDIS_PW (Passwort-Legitimierung einschränken und freigeben), 301
- Entwerfen einer Automatisierungslösung mit einem PLC-Gerät, 175, 176
- Enum-Typen definieren, benutzerdefinierte Webseiten, 905
- Enum-Typen in benutzerdefinierten Webseiten, 905
- Enum-Typen zuweisen, benutzerdefinierte Webseiten, 905
- EQ_ElemType (Vergleiche, ob Datentyp eines ARRAY-Elements GLEICH dem Datentyp einer Variablen ist), 233
- EQ_Type (Vergleiche, ob Datentyp GLEICH dem Datentyp einer Variablen ist), 233
- Ereignisausführung und Warteschlange, 86
- Erkennung zum Laden einer Online-CPU, 138
- Erreichbare Teilnehmer
 - Formatieren einer Memory Card, 1217
- Erreichbare Teilnehmer, Firmware aktualisieren, 1213
- Ersatz einer CPU V3.0 durch eine CPU V4.x, 1456
- Erweiterte Bausteinschnittstelle
 - Laden im Betriebszustand RUN, 1237
- Erweiterung der Leistungsmerkmale der S7-1200, 31
- Erweiterungskabel, 1433
 - Ausbau, 58
 - Einbau, 58
- Ethernet
 - Ad-hoc-Modus, 620
 - Alte Anweisung TRCV_C (Daten über Ethernet empfangen (TCP)), 650
 - Alte Anweisung TSEND_C (Daten über Ethernet senden (TCP)), 650
 - Alte Anweisungen TCON, TDISCON, TSEND und TRCV, 674
 - Anzahl der Kommunikationsverbindungen, 596
 - CSM 1277 Compact Switch Module, 1434
 - DPNRM_DG (Diagnosedaten eines DP-Slaves lesen), 401
 - DPRD_DAT (Konsistente Daten eines DP-Normslaves lesen), 393
 - DPWR_DAT (Konsistente Daten eines DP-Normslaves schreiben), 393
 - GET (Daten aus entfernter CPU auslesen), 799
 - IP-Adresse, 608
 - Kommunikationsarten, 589
 - MAC-Adresse, 608
 - Netzwerkverbindung, 600
 - PRVREC (Datensatz bereitstellen), 398
 - PUT (Daten in entfernte CPU schreiben), 799
 - RALRM (Alarm empfangen), 380
 - RCVREC (Datensatz empfangen), 396
 - RDREC (Datensatz lesen), 373
 - T_CONFIG (Schnittstelle konfigurieren), 717
 - TCON, 657
 - TDISCON, 657
 - TRCV, 657
 - TRCV_C, 636
 - TSEND, 657
 - TSEND_C, 636
 - TURCV (Daten über Ethernet empfangen (UDP)), 712
 - TUSEND (Daten über Ethernet senden (UDP)), 712
 - Übersicht, 617
 - Verbindungs-IDs, 621
 - WRREC (Datensatz schreiben), 373
- Ethernet-Protokolle, 617
 - Verbindungen mit mehreren Teilnehmern, 803
- EXIT, SCL, 317

EXP (Exponentialwert bilden), 242
 EXPT (Potenzieren), 242

F

F_TRIG (Variable bei negativer Signalflanke setzen), 216
 FAQs, 4
 FB (Funktionsbaustein)
 Übersicht, 67
 FC (Funktion), 67, 180
 Fehler
 Diagnosefehler, 80
 Häufige Fehler bei erweiterten Anweisungen, 535
 Zeitfehler, 79
 Fehler der Empfangskonfiguration, 974, 1125
 Fehler der Sendekonfiguration, 969, 1120
 Fehler durch inkompatible CPU-Version, 1206
 Fehler durch unbekannte CPU-Version, 1206
 Fehlerbehebung
 Diagnosepuffer, 1219
 LED-Anzeigen, 1205
 FeldLesen (Feld lesen), 265
 FeldSchreiben (Feld schreiben), 265
 Feste Länge, 957
 FileDelete, 540
 FileReadC, 535
 FileWriteC, 538
 FILL_BLK (Speicher mit Bitmuster belegen), 253
 Filterzeit, 155
 FIND (Zeichen in einer Zeichenkette finden), 357
 Firmware aktualisieren
 aus STEP 7, 1212
 mit einer Memory Card, 132
 Firmware-Aktualisierung
 über Webserver, 874
 Firmware-Update
 aus STEP 7, 1212
 mit einer Memory Card, 132
 Flankenoperationen, steigend und fallend, 214
 Flexible Maschinen (Konfigurationssteuerung), 141
 FLOOR (Aus Gleitpunktzahl nächstniedere Ganzzahl erzeugen), 290
 Flusskontrolle, 948
 Konfiguration, 948
 Verwalten, 949
 FM-Zertifizierung, 1254
 FOR, SCL, 313
 Forcen, 1230
 Beobachtungstabelle, 1226
 Eingänge der Peripherie, 1230
 Eingänge und Ausgänge, 1230

Speicherbereich E, 1230
 Zyklus, 1230

Forcetabelle

Eingänge der Peripherie ansprechen, 1230
 Forcefunktion, 1230
 Forcen, 1230

Formatieren einer Memory Card, 1217

FRAC (Nachkommastellen ermitteln), 242

Frequenz, Taktbits, 95

Funktion (FC)

Codebausteine im Anwenderprogramm aufrufen, 178
 Gültige FC-Nummern, 67
 Knowhow-Schutz, 167
 Lineare und strukturierte Programme, 176
 Übersicht, 67, 180

Funktion, I-Device, 740

Funktionen, neu, 35

Funktionsbaustein (FB)

Anfangswert, 181
 Ausgangsparameter, 181
 Codebausteine im Anwenderprogramm aufrufen, 178
 Einzelner FB mit Multiinstanz-DBs, 182
 Gültige FB-Nummern, 67
 Instanz-Datenbaustein, 181
 Knowhow-Schutz, 167
 Lineare und strukturierte Programme, 176
 Übersicht, 67, 181

Funkverbindung zum Webserver, 856

FUP (Funktionsplan), 190

Für den Webserver unterstützte Browser, 849

G

Gen_UsrMsg (Anwenderdiagnosemeldungen erzeugen), 418

GEOADDR, 534

Gerät

PROFINET IO, 737

PROFINET IO-Device-Namen, 738

Shared, 752

Gerät ändern, 152

Geräteaustausch

CPU V3.0 durch CPU V4.x, 1456

Vorgehensweise, 152

Gerätekonfiguration, 135, 729

Ändern eines Gerätetyps, 152

ASi, 791

AS-i-Port, 791

CPU konfigurieren, 153

Eigenschaft Uhrzeitsynchronisation (PROFINET), 615
 Erkennen, 138
 Ethernet-Port, 608
 Laden in CPU, 199
 Module hinzufügen, 140
 Module konfigurieren, 169
 Netzwerkverbindung, 600
 Neues Gerät hinzufügen, 137
 PROFIBUS, 786
 PROFINET-Port, 608
 GET (Daten aus entfernter CPU auslesen), 799
 Verbindung konfigurieren, 602
 GET_DIAG (Diagnoseinformationen lesen), 459
 GET_ERROR (Fehler lokal abrufen), 305
 GET_ERROR_ID (Fehler-ID lokal abrufen), 307
 Get_Features (Erweiterte Funktionen abrufen), 985
 GetBlockName (Name des Bausteins auslesen), 369
 GetInstanceName (Namen der Baustein-Instanz auslesen), 364
 GetInstancePath (Zusammengesetzten globalen Namen der Baustein-Instanz abfragen), 367
 GETIO, 376
 GETIO_PART, 378
 GETSMCInfo, 466
 GetSMCInfo (Informationen über die Memory Card auslesen), 466
 GetStationInfo, 439
 GetSymbolName (Namen einer Variable am Eingangsparameter auslesen), 358
 GetSymbolPath (Zusammengesetzten globalen Namen der Eingangsparameterversorgung abfragen), 361
 Globale Bibliothek
 Überblick über das alte USS-Protokoll, 1132
 Überblick über das USS-Protokoll, 998
 Globaler Datenbaustein, 98, 182
 GOTO, SCL, 317
 Grundlegende Panels (HMI), 32
 GSD-Datei, 749

H

Handbücher, 4
 Hardware-Flusskontrolle, 949
 Hardwarekonfiguration, 135
 ASi, 791
 AS-i-Port, 791
 CPU konfigurieren, 153
 Erkennen, 138
 Ethernet-Port, 608
 Laden in CPU, 199

Module hinzufügen, 140
 Module konfigurieren, 169
 Netzwerkverbindung, 600
 Neues Gerät hinzufügen, 137
 PROFIBUS, 786
 PROFINET-Port, 608
 Haupteingabe, 845
 Herunterladen
 Firmware-Update, 132
 Siemens-Sicherheitszertifikat auf PC, 867
 HMI-Geräte
 Netzwerkverbindung, 600
 PROFINET-Kommunikation konfigurieren, 731
 Übersicht, 32
 Hochrüsten einer CPU V3.0 auf V4.x, 1456
 Hotline, 3
 HSC (schneller Zähler)
 Betriebsphase, 562
 Konfiguration, 560
 Zählarten, 561
 HTA (Hexadezimalzahl in ASCII-Zeichenkette umwandeln), 348
 HTML-Seiten
 Anwenderdefiniert, 895
 Liste, Beispiel für benutzerdefinierte Webseiten, 922
 HTML-Seiten, benutzerdefiniert
 Aktualisieren, 896
 Entwickeln, 895
 Seitenanordnung, 911
 Speicherorte für Sprachen, 931
 Zugriff auf S7-1200 Daten, 896
 HTTP-Variablen lesen, 901
 HTTP-Verbindungen, Webserver, 937

