

SIEMENS

SIMATIC

S7-1200 Easy Book

Gerätehandbuch

Vorwort

Einführung in die leistungsstarke und flexible S7-1200	1
STEP 7 vereinfacht Ihre Arbeit	2
Erste Schritte	3
SPS-Grundlagen leicht gemacht	4
Einfache Erstellung der Gerätekonfiguration	5
Programmierung leicht gemacht	6
Einfache Kommunikation zwischen Geräten	7
Einfache PID-Regelung	8
Webserver für einfachen Internetanschluss	9
Einfache Bewegungssteuerung	10
Einfaches Arbeiten mit den Online-Tools	11
IO-Link – ganz einfach	12
Technische Daten	A
Austauschen einer V3.0-CPU durch eine V4.0-CPU	B

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

Willkommen in der Welt der S7-1200. Die kompakte Steuerung SIMATIC S7-1200 ist die modulare, Platz sparende Steuerung für klein dimensionierte Automatisierungssysteme, die für Logik, HMI und Netzwerkfunktionen entweder einfache oder hoch entwickelte Funktionen benötigen. Durch das kompakte Design, den günstigen Preis und die leistungsstarken Funktionen eignet sich die S71200 hervorragend für kleinere Steuerungsanwendungen.

Dank der Einbindung der SIMATIC in das Konzept der "Totally Integrated Automation" (TIA) bieten Ihnen die Produktfamilie der S7-1200 und die Programmiersoftware TIA Portal die erforderliche Flexibilität, um Lösungen für Ihren Automatisierungsbedarf zu erstellen.

Mit der S7-1200 bewältigen Sie mühelos die anspruchsvollsten Aufgaben!

Die in der "Kompaktklasse" der Steuerungen angesiedelte Steuerungslösung SIMATIC S7-1200 besteht aus der Steuerung SIMATIC S7-1200 und den SIMATIC HMI Basic Panels. Beide Geräte sind mit der Engineering-Software TIA Portal programmierbar. Diese geräteübergreifende Programmierbarkeit bedeutet eine erhebliche