I

Identifikation der CPU, Anzeige mit Webserver, 868
 I-Device (Intelligent IO Device)
 Funktionalität, 740
 Konfigurieren, 747
 Konfigurieren mit GSD-Datei, 749
 Shared, 757
 I-Device (intelligentes IO-Device)
 Eigenschaften, 741
 I-Device (intelligentes IO-Gerät)
 untergeordnetes PN IO-System, 741
 IF-THEN, SCL, 311
 Impuls (TP), 217
 Impulsabgriff, 156, 170
 Impulsabgriffbits, Konfiguration der
 Digitaleingänge, 156

Impulsausgänge, 475
 IN_Range (Wert innerhalb Bereich), 231
 INC (Inkrementieren), 239
 Induktive Lasten, 64
 Industrieumgebungen
 Zulassungen, 1257
 Informationsquellen, 4
 INSERT (Zeichen in einer Zeichenkette einfügen), 355
 Installation
 Leistungsbilanz, 45
 Instanz-Datenbaustein, 98
 INV (Einerkomplement erstellen), 319
 IO2MOD (Aus einer E/A-Adresse die
 Hardwarekennung ermitteln), 531
 IO-Link
 Adressbereich, 1379
 Anschlussbelegung, 1377
 Austausch, 1376
 Datensatz, 1380
 Diagnose, 1383
 Diagramm, 1378
 Fehlermeldungen, 1380, 1382, 1383
 Funktionen, 1376
 Gerätelagerung, 1377
 Geräteprofil, 1375
 Konfigurieren, 1379
 LED-Anzeige, 1382
 Parameter, 1379
 Parameter während der Laufzeit ändern, 1380
 Rücksetzen auf Werkseinstellungen, 1376
 IO-Link-Master-Signalmodul, 1372
 IP-Adresse, 608, 609
 CPU online konfigurieren, 1212
 Gerätekonfiguration, 153
 Konfigurieren, 608
 MAC-Adresse, 608
 Online zuweisen, 606
 Zuordnen, 604, 612
 IP-Adresse des Routers, 609
 IP-Adresse, Notadresse (temporär), 812
 IP-Router, 608
 IS_ARRAY (Auf ARRAY prüfen), 234
 IS_NULL (Abfrage nach Pointer GLEICH Null), 234
 ISO on TCP
 Ad-hoc-Modus, 620
 ISO-on-TCP
 Parameter, 624
 Verbindungs-IDs, 621
 Verbindungskonfiguration, 601
 ISO-on-TCP-Protokoll, 617

J

JavaScript, Standard-Webseiten, 938
 JMP (Springen bei VKE = 1), 296
 JMP_LIST (Sprungliste definieren), 297
 JMPN (Springen bei VKE = 0), 296

K

Kabel
 Erweiterung, 1433
 Netzwerkcommunication, 942
 Kalender, 327
 Kein Neustart, 71
 Klemmenblock, 57, 1446
 Klemmenblock, Einbau und Ausbau, 57
 Knowhow-Schutz
 Passwortschutz, 167
 Knowhow-Schutz, Anzeige mit Webserver, 868
 Kommunikation
 Abfragearchitektur, 988
 Aktiv/passiv, 601, 624, 805
 Anzahl der Verbindungen (PROFINET/
 PROFIBUS), 596
 ASi-Adresse, 791
 Eigenschaft Uhrzeitsynchronisation
 (PROFINET), 615
 Flusskontrolle, 949
 Hardwareanschluss, 729
 IP-Adresse, 608
 Kommunikationslast, 91
 Konfiguration, 601, 624, 805
 MAC-Adresse, 608
 Netzwerk, 728
 Netzwerkverbindung, 600
 PROFIBUS-Adresse, 786
 PROFINET und PROFIBUS, 589
 Protokolle, 619
 Sende- und Empfangsparameter, 950
 TCON_Param, 624
 Verbindungs-IDs, 621
 Verlust, Ziehen oder Stecken von Modulen, 82
 Zykluszeit, 91
 Kommunikationsboard (CB)
 Ausbau, 52
 CB 1241 RS485, 1425
 Einbau, 52
 Gerätekonfiguration, 135
 Konfiguration von Parametern, 169
 LED-Anzeigen, 941, 1205

- Module hinzufügen, 140
- Programmieren, 987
- RS485, 941
- Übersicht, 31
- Vergleichstabelle, 31
- Kommunikationsmodul (CM)
 - AS-i-Mastermodul CM1243-2 hinzufügen, 789
 - Ausbau, 56
 - CM 1241 RS232, 1427
 - CM 1241 RS422/RS485, 1428
 - CM 1243-5 (DP-Master) hinzufügen, 785
 - Datenempfang, 980, 1128
 - Einbau, 56
 - Gerätekonfiguration, 135
 - Konfiguration des PtP-Beispielprogramms, 990
 - Konfiguration von Parametern, 169
 - LED-Anzeigen, 941, 1205
 - Leistungsbedarf, 1437
 - Module hinzufügen, 140
 - Programmieren, 987
 - RS232 und RS485, 941
 - Übersicht, 32
 - Vergleichstabelle, 31
- Kommunikationsprozessor (CP)
 - Gerätekonfiguration, 135
 - Konfiguration von Parametern, 169
 - Module hinzufügen, 140
 - Übersicht, 32
 - Vergleichstabelle, 31
- Kommunikationsschnittstellen
 - CB 1241 RS485, 1425
 - CM 1241 RS232, 1427
 - Gerätekonfiguration, 135
 - Konfiguration, 947
 - LED-Anzeigen, 1205
 - Module hinzufügen, 140
 - Programmieren, 987
 - RS232 und RS485, 941
 - Vergleichstabelle der Module, 31
- Kommunikationsschnittstellen, 3964(R), 960
- Kompatibilität, 40
- Kompensation der kalten Verbindungsstelle, Thermoelement, 1363, 1405
- Konfiguration
 - Anlaufparameter, 121
 - ASi, 791
 - AS-i-Port, 790
 - CPU-Parameter, 153
 - Eigenschaft Uhrzeitsynchronisation (PROFINET), 615
 - Erkennen, 138
 - Ethernet-Port, 608
 - HSC (schneller Zähler), 560
 - IP-Adresse, 608
 - Kommunikation PLC/PLC, 732
 - Kommunikationslast, 91
 - Kommunikationsschnittstellen, 947
 - Laden in CPU, 199
 - MAC-Adresse, 608
 - Meldung empfangen, 952
 - Module, 169
 - Module hinzufügen, 140
 - Netzwerkverbindung, 600
 - PROFIBUS, 786
 - PROFIBUS-Adresse, 786
 - PROFINET-Port, 608
 - RS422, Betriebsarten, 993
 - RS485, Betriebsarten, 995
 - Schnittstellen, 947
 - Zykluszeit, 90
- Konfiguration der gesendeten Meldung, 950
- Konfiguration Empfangsparameter, 736
- Konfiguration für Meldungssendung, 950
- Konfiguration Sendeparameter, 601, 735, 805
- Konfiguration, 3964(R)
 - Kommunikationsschnittstellen, 960
 - Priorität und Protokollparameter, 961
 - Schnittstellen, 960
- Konfiguration, benutzerdefinierte Webseiten
 - Mehrere Sprachen einrichten, 931
 - STEP 7-Konfiguration, 911
- Konfigurationssteuerung (Optionenhandling)
 - Steuerdatensatz, 145
- Konfigurationssteuerung (Optionsverwaltung), 141
 - Beispiel, 148
- Konsistenzprüfung, 208
- Kontaktdaten, 3, 152
- Kontakte, 209
- Konventionen bei Anführungszeichen, Webserver, 909
- KOP (Kontaktplan)
 - Beobachten, 1224
 - Programmiereditor, 1224
 - Übersicht, 189
 - Überwachungsstatus oder -wert, 1223
 - Zustand, 1224, 1230
- Kopieren, Ausschneiden und Einfügen in STEP 7, 40
- Kopierschutz
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Kühlen, 44
- Kunden-Support, 3

L

Label (Sprungmarke), 296
 Laden aus CPU
 Anwenderprogramm, 205
 Bausteine aus einer Online-CPU kopieren, 205
 Laden im Betriebszustand RUN
 aus STEP 7, 1234
 Einschränkungen, 1239
 Erweiterte Bausteinschnittstelle, 1237
 Globale Einstellungen für die Speicherreserve, 1239
 Laden ausgewählter Bausteine, 1235
 Laden fehlgeschlagen, 1240
 Laden ohne Reinitialisierung, 1237
 Sicherheitsaspekte, 1240
 Speicherreserve und remanente Speicherreserve, 1237
 Übersetzungsfehler, 1236
 Übersicht, 1232
 Voraussetzungen, 1233
 Laden in CPU
 Anwenderprogramm, 199
 DBs für benutzerdefinierte Webseiten, 915
 MAC- und IP-Adressen anzeigen, 613
 Projekt, 199
 Ladespeicher, 28
 Benutzerdefinierte Webseiten, 916
 CPU 1211C, 1263
 CPU 1212C, 1276
 CPU 1214C, 1288
 CPU 1215C, 1300
 CPU 1217C, 1314
 Lampenlasten, 63
 Länge PtP-Meldung, 958
 Latenz, 86
 Laufzeitfehler beim Senden, 978, 1127
 Lebensdauer eines Relais, 1261
 Lebensdauer Memory Card, 128
 LED (LED-Status lesen), 430
 LED-Anzeigen
 CPU-Status, 1205
 Kommunikationsschnittstelle, 941, 1205
 Leerzeile, 953
 LEFT (linke Zeichen einer Zeichenkette lesen), 353
 Leistungsbilanz, 45
 Beispiel, 1438
 Formular für Berechnungen, 1439
 Übersicht, 1437
 Leistungszeiten, 1265, 1277, 1289, 1302, 1316
 Leitung im Leerlauf, 951

LEN (Länge einer Zeichenkette ermitteln), 351
 Lesen aus DBs, E/A oder Speicher, 195, 258
 LIMIT (Grenzwert setzen), 241
 Lineare Programmierung, 176
 Lizenzen, OPC UA, 1453
 LN (Natürlichen Logarithmus bilden), 242
 LOG2GEO (Steckplatz über Hardwarekennung ermitteln), 530
 Logic Analyzer, 1242
 Lokaldaten
 Maximum je OB-Prioritätsstufe, 102
 Verwendung durch Bausteine, 102
 LOWER_BOUND (Untere ARRAY-Grenze auslesen), 255
 Luftströmung, 44

M

MAC- und IP-Adressen anzeigen, 613
 MAC-Adresse, 608, 613
 Manuelle Steuerung von DB-Fragmenten, 932
 MAX (Maximum abrufen), 240
 MAX_LEN (Maximale Länge einer Zeichenkette), 351
 Maximale Anzahl Webserver-Verbindungen, 937
 Maximale Meldungslänge, 957
 MB_CLIENT, 1022
 MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren), veraltet, 1148
 MB_COMM_LOAD (Port am PtP-Modul für Modbus RTU konfigurieren), veraltet, 1166
 MB_MASTER (Über PtP-Port als Modbus-Master kommunizieren), veraltet, 1169
 MB_RED_CLIENT, 1044
 MB_RED_SERVER, 1063
 MB_SERVER, 1033
 MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren), veraltet, 1155
 MB_SLAVE (Über den PtP-Port als Modbus-Slave kommunizieren), veraltet, 1175
 MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern), 582
 MC_CommandTable, 581
 MC_Halt (Achse pausieren), 578
 MC_Home (Referenzpunktfahrt der Achse durchführen), 578
 MC_MoveAbsolute (Achse absolut positionieren), 579
 MC_MoveJog (Achse im Tippbetrieb bewegen), 580
 MC_MoveRelative (Achse relativ positionieren), 579
 MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 580
 MC_Power (Achse freigeben/sperrern), 577

- MC_ReadParam (Parameter eines Technologieobjekts lesen), 582
- MC_Reset (Fehler bestätigen), 577
- MC_WriteParam (Parameter des Technologieobjekts schreiben), 581
- Medienredundanz
 - Funktionsweise in einer Ringtopologie, 769
 - Konfigurieren, 773
- Mehrere AWP-Variablendefinitionen, 909
- Meldung
 - Ende, 955
 - Länge, 957
 - Starten, 953
- Meldungsempfang konfigurieren
 - PtP-Beispielprogramm, 991
 - PtP-Gerät, Konfiguration, 952
- Meldungskonfiguration
 - Anweisungen, 988
 - Empfangen, 952
 - Senden, 950
- Meldungssendung konfigurieren
 - PtP-Beispielprogramm, 990
 - PtP-Gerät, Konfiguration, 950
- Memory Card, 1430
 - Firmware-Update, 132
 - In CPU einfügen, 118
 - Informationen lesen, 466
 - Inkompatibilitätsfehler, 1206
 - Leere Übertragungskarte für ein verlorenes Passwort, 134
 - Programmkarte, 126
 - Schutz vertraulicher PLC-Konfigurationsdaten, 129
 - Übersicht, 118
 - Übertragungskarte, 122
- Memory Card in CPU stecken, 118
- Messungen, Trace-Aufträge, 1243
- MicroMaster-Antrieb, verbinden, 1014
- MID (mittlere Zeichen einer Zeichenkette lesen), 353
- MIN (Minimum abrufen), 240
- Mobiles Gerät, Zugriff auf Webserver, 856
- Mobilgeräte
 - Webseitenaufbau, 862
- MOD (Divisionsrest einer Division), 237
- Modbus
 - Adressen im Speicher, 1019
 - Funktionscodes, 1017
 - MB_CLIENT (Über PROFINET als Modbus TCP-Client kommunizieren), veraltet, 1148
 - MB_COMM_LOAD (Port am PtP-Modul für Modbus RTU konfigurieren), veraltet, 1166
 - MB_MASTER (Über PtP-Port als Modbus-Master kommunizieren), veraltet, 1169
 - MB_SERVER (Über PROFINET als Modbus TCP-Server kommunizieren), veraltet, 1155
 - MB_SLAVE (Über den PtP-Port als Modbus-Slave kommunizieren), veraltet, 1175
 - Modbus_Comm_Load (SIPLUS I/O oder Port am PtP-Modul für Modbus RTU konfigurieren), 1092
 - Modbus_Master (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Master kommunizieren), 1096
 - Modbus_Slave (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Slave kommunizieren), 1103
 - RTU-Kommunikation, 1019
 - Stationsadressen im Netzwerk, 1019
 - Versionen, 1000, 1091, 1133, 1165
- MODBUS
 - MB_CLIENT, 1022
 - MB_RED_CLIENT, 1044
 - MB_RED_SERVER, 1063
 - MB_SERVER, 1033
- Modbus RTU
 - Beispiel Slave, 1116
 - Master-Programm, 1113
- Modbus TCP
 - Versionen, 1021, 1147
- Modbus_Master (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Master kommunizieren), 1096
- Modbus_Slave (Über SIPLUS I/O oder den PtP-Port als Modbus RTU-Slave kommunizieren), 1103
- Module
 - Kommunikationsboards (CB), 31
 - Kommunikationsmodul (CM), 32
 - Kommunikationsprozessor (CP), 32
 - Parameter konfigurieren, 169
 - Signalboard (SB), 31
 - Signalmodule (SM), 32
 - Thermischer Bereich, 44, 47
 - Vergleichstabelle, 31
- ModuleStates, 454
- ModuleStates Beispiel, 455
- Momentaufnahme von DB-Werten, 1225
- Montage
 - Abmessungen, 47
 - Abstand, 44
 - CPU, 49
 - Erdung, 62
 - Erweiterungskabel, 58
 - induktive Lasten, 64
 - Klemmenblock, 57
 - Kommunikationsboard (CB), 52
 - Kommunikationsmodul (CM), 56
 - Kühlung, 44

- Lampenlasten, 63
- Luftströmung, 44
- Richtlinien, 43
- Signalboard (SB), 52
- Signalmodul (SM), 54
- Thermischer Bereich, 44, 47
- Trennung, 61
- Übersicht, 48
- Verdrahtungsrichtlinien, 60, 62
- MOVE (Wert kopieren), 244
- MOVE_BLK (Bereich kopieren), 244
- MUL (Multiplizieren), 236
- MUX (Multiplexen), 322

N

- N (Operand auf negative Signalflanke abfragen), 214
- N_TRIG (VKE auf negative Signalflanke abfragen), 215
- N=Box und N-Spule (Operand bei negativer Signalflanke setzen), 215
- NE_ElemType (Vergleiche, ob Datentyp eines ARRAY-Elements UNGLEICH dem Datentyp einer Variablen ist), 233
- NE_Type (Vergleiche, ob Datentyp UNGLEICH dem Datentyp einer Variablen ist), 233
- NEG (Zweierkomplement erstellen), 238
- Network Time Protocol (NTP), 615
- Netzwerkcommunication, 728
 - Kabelabschluss, 942
- Netzwerkverbindung
 - Geräte verbinden, 600
 - Mehrere CPUs, 732, 733, 737, 785, 790
- Netzwerkverbindung erstellen
 - Zwischen PLC-Geräten, 600
- Neue Funktionen, 35
- Neues Gerät hinzufügen
 - CPU, 137
 - Nicht spezifizierte CPU, 138
 - Vorhandene Hardware erkennen, 138
- Nicht spezifizierte CPU, 138
- NORM_X (Normieren), 291
- NOT (VKE invertieren), 211
- NOT_NULL (Abfrage nach Pointer UNGLEICH Null), 234
- NOT_OK (Ungültigkeit prüfen), 232

O

- OB, (Organisationsbaustein)
- OB "Ziehen oder Stecken von Modulen", 82

- OB MC-PostServo, 86
- OB MC-PreServo, 85
- ODER (logische Verknüpfung), 318
- Offene Benutzerkommunikation
 - Herstellen einer Verbindung und Lesen von Daten mit der alten Anweisung TRCV_C, 650
 - Herstellen einer Verbindung und Lesen von Daten mit TRCV_C, 636
 - Herstellen einer Verbindung und Senden von Daten mit der alten Anweisung TSEND_C, 650
 - Herstellen einer Verbindung und Senden von Daten mit TSEND_C, 636
- OK (Gültigkeit prüfen), 232
- Online
 - Bedienpanel, 1218
 - Beobachtungstabelle, 1223, 1224, 1226
 - DB-Werte erfassen und zurücksetzen, 1225
 - Diagnosepuffer, 1219
 - Forcefunktion, 1230
 - Forcen, 1230
 - IP-Adresse, 1212
 - IP-Adresse zuordnen, 606
 - Online gehen, 1209
 - Speicherauslastung, 1219
 - Tageszeit, 1212
 - Tools, 1223
 - Überwachungsstatus oder -wert, 1223
 - Vergleichen und synchronisieren, 1221
 - Zustand, 1224
 - Zykluszeit, 1219
- Online- und Diagnose-Tools
 - Laden im Betriebszustand RUN, 1232
- Online- und Offline-CPU's vergleichen und synchronisieren, 1221
- Online verbinden, 1209
- Online-Gerätenamen
 - PROFINET IO, 1210
- OPC UA
 - Methodenaufrufe, 841
- OPC UA-Lizenzen, 1453
- OPC UA-Methodenaufrufe, 841
- OPC UA-Server
 - Aktivieren, 814
 - Begleitspezifikationen, 827
 - Benutzer-Authentifizierung, 822
 - Konfigurationseinstellungen, 817
 - PLC-Teilnehmer, 825
 - Schnittstellen, 827
 - Serverschnittstelle definieren, 824
 - Überblick Sicherheit, 820
 - unterstützte Datentypen, 824
 - unterstützte Sicherheitsrichtlinien, 820

- unterstützte Teilnehmertypen, 825
- vertrauenswürdige Clients, 822
- OPC, Projektierung, 1194
- Optimierte Datenbausteine, 183
- Optionsverwaltung (Konfigurationssteuerung), 141
- Ordner, Sprachen für benutzerdefinierte
- Webseiten, 928
- Organisationsbaustein
 - Anlaufverarbeitung, 74
 - Aufrufen, 75
 - Bearbeiten, 178
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - Erstellen, 179
 - Funktion, 75
 - Funktionsweise konfigurieren, 180
 - Knowhow-Schutz, 167
 - Lineare und strukturierte Programmierung, 176
 - Mehrere Zyklus-OBs, 180
 - Prioritätsklassen, 75
 - Übersicht, 67
 - Weckalarm, 77
 - Zuordnung zum temporären Speicher, 102
- Organisationsbaustein (OB)
 - Startinformation mit RD_SINFO lesen, 421
- Ortszeit
 - RD_LOC_T (Lokalzeit lesen), 330
 - WR_LOC_T (Lokalzeit einstellen), 330
- OUT_Range (Wert außerhalb Bereich), 231

P

- P (Operand auf positive Signalflanke abfragen), 214
- P_TRIG (VKE auf positive Signalflanke abfragen), 215
- P=Box und P-Spule (Operand bei positiver Signalflanke setzen), 215
- P3964_Config (3964(R)-Protokoll konfigurieren), 975
 - Error, 976
- Panels (HMI), 32
- Parameter BUFFER, SEND_P2P, 979
- Parameter LENGTH, SEND_P2P, 979
- Parameterkonfiguration
 - Empfangen, 736
 - LENGH und BUFFER für SEND_P2P, 979
 - Senden, 735
- Parametrieren, 181
- Parität, 948
- Passive/aktive Kommunikation
 - Parameter, 624
 - Partner konfigurieren, 601, 805
 - Verbindungs-IDs, 621

- Passwortschutz
 - Codebaustein, 167
 - ENDIS_PW (Passwort-Legitimierung einschränken und freigeben), 301
 - Kopierschutz, 168
 - Leere Übertragungskarte, 134
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
 - Verlorenes Passwort, 134
 - Zugriff auf die CPU, 163
- Pause, 951, 953
- PEEK, PEEK_WORD, PEEK_BOOL, PEEK_DWORD, PEEK_BLK, 195, 258
- Phasenverschiebung, Weckalarm-OBs, 77
- PID
 - PID_3STEP (PID-Regler mit Einstellung für Ventile), 586
 - PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 585
 - PID_Temp (universeller PID-Regler für die Temperaturregelung), 587
- PID-Ventileinstellung, 586
- PLC
 - Anlaufverarbeitung, 74
 - Bausteine aus einer Online-CPU kopieren, 205
 - Beobachten, 1223
 - Beobachtungstabelle, 1226
 - Betriebszustände, 71
 - Eigenschaft Uhrzeitsynchronisation, 615
 - Einbau, 48, 49
 - Entwerfen einer Automatisierungslösung, 175
 - Erweiterungskabel, 58
 - Forcefunktion, 1230
 - Forcen, 1230
 - Gerätekonfiguration, 135
 - HSC-Konfiguration, 560
 - Klemmenblock, 57
 - Knowhow-Schutz, 167
 - Kommunikationslast, 91
 - Laden in CPU, 199
 - Leistungsbilanz, 45
 - Module hinzufügen, 140
 - RTM (Betriebsstundenzähler), 334
 - Übersicht über die CPU, 27
 - Variablen, 98
 - Vergleichen und synchronisieren, 1221
 - Zuweisen einer IP-Adresse zu einer Online-CPU, 606
 - Zykluszeit, 90, 91
- PLC-Gerät
 - Bausteine verwenden, 176
- PM 1207 Stromversorgungsmodul, 1434

- PN-Slave
 - Aktivieren und Deaktivieren mit D_ACT_DP, 383
- Pointer
 - Datentyp Variant, 114
- POKE, POKE_BOOL, POKE_BLK, 195, 258
- PORT_CFG (Kommunikationsparameter dynamisch konfigurieren), veraltet, 1117
- Port_Config (Portkonfiguration), 965
- Portalansicht, 39
- Portkonfiguration, 947
 - Error, 967, 1119
 - PtP-Beispielprogramm, 990
- Portkonfiguration, 3964(R), 960
- Portnummern
 - eingeschränkt, 728
 - Zu Kommunikationspartnern zuweisen, 617
- Potentiometermodul
 - Technische Daten, 1432
- Priorität
 - Priorität bei der Verarbeitung, 86
 - Prioritätsklassen, 75
- PROFIBUS
 - Adresse, 786
 - Adresse konfigurieren, 786
 - Anweisungen für die dezentrale Peripherie, 371
 - Anzahl der Kommunikationsverbindungen, 596
 - CM 1242-5 (DP-Slave), 782
 - CM 1243-5 (DP-Master), 782
 - CM 1243-5 (DP-Master) hinzufügen, 785
 - DPNRM_DG (Diagnosedaten eines DP-Slaves lesen), 401
 - DPRD_DAT (Konsistente Daten eines DP-Normslaves lesen), 393
 - DP-Slave hinzufügen, 785
 - DPWR_DAT (Konsistente Daten eines DP-Normslaves schreiben), 393
 - GET (Daten aus entfernter CPU auslesen), 799
 - Master, 782
 - Netzwerkverbindung, 600, 785
 - PUT (Daten in entfernte CPU schreiben), 799
 - RALRM (Alarm empfangen), 380
 - RDREC (Datensatz lesen), 373
 - S7-Verbindung, 803
 - Slave, 782
 - WRREC (Datensatz schreiben), 373
- PROFIBUS und PROFINET
 - Beispiel DeviceStates, 450
 - ModuleStates Beispiel, 455
- PROFIenergy, 403
- Profil-OB, 84
- PROFINET
 - Ad-hoc-Modus, 620
 - Anweisungen für die dezentrale Peripherie, 371
 - Anzahl der Kommunikationsverbindungen, 596
 - CPU-zu-CPU-Kommunikation, 732
 - DPRD_DAT (Konsistente Daten eines DP-Normslaves lesen), 393
 - DPWR_DAT (Konsistente Daten eines DP-Normslaves schreiben), 393
 - Eigenschaft Uhrzeitsynchronisation, 615
 - Eigenschaften der Ethernet-Adresse, 609
 - Gerätenamen und Adressierung, 617
 - GET (Daten aus entfernter CPU auslesen), 799
 - IP-Adresse, 608
 - IP-Adresse konfigurieren, 153
 - Kommunikation PLC/PLC, 732
 - Kommunikation zwischen CPU und HMI-Gerät konfigurieren, 731
 - Kommunikationsarten, 589
 - MAC-Adresse, 608
 - Netzwerkverbindung, 600, 732, 733, 737
 - PRVREC (Datensatz bereitstellen), 398
 - PUT (Daten in entfernte CPU schreiben), 799
 - RALRM (Alarm empfangen), 380
 - RCVREC (Datensatz empfangen), 396
 - RDREC (Datensatz lesen), 373
 - S7-Verbindung, 803
 - Startzeit des Systems, 616
 - Testen eines Netzwerks, 612
 - Übersicht, 617
 - Uhrzeitsynchronisation, 153
 - Verbindung beenden und neu aufbauen, 682
 - Verbindungs-IDs, 621
 - WRREC (Datensatz schreiben), 373
 - Zuweisung der IP-Adresse, 617
- PROFINET IO
 - CPU zuweisen, 738
 - Geräte, 737
 - Gerätenamen, 738
 - Gerätenamen online zuweisen, 1210
 - Gerätenamen zuweisen, 738
 - Hinzufügen eines Geräts, 737
 - Online-Gerätenamen, 1210
- PROFINET IO-Devices
 - Alle Ausgänge mit SETIO schreiben, 377
 - Teil der Ausgänge mit SETIO_PART schreiben, 379
 - Teil der Eingänge mit GETIO_PART lesen, 378
- PROFINET RT, 617
- PROFINET-Anweisungen
 - Alte Anweisung TRCV_C (Daten über Ethernet empfangen (TCP)), 650
 - Alte Anweisung TSEND_C (Daten über Ethernet senden (TCP)), 650

- Alte Anweisungen TCON, TDISCON, TSEND und TRCV, 674
- T_CONFIG (Schnittstelle konfigurieren), 717
- T_DIAG, 684
- T_RESET, 682
- TCON, 657
- TDISCON, 657
- TRCV, 657
- TRCV_C, 636, 735
- TSEND, 657
- TSEND_C, 636
- TURCV (Daten über Ethernet empfangen (UDP)), 712
- TUSEND (Daten über Ethernet senden (UDP)), 712
- PROFINET-Port
 - Autonegotiation, 610
- Programm
 - Bausteine aus einer Online-CPU kopieren, 205
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - Laden in CPU, 199
 - Lineare und strukturierte Programme, 176
 - Memory Card, 118
 - Organisationsbausteine (OBs), 178
 - Passwortschutz, 167
 - Prioritätsklassen, 75
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Programm beobachten, 206
- Programm testen, 206
- Programmausführung, 67
- Programmiereditor
 - Beobachten, 1224
 - Zustand, 1224
- Programmierung
 - Anfangswert eines FB, 181
 - Anzahl der Codebausteine, 67
 - Bausteinaufrufe, 67
 - Betriebszustände der CPU, 71
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - Codebausteine vergleichen und synchronisieren, 1221
 - Datenbaustein (DB), 67
 - Funktion (FC), 180
 - Funktionsbaustein (FB), 67, 181
 - FUP (Funktionsplan), 190
 - Gültige FC-, FB- und DB-Nummern, 67
 - Instanz-Datenbaustein (DB), 181
 - KOP (Kontaktplan), 189
 - Lineares Programm, 176
 - Nicht spezifizierte CPU, 138
 - PID_3STEP (PID-Regler mit Einstellung für Ventile), 586
 - PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 585
 - PID_Temp (universeller PID-Regler für die Temperaturregelung), 587
 - Prioritätsklassen, 75
 - PtP-Anweisungen, 987
 - RTM (Betriebsstundenzähler), 334
 - SCL (Structured Control Language), 190, 192
 - Signalfluss (EN und ENO), 197
 - Strukturiertes Programm, 176
 - Systemzeit, 330
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Programminformationen
 - In der Aufrufstruktur, 208
- Programmkarte
 - Anlaufparameter konfigurieren, 121
 - Anlegen, 126
 - In CPU einfügen, 118
 - Übersicht, 118
- Programmsteuerung (SCL), 310
 - CASE, 312
 - CONTINUE, 316
 - EXIT, 317
 - FOR, 313
 - GO TO, 317
 - IF-THEN, 311
 - REPEAT, 315
 - RETURN, 318
 - WHILE, 314
- Programmstruktur, 178
- Programmzyklus-OB, 75
- Projekt
 - Codebaustein schützen, 167
 - Laden in CPU, 199
 - Leere Übertragungskarte, 134
 - Programmkarte, 126
 - Übertragungskarte, 122
 - Vergleichen und synchronisieren, 1221
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
 - Verlorenes Passwort, 134
 - Zugriffsschutz durch Passwörter, 163
- Projektansicht, 39, 40
- Protokoll
 - Frei programmierbare Kommunikation, 944
 - ISO on TCP, 617
 - Kommunikation, 944
 - Modbus, 944
 - PROFINET RT, 617

- TCP, 617
 - UDP, 617
 - USS, 944
 - Protokoll der frei programmierbaren Kommunikation, 944
 - Protokolle, Kommunikation, 619
 - Prozessabbild
 - Ausgänge mit SETIO schreiben, 377
 - Beobachten, 1224
 - Eingänge mit GETIO lesen, 376
 - Forcefunktion, 1230
 - Forcen, 1230
 - Prozessabbildbereich mit GETIO_PART lesen, 378
 - Prozessabbildbereich mit SETIO_PART übertragen, 379
 - Überwachungsstatus oder -wert, 1223
 - Zustand, 1224, 1230
 - Prozessalarm-OB, 78
 - Prüfen der Verbindung, 684
 - PRVREC (Datensatz bereitstellen), 398
 - PTO (Impulsfolge)
 - CTRL_PTO (Impulsfolge), 472
 - CTRL_PWM (Impulsdauermodulation), 470
 - Funktionsweise, 475
 - Impulskanäle konfigurieren, 477
 - Kann nicht geforct werden, 1231
 - PtP-Anweisungen Rückgabewerte, 963
 - PtP-Fehlerklassen, 965, 1117
 - PtP-Kommunikation, 944
 - Anschlüsse konfigurieren, 947
 - Beispielprogramm, 989
 - Beispielprogramm, Ausführen, 997
 - Beispielprogramm, STEP 7-Programmierung, 995
 - Konfiguration des Beispielprogramms, 990
 - Parameter konfigurieren, 950
 - Programmieren, 987
 - Terminalemulator für Beispielprogramm, 997
 - PtP-Kommunikation, 3964(R)
 - Anschlüsse konfigurieren, 960
 - Konfigurieren von Priorität und Protokollparametern, 961
 - Punkt-zu-Punkt-Kommunikation, 944
 - Punkt-zu-Punkt-Programmierung, 987
 - PUT (Daten in entfernte CPU schreiben), 799
 - Verbindung konfigurieren, 602
 - PWM (Impulsdauermodulation)
 - CTRL_PTO (Impulsfolge), 472
 - CTRL_PWM (Impulsdauermodulation), 470
 - E/A-Adressen, 480
 - Funktionsweise, 475
 - Impulsdauer, 478
 - Impulsdauer ändern, 480
 - Impulskanäle konfigurieren, 477
 - Kann nicht geforct werden, 1231
 - Zykluszeit, 478
 - Zykluszeit ändern, 480
- Q**
- QRY_CINT (Weckalarmparameter abfragen), 409
 - QRY_DINT (Status des Verzögerungsalarms abfragen), 414
 - QRY_TINT (Status des Uhrzeitalarms abfragen), 413
 - Querverweis zum Anzeigen der Verwendung, 207
- R**
- R (Ausgang rücksetzen), 212
 - R_TRIG (Variable bei positiver Signalflanke setzen), 216
 - RALRM (Alarm empfangen), 380, 388
 - RCV_CFG (Serielle Empfangsparameter dynamisch konfigurieren), veraltet, 1120
 - RCV_PTP (Empfangsmeldungen aktivieren), veraltet, 1128
 - RCV_RST (Empfangspuffer löschen), veraltet, 1129
 - RCVREC (Datensatz empfangen), 396
 - RD_ADDR (E/A-Adressen über Hardwarekennung ermitteln), 532
 - RD_LOC_T (Lokalzeit lesen), 330
 - RD_SINFO (Startinformation des aktuellen OBs auslesen), 421
 - RD_SYS_T (Uhrzeit lesen), 330
 - RDREC (Datensatz lesen), 373, 388
 - RE_TRIGR (Zyklusüberwachungszeit neu starten), 304
 - READ_BIG (Daten in Big-Endian-Format lesen), 260
 - READ_DBL (Aus Datenbaustein im Ladespeicher lesen), 522
 - READ_LITTLE (Daten in Little-Endian-Format lesen), 260
 - Receive_Config (Empfangskonfiguration), 970
 - Receive_P2P (Punkt-zu-Punkt-Daten empfangen), 980
 - Receive_Reset (Empfangspuffer löschen), 982
 - Redundanz
 - Redundanz-Clients, 769
 - Redundanzdomänen, 771
 - Remanente Bausteinvariablen
 - Laden im Betriebszustand RUN, 1238
 - Remanenter Speicher, 28, 92
 - CPU 1211C, 1263
 - CPU 1212C, 1276

CPU 1214C, 1288
 CPU 1215C, 1300
 CPU 1217C, 1314
 REPEAT, SCL, 315
 REPLACE (Zeichen in einer Zeichenkette ersetzen), 356
 RESET_BF (Bitfeld rücksetzen), 213
 RET (Rückgabewert), 300
 RETURN, SCL, 318
 Rezept
 Beispielprogramm, 491
 DB-Struktur, 483
 RecipeExport (Rezeptexport), 487
 RecipeImport (Rezeptimport), 489
 Übersicht, 482
 Rezeptprogramm Beispiel, 491
 Richtlinien
 Einbau der CPU, 49
 Erdung, 62
 induktive Lasten, 64
 Installation, 43
 Lampenlasten, 63
 Trennung, 61
 Verdrahtungsrichtlinien, 60, 62
 Vorgehensweise für den Einbau, 48
 Richtlinien für Potentialtrennung, 61
 RIGHT (rechte Zeichen einer Zeichenkette lesen), 353
 Ringport, 776
 Ringtopologie, 769
 ROL (Links rotieren) und ROR (Rechts rotieren), 325
 ROUND (Zahl runden), 289
 RS (Flipflop rücksetzen/setzen), 213
 RS232- und RS485-Kommunikationsmodule, 941
 RS485-Stecker
 Abschluss, 942
 RT (Zeit zurücksetzen), 217
 RTS (Hardware-Flusskontrolle, PtP), 949
 RTS-Einschaltverzögerung, -
 Ausschaltverzögerung, 951
 Rückgabewerte
 Anweisungen für die offene
 Benutzerkommunikation, 727
 PtP-Anweisungen, 963
 Rückgabewerte der Anweisungen für die offene
 Benutzerkommunikation, 727
 Rückgabewerte Empfangs-Laufzeit, 980, 1128
 RUNTIME (Programmlaufzeit messen), 309

S

S (Ausgang setzen), 212
 S_CONV (Zeichenkette umwandeln), 337

S_MOV (Zeichenkette verschieben), 337
 S7-Kommunikation
 Verbindung konfigurieren, 602
 S7-Routing, 776
 SCALE_X (Skalieren), 291
 Schachtelungstiefe, 67
 Schaltpläne
 CB 1241 RS 485, 1426
 CPU 1211C, 1273
 CPU 1212C, 1285
 CPU 1214C, 1297
 CPU 1215C, 1310
 CPU 1217C, 1327
 SB 1221, 1386
 SB 1222, 1389
 SB 1223, 1392, 1395
 SB 1231, 1398
 SB 1231 RTD, 1410
 SB 1231 Thermoelement, 1407
 SB 1232, 1399
 SM 1221, 1331
 SM 1222, 1335
 SM 1223, 1342, 1347
 SM 1231, 1350
 SM 1231 RTD, 1367
 SM 1231 Thermoelement, 1361
 SM 1232, 1353
 SM 1234, 1356
 SM 1278 IO-Link-Master, 1374
 Schließer-/Öffnerkontakt, 209
 Schließer-/Öffnerspule, 211
 Schneller Zähler, 544, 571
 Betriebsphase, 562
 Kann nicht geforct werden, 1231
 Konfiguration, 560
 Zählarten, 561
 Schnittstellenkonfiguration
 Anweisungen, 988
 Schreiben in DBs, E/A oder Speicher, 195, 258
 Schutz
 CPU Zugriffsschutz, 163
 Knowhow-Schutz für einen Codebaustein, 167
 Schutz vertraulicher PLC-Konfigurationsdaten, 129,
 160
 Schutzart, 1259
 Schutzbeschaltungen für induktive Lasten, 64
 Schutzgrad, 1259
 Schutzstufe
 Codebaustein, 167
 CPU, 163

- Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
- Verlorenes Passwort, 134
- SCL (Structured Control Language)
 - Adressierung, 192
 - Ausdrücke, 192
 - Bausteine aufrufen, 178
 - Bedingungen, 192
 - Bitverknüpfung, 209
 - EN und ENO (Signalfluss), 197
 - FB oder FC aufrufen, 192
 - Operatoren, 192
 - Priorität von Operatoren, 192
 - Programmiereditor, 190
 - Programmsteuerung, 310
 - Steuerungsanweisungen, 192, 310
 - Übersicht, 190
 - Umwandlungsanweisungen, 286
 - Var-Abschnitt, 190
 - Werte vergleichen, 230
 - Zeiten, 217
- Secure Communication, 593
- Security-Assistent, 160
- SEL (Selektieren), 321
- SEND_CFG (Serielle Übertragungsparameter dynamisch konfigurieren), veraltet, 1119
- Send_Config (Sendekonfiguration), 968
- Send_P2P (Punkt-zu-Punkt-Daten senden), 977
- SEND_P2P (Punkt-zu-Punkt-Daten senden)
 - Parameter LENGH und BUFFER, 979
- SEND_PTP (Sendepufferdaten übertragen), veraltet, 1125
- Serialize, 250
- Serielle Kommunikation, 944
- Service und Support, 3
- SET_BF (Bitfeld setzen), 213
- SET_CINT (Weckalarm parametrieren), 407
- Set_Features (Erweiterte Funktionen setzen), 986
- SET_TIMEZONE (Zeitzone setzen), 333
- SET_TINTL (Datum und Uhrzeit eines Uhrzeitalarms festlegen), 411
- SETIO, 377
- SETIO_PART, 379
- SGN_GET (RS232-Signale abfragen), veraltet, 1130
- SGN_GET (RS232-Signale abrufen), 983
- SGN_SET (RS-232-Signale einstellen), veraltet, 1131
- Shared Device
 - Konfiguration, 752
 - Konzept, 749
- Shared I-Device, Konfiguration, 757
- SHL (Links schieben) und SHR (Rechts schieben), 324
- Sicherheit
 - Kopierschutz, 168
 - Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168
 - Verlorenes Passwort, 134
- Sicherheitsereignisse im Diagnosepuffer, 96
- Sichern einer CPU, 1249
- Siemens-Sicherheitszertifikat, Webseiten, 867
- Signal_Set (RS232-Signale setzen), 984
- Signalboards (SB)
 - Ausbau, 52
 - Darstellung Analogausgang (Spannung), 1359, 1401
 - Darstellung Analogausgang (Strom), 1359, 1402
 - Darstellung Eingang (Spannung), 1358, 1400
 - Darstellung Eingang (Strom), 1358, 1401
 - Einbau, 52
 - Konfiguration von Parametern, 169
 - Leistungsbedarf, 1437
 - Module hinzufügen, 140
 - SB 1221 DI 4 x 24 V DC, 200 kHz, 1385
 - SB 1221 DI 4 x 5 V DC, 200 kHz, 1385
 - SB 1222 DQ 4 x 24 V DC, 200 kHz, 1387
 - SB 1222 DQ 4 x 5 V DC, 200 kHz, 1387
 - SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1393
 - SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1390
 - SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1390
 - SB 1231 AI 1 x 12 Bit, 1396
 - SB 1231 AI 1 x 16 Bit RTD, 1408
 - SB 1231 AI 1 x 16 Bit Thermoelement, 1403
 - SB 1232 AO 1 x 12 Bit, 1398
 - Übersicht, 31
- Signalmodule
 - SM 1222 DQ 16 x 24 V DC stromziehend, 1333
- Signalmodule (SM)
 - Ausbau, 55
 - Darstellung Analogausgang (Spannung), 1359, 1401
 - Darstellung Analogausgang (Strom), 1359, 1402
 - Darstellung Analogeingang (Spannung), 1358, 1400
 - Darstellung Analogeingang (Strom), 1358, 1401
 - Einbau, 54
 - Erweiterungskabel, 58
 - Konfiguration von Parametern, 169
 - Leistungsbedarf, 1437
 - Module hinzufügen, 140
 - Schrittantwortzeiten, 1357
 - SM 1221 DI 16 x 24 V DC, 1330
 - SM 1221 DI 8 x 24 V DC, 1330

- SM 1222 DO 16 x Relais, 1333
- SM 1222 DO 8 x Relais, 1332
- SM 1222 DO 8 x Relais (Umschaltung), 1332
- SM 1222 DQ 16 x 24 V DC, 1333
- SM 1222 DQ 8 x 24 V DC, 1332
- SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1339
- SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC
stromziehend, 1339
- SM 1223 DI 16 x 24 V DC, DQ 16 x Relais, 1339
- SM 1223 DI 8 x 120/230 V AC/DQ 8 x Relais, 1345
- SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1339
- SM 1223 DI 8 x 24 V DC, DQ 8 x Relais, 1339
- SM 1231 AI 4 x 13 Bit, 1347
- SM 1231 AI 4 x 16 Bit, 1347
- SM 1231 AI 4 x 16 Bit TC, 1360
- SM 1231 AI 4 x RTD x 16 Bit, 1366
- SM 1231 AI 8 x 13 Bit, 1347
- SM 1231 AI 8 x 16 Bit TC, 1360
- SM 1231 AI 8 x RTD x 16 Bit, 1366
- SM 1232 AO 2 x 14 Bit, 1352
- SM 1232 AO 4 x 14 Bit, 1352
- SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 1354
- SM 1278 4xIO-Link-Master, 1372
- Übersicht, 32
- Signalverarbeitungsfehler, 984, 985, 1131, 1132
- Simulatoren, 1430
- SIN (Sinuswert bilden), 242
- Slice (eines Variablentyps), 114
- SM 1231 RTD
 - Auswahltabellen, 1369, 1411
- SM und SB
 - Gerätekonfiguration, 135
 - Vergleichstabelle, 31
- Smartphone, Zugriff auf Webserver, 856
- SMS, 1193
- Snubber-Schutzbeschaltungen für induktive Lasten, 64
- Software-Flusskontrolle, 950
- Sommerzeit TimeTransformationRule, 332
- Sonderzeichen
 - Benutzerdefinierte Webseiten, 909
- Spannungsversorgungsmodul
 - PM1207, 1434
- Speicher
 - A (Prozessabbild der Ausgänge), 101
 - Adressen der Peripherieeingänge (Forcetabelle), 1230
 - Arbeitsspeicher, 92
 - Beobachten der Speicherauslastung, 1219
 - E (Prozessabbild der Eingänge), 100
 - L (Lokaldatenspeicher), 98
 - Ladespeicher, 92
 - M (Bitmerker), 102
 - Remanenter Speicher, 92
 - Systemspeicher, 94
 - Taktmerker, 94
 - Temporärer Speicher, 102
- Speicherbereich A
 - Impulsausgänge, 475
 - Impulskanäle konfigurieren, 477
- Speicherbereich E
 - Adressen der Peripherieeingänge (Forcetabelle), 1230
 - Beobachten, 1223
 - Beobachtungstabelle, 1223
 - Forcefunktion, 1230
 - Forcen, 1230
 - Forcetabelle, 1230
 - KOP beobachten, 1224
- Speicherbereiche
 - Adressierung von Booleschen Werten oder Bitwerten, 99
 - Direkter Zugriff, 99
 - Prozessabbild, 99
- Speicherbereiche, Anzeige mit Webserver, 868
- Speicherkarte
 - Anlaufparameter konfigurieren, 121
 - Verlorenes Passwort, 134
- Speichern von Sicherungsdateien, 1250
- Sprachen wechseln, benutzerdefinierte Webseiten, 928
- Sprachen, benutzerdefinierte Webseiten, 928
- Sprachumschaltung für benutzerdefinierte Webseiten programmieren, 929
- Spulen, 209
- SQR (Quadrat bilden), 242
- SQRT (Quadratwurzel bilden), 242
- SR (Flipflop setzen/rücksetzen), 213
- SRT_DINT (Verzögerungsalarm starten), 414
- Standard-Datenbausteine, 183
- Standard-Hutschiene, 48
- Standard-Maschinenprojekte (Konfigurationssteuerung), 141
- Standard-Webseite "Diagnose", 868, 870
- Standard-Webseite "Einführung", 866
- Standard-Webseite "Kommunikation", 876
- Standard-Webseite "Modulinformationen", 872
- Standard-Webseite "Start", 867
- Standard-Webseite "Variablenstatus", 879
- Standard-Webseiten, 847
 - An- und Abmelden, 864
 - Aufbau, 861
 - Betriebszustand wechseln, 867
 - Cookie-Einschränkungen, 938

- Diagnose, 868, 870
 - Einführung, 866
 - JavaScript, 938
 - Kommunikation, 876
 - Modulinformationen, 872
 - Sicherer Zugriff, 856
 - Start, 867
 - Variablenstatus, 879
 - Zugriff über PC, 855
 - Startbedingungen, 953
 - STARTUP, Betriebszustand
 - Forcefunktion, 1230
 - Station
 - Informationen mit GetStationInfo lesen, 439
 - STEP 7
 - Anfangswert eines FB, 181
 - Anlaufverarbeitung, 74
 - Anzahl der Codebausteine, 67
 - ASi, 791
 - AS-i-Port, 790
 - Bausteinaufrufe, 67
 - Bausteine aus einer Online-CPU kopieren, 205
 - Beobachten, 1223, 1224
 - Betriebszustände, 71
 - Codebausteine im Anwenderprogramm aufrufen, 178
 - CPU konfigurieren, 153
 - Datenbaustein (DB), 67
 - Eigenschaft Uhrzeitsynchronisation (PROFINET), 615
 - Ethernet-Port, 608
 - Forcefunktion, 1230
 - Forcen, 1230
 - Funktion (FC), 180
 - Funktionsbaustein (FB), 67, 181
 - Funktionsweise, 1226
 - Gerätekonfiguration, 135
 - Gültige FC-, FB- und DB-Nummern, 67
 - HSC-Konfiguration, 560
 - Instanz-Datenbaustein (DB), 181
 - Kommunikationslast, 91
 - Laden in CPU, 199
 - Lineare und strukturierte Programme, 176
 - Module hinzufügen, 140
 - Module konfigurieren, 169
 - Netzwerkverbindung, 600
 - Neues Gerät hinzufügen, 137
 - Passwortschutz, 167
 - Portalansicht und Projektansicht, 39
 - Prioritätsklasse (OB), 75
 - PROFIBUS, 786
 - PROFINET IO-Device hinzufügen, 737
 - PROFINET-Port, 608
 - Programmkarte, 118
 - RTM (Betriebsstundenzähler), 334
 - Vergleichen und synchronisieren, 1221
 - Zuweisen einer IP-Adresse zu einer Online-CPU, 606
 - Zykluszeit, 90, 91
 - STEP 7 Webseiten, 4
 - STEP 7-Programmierung
 - Benutzerdefinierte Webseiten, 913
 - PtP-Beispielprogramm, 995
 - STEP 7
 - Versionskompatibilität, 40
 - Steuer-DB für benutzerdefinierte Webseiten
 - Anforderungsbefehle und -zustände, 932
 - Globale Befehle, 932
 - Parameter WWW-Anweisung, 913
 - Stoppbits, 948
 - Stoßwellenfestigkeit, 1258
 - STP (Programm beenden), 305
 - Strg_TO_Chars (Zeichenkette in Array aus CHAR umwandeln), 346
 - STRG_VAL (Zeichenkette in Zahlenwert umwandeln), 337
 - String
 - Datentyp String, 336
 - Datentyp STRING, 111
 - S_MOV (Zeichenkette verschieben), 337
 - Übersicht über Zeichenkettenoperationen, 350
 - Stromaufnahme, 45, 1437
 - Strukturierte Programmierung, Bausteinstruktur, 176
 - SUB (Subtrahieren), 236
 - Subnetzmaske, 608
 - Support, 3
 - SWAP (Anordnung ändern), 255
 - SWITCH (Sprungverteilung), 298
 - Synchronisation
 - Eigenschaft Uhrzeitsynchronisation (PROFINET), 615
 - Synchronisierung der Startwerte der Datenbausteine, 203
 - Systemspeicherbyte, 95
 - Systemuhr
 - RD_SYS_T (Uhrzeit lesen), 330
 - WR_LOC_T (Lokalzeit einstellen), 330
 - WR_SYS_T (Uhrzeit einstellen), 330
 - Systemvoraussetzungen, 37
- T**
- T_ADD (Zeiten addieren), 328
 - T_COMBINE (Zeiten verknüpfen), 329

- T_CONFIG (Schnittstelle konfigurieren), 717
- T_CONV (Zeiten umwandeln und extrahieren), 327
- T_DIAG, 684
- T_DIFF (Zeitdifferenz), 329
- T_RESET, 682
- T_SUB (Zeiten subtrahieren), 328
- Tablet, Zugriff auf Webserver, 856
- Tageszeit
 - CPU online konfigurieren, 1212
- Taktmerkerbyte, 95
- TAN (Tangenswert bilden), 242
- Taskcards
 - Spalten und Überschriften, 636, 649, 656, 673, 1000, 1021, 1091, 1133, 1147, 1165
- TCON, 657
 - Konfiguration, 601
 - Verbindungs-IDs, 621
 - Verbindungsparameter, 624
- TCON, TDISCON, TSEND und TRCV
 - Versionen, 656, 673
- TCON_Param, 624
- TCP
 - Ad-hoc-Modus, 620
 - Parameter, 624
 - Protokolle, 617
 - Verbindungs-IDs, 621
 - Verbindungskonfiguration, 601
- TCP/IP-Kommunikation, 617
- TDISCON, 657
- Technische Daten, 1253
 - Allgemeine technische Daten, 1253
 - BB 1297, 1413
 - Bemessungsspannungen, 1260
 - CB 1241 RS485, 1425
 - CM 1241 RS232, 1427
 - CM 1241 RS422/485, 1428
 - CPU 1211C AC/DC/Relais, 1263
 - CPU 1211C DC/DC/DC, 1263
 - CPU 1211C DC/DC/Relais, 1263
 - CPU 1212C AC/DC/Relais, 1276
 - CPU 1212C DC/DC/DC, 1276
 - CPU 1212C DC/DC/Relais, 1276
 - CPU 1214C AC/DC/Relais, 1288
 - CPU 1214C DC/DC/DC, 1288
 - CPU 1214C DC/DC/Relais, 1288
 - CPU 1215C AC/DC/Relais, 1300
 - CPU 1215C DC/DC/DC, 1300
 - CPU 1215C DC/DC/Relais, 1300
 - CPU 1217C DC/DC/DC, 1314
 - Darstellung Analogausgang (Spannung), 1359, 1401
 - Darstellung Analogausgang (Strom), 1359, 1402
 - Darstellung Analogeingang (Spannung), 1358, 1400
 - Darstellung Analogeingang (Strom), 1358, 1401
 - Eingangssimulatoren, 1430
 - Elektromagnetische Verträglichkeit (EMV), 1257
 - Industrienumgebungen, 1257
 - Memory Cards, 1430
 - Potentiometermodul, 1432
 - SB 1221 DI 4 x 24 V DC, 200 kHz, 1385
 - SB 1221 DI 4 x 5 V DC, 200 kHz, 1385
 - SB 1222 DQ 4 x 24 V DC, 200 kHz, 1387
 - SB 1222 DQ 4 x 5 V DC, 200 kHz, 1387
 - SB 1223 DI 2 x 24 V DC, DQ 2 x 24 V DC, 1393
 - SB 1223 DI 2 x 24 V DC/DQ 2 x 24 V DC, 200 kHz, 1390
 - SB 1223 DI 2 x 5 V DC/DQ 2 x 5 V DC, 200 kHz, 1390
 - SB 1231 AI 1 x 12 Bit, 1396
 - SB 1231 AI 1 x 16 Bit RTD, 1408
 - SB 1231 AI 1 x 16 Bit Thermoelement, 1403
 - SB 1232 AO 1 x 12 Bit, 1398
 - Schrittantwortzeiten (CPU), 1271, 1284, 1296, 1308, 1324
 - Schrittantwortzeiten (SB), 1400
 - Schrittantwortzeiten (SM), 1357
 - Signalmodul SM 1231 AI 4 x RTD x 16 Bit, 1366
 - Signalmodul SM 1231 AI 8 x RTD x 16 Bit, 1366
 - SM 1221 DI 16 x 24 V DC, 1330
 - SM 1221 DI 8 x 24 V DC, 1330
 - SM 1222 DO 16 x Relais, 1333
 - SM 1222 DO 8 x Relais, 1332
 - SM 1222 DO 8 x Relais (Umschaltung), 1332
 - SM 1222 DQ 16 x 24 V DC, 1333
 - SM 1222 DQ 16 x 24 V DC stromziehend, 1333
 - SM 1222 DQ 8 x 24 V DC, 1332
 - SM 1223 DI 16 x 24 V DC, DQ 16 x 24 V DC, 1339
 - SM 1223 DI 16 x 24 V DC, DQ 16 x Relais, 1339
 - SM 1223 DI 8 24 V DC, DQ 8 x Relais, 1339
 - SM 1223 DI 8 x 120/230 V AC/DQ 8 x Relais, 1345
 - SM 1223 DI 8 x 24 V DC, DQ 8 x 24 V DC, 1339
 - SM 1231 AI 4 x 13 Bit, 1347
 - SM 1231 AI 4 x 16 Bit, 1347
 - SM 1231 AI 4 x 16 Bit TC, 1360
 - SM 1231 AI 8 x 13 Bit, 1347
 - SM 1231 AI 8 x 16 Bit TC, 1360
 - SM 1232 AO 2 x 14 Bit, 1352
 - SM 1232 AO 4 x 14 Bit, 1352
 - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 1354
 - SM 1278 4xIO-Link-Master, 1372
 - Umgebungsbedingungen, 1258
 - Zulassungen, 1253
- Technischer Support, 3

- Technischer Support von Siemens, 3
 - Technologieanweisungen, 544, 571
 - Technologiemodul, SM 1278 4xIO-Link-Master, 1372
 - Telecontrol, 1190
 - TeleControl
 - Kommunikationsprozessoren, 1185
 - TeleService über GPRS, 1190
 - Teleservice-Kommunikation
 - TM_MAIL (E-Mail senden), 1197
 - Temporärer Speicher
 - Maximum je OB-Prioritätsstufe, 102
 - Verwendung durch Bausteine, 102
 - Terminalemulator für PtP-Beispielprogramm, 997
 - Testen im Betriebszustand RUN, 1232, 1240
 - Thermischer Bereich, 44, 47
 - Thermoelement
 - Filterauswahltabelle SB 1231
 - Thermoelement, 1405
 - Filterauswahltabelle SM 1231
 - Thermoelement, 1363
 - Grundlegende Funktionsweise, 1363, 1405
 - Kompensation der kalten Verbindungsstelle, 1363, 1405
 - SB 1231 AI 1 x 16 Bit, 1403
 - SB 1231 Filterauswahltabelle, 1406
 - SM 1231 Thermoelement-Auswahltabelle, 1363
 - TIA Portal, Portalansicht und Projektansicht, 39
 - Time
 - Datentyp Time, 109
 - DTL (Datentyp Date and Time long), 110
 - RD_LOC_T (Lokalzeit lesen), 330
 - SET_TIMEZONE (Zeitzone setzen), 333
 - T_DIFF (Zeitdifferenz), 329
 - TOD (Datentyp Time of Day), 109
 - TimeTransformationRule für Sommerzeit, 332
 - TM_MAIL (E-Mail senden), 1197
 - TMAIL_C, 688
 - Topologie
 - Ring, 769
 - Topologieansicht, 39
 - Trace-Funktion, 1242
 - TRCV, 657
 - Verbindungs-IDs, 621
 - TRCV (Daten über Ethernet empfangen (TCP))
 - Ad-hoc-Modus, 620
 - Parameterkonfiguration, 736
 - TRCV_C
 - Ad-hoc-Modus, 620
 - TRCV_C (Daten über Ethernet empfangen (TCP)), 636
 - Konfiguration, 601
 - Verbindungs-IDs, 621
 - Verbindungsparameter, 624
 - Trigonometrische Anweisungen, 242
 - TRUNC (Ganzzahl erzeugen), 289
 - TS-Adapter, 31
 - TSAP (Transport Service Access Point), 603
 - Allgemeine Parameter konfigurieren, 734, 804
 - Anweisungen zum Zuweisen zu Geräten, 617
 - Definition, 619
 - Eingeschränkte TSAPs und Portnummern, 728
 - TSEND, 657
 - Verbindungs-IDs, 621
 - TSEND_C (Daten über Ethernet senden (TCP)), 636
 - Anweisungskonfiguration, 735
 - Konfiguration, 601
 - Verbindungs-IDs, 621
 - Verbindungsparameter, 624
 - TSEND_C und TRCV_C
 - alte Versionen, 649
 - Versionen, 636
 - TURCV (Daten über Ethernet empfangen (UDP)), 712
 - Konfiguration, 601
 - Verbindungsparameter, 624
 - TUSEND (Daten über Ethernet senden (UDP)), 712
 - Konfiguration, 601
 - Parameter, 624
- ## U
- Überspannungskategorie, 1259
 - Übertragungsbaustein (T-Baustein), 734
 - Übertragungskarte, 122
 - Anlaufparameter konfigurieren, 121
 - In CPU einfügen, 118
 - Leere Übertragungskarte für ein verlorenes Passwort, 134
 - Übersicht, 118
 - Verlorenes Passwort, 134
 - Übertragungskarte (Programmkarte), 1430
 - Überwachen von Variablen über Webserver, 879
 - UDP
 - Parameter, 624
 - Verbindungskonfiguration, 601
 - UDP-Protokoll, 617
 - UFILL_BLK (Speicher ununterbrechbar mit Bitmuster belegen), 253
 - Uhr
 - Echtzeituhr, 97
 - RD_LOC_T (Lokalzeit lesen), 330
 - RD_SYS_T (Uhrzeit lesen), 330
 - WR_LOC_T (Lokalzeit einstellen), 330
 - WR_SYS_T (Uhrzeit einstellen), 330
 - Uhrzeit
 - T_COMBINE (Zeiten verknüpfen), 329

Uhrzeit-OB, 83
 Uhrzeitsynchronisation, 173
 Umgebungsbedingungen
 Betriebsbedingungen, 1258
 Transport und Lagerung, 1258
 UMOVE_BLK (Bereich ununterbrechbar kopieren), 244
 Umwandlung (SCL-Anweisungen), 286
 UND (logische Verknüpfung), 318
 Unterschiede
 in Anweisungen TSEND_C und TRCV_C, 635
 in den Anweisungen TCON, TDISCON, TSEND und TRCV, 656
 in Modbus RTU-Anweisungen, 1089
 in Modbus TCP-Anweisungen, 1020
 in PtP-Anweisungen, 944
 in USS-Anweisungen, 999
 UPPER_BOUND (Obere ARRAY-Grenze auslesen), 257

V

VAL_STRG (Zahlenwert in Zeichenkette umwandeln), 337
 Variable
 Forcefunktion, 1230
 Slice, 114
 Überlagerung, 116
 Überwachungsstatus oder -wert, 1223
 Variablen überwachen und ändern über Webserver, 879
 Variablenindex für ein Array, 266
 VARIANT_TO_DB_ANY (VARIANT in DB_ANY konvertieren), 294
 VariantGet (VARIANT Variablenwert lesen), 262
 VariantPut (VARIANT Variablenwert schreiben), 263
 Verbindung zwischen lokaler und Partner-CPU, 600
 Verbindungen
 Anzahl der Verbindungen (PROFINET/ PROFIBUS), 596
 Verbindungen mit mehreren Teilnehmern
 Ethernet-Protokolle, 803
 Verbindungsarten, 803
 Verdrahtungsrichtlinien, 62
 Abstand für Luftströmung und Kühlung, 44
 Erdung, 62
 Voraussetzungen, 60
 Vergleichstabelle
 CPU-Varianten, 28
 HMI-Geräte, 32
 Module, 31
 Verknüpfung mit einer CPU, einer Memory Card oder einem Passwort, 168

Verlorenes Passwort, 134
 Verlust der CPU-Kommunikation an Modulen, 82
 Verpolschutz, 1260
 Verschiedene PtP-Parameterfehler, 964
 Verschmutzungsgrad, 1259
 Verschmutzungsgrad/Überspannungskategorie, 1259
 Versionen von Anweisungen, 636, 649, 656, 673, 1000, 1021, 1091, 1133, 1147, 1165
 Verzeichnisse, Sprachen für benutzerdefinierte Webseiten, 928
 Verzögerungsalarme, 414
 Verzögerungsalarm-OB, 76
 Visualisierung, HMI-Geräte, 32
 Voraussetzungen PC, 37
 Voraussetzungen, Installation, 37

W

Warmstart, 71
 Warteschlange, 86
 Wartezeit, 948
 WChar (Datentyp Word Character), 111
 Webseite "Variablenstatus", 879
 Webseiten
 STEP 7-Service, -Support und -Dokumentation, 4
 Webserver
 Aktivieren, 850
 Aktualisierungsrate, 850
 Benutzerdefinierte Webseiten, 895
 Benutzerkonfiguration, 852
 Darstellung auf einem Mobilgerät, 862
 Einschränkungen, 937
 Konventionen bei Anführungszeichen, 909
 maximale Anzahl HTTP-Verbindungen, 937
 mobiles Gerät Zugriff, 856
 Standard-Webseiten, 855
 Unterstützte Browser, 849
 Zugriff über CP-Modul, 858
 Wechsel von RUN nach STOP, 97
 Weckalarm-OB, 77
 Werte eines DBs zurücksetzen, 1225
 Werte eines Online-DBs erfassen, 1225
 Werte vergleichen, 230
 WHILE, SCL, 314
 Wiederherstellen einer Sicherungskopie, 1252
 WR_LOC_T (Lokalzeit einstellen), 330
 WR_SYS_T (Uhrzeit einstellen), 330
 WRIT_DBL (In Datenbaustein im Ladespeicher schreiben), 522
 WRITE_BIG (Daten in Big-Endian-Format schreiben), 260

WRITE_LITTLE (Daten in Little-Endian-Format schreiben), 260

WRREC (Datensatz schreiben), 373, 388

WString (Datentyp Word String), 111

WWW (benutzerdefinierte Webseiten synchronisieren), 913

X

x-Box (logische EXKLUSIV-ODER-Verknüpfung in FUP), 210

XON / XOFF, 950

XOR (logische Verknüpfung), 318

Z

Zählarten

 Schneller Zähler, 561

Zahlen

 Binär, 107

 Ganzzahl, 108

 Real, 108

Zähler

 Anzahl, 30, 1266, 1278, 1290, 1303, 1317

 CTD (Rückwärtszählen), 225

 CTRL_HSC (Schnellen Zähler steuern), 571

 CTRL_HSC_EXT (Schnellen Zähler steuern (erweitert)), 544

 CTU (Vorwärtszählen), 225

 CTUD (Vorwärts- und Rückwärtszählen), 225

 Größe, 30, 1266, 1278, 1290, 1303, 1317

 HSC-Konfiguration, 560

 Operation (Standardzähler), 226

Zeichen für das Ende einer Meldung, 957

Zeichen für den Beginn einer Meldung, 953

Zeichenabstand, 957

Zeichenfolge

 Meldungsbeginn, 953

 Meldungsende, 957

Zeichenposition, Meldungslänge, 958

Zeit

 RD_SYS_T (Uhrzeit lesen), 330

 T_ADD (Zeiten addieren), 328

 T_CONV (Zeiten umwandeln und extrahieren), 327

 T_SUB (Zeiten subtrahieren), 328

 WR_LOC_T (Lokalzeit einstellen), 330

 WR_SYS_T (Uhrzeit einstellen), 330

Zeit akkumulieren (TONR), 217

Zeit zurücksetzen (RT), 217

Zeiten

 Anzahl, 30, 1266, 1278, 1290, 1303, 1317

 Funktionsweise, 220

 Größe, 30, 1266, 1278, 1290, 1303, 1317

 RT (Zeit zurücksetzen), 217

 TOF (Ausschaltverzögerung), 217

 TON (Einschaltverzögerung), 217

 TONR (Zeit akkumulieren), 217

 TP (Impuls), 217

Zeitfehler-OB, 79

Zugreifen

 Benutzerdefinierte Webseiten, 915

Zugriffsschutz, CPU, 163

Zulassung für das Seewesen, 1256

Zulassung koreanische Zertifizierung, 1256

Zulassungen

 ATEX, 1255

 Australien und Neuseeland - RCM Mark, 1256

 CE, 1253

 cULus, 1254

 FM, 1254

 Koreanische Zertifizierung, 1256

 Seewesen, 1256

 Zulassung CCCEX, 1255

Zuordnung der Lokaldaten in der Aufrufstruktur, 102

Zurücksetzen auf Werkseinstellungen, 1214

Zustand

 LED-Anzeigen, 1205

 LED-Anzeigen

 (Kommunikationsschnittstelle), 941

Zustands-OB, 84

Zyklus

 Forcefunktion, 1230

 Übersicht, 90

Zykluszeit

 Beobachten, 1219

 Konfiguration, 91

 Übersicht, 90

Zykluszeitüberwachung (Anweisung RE_TRIGR), 304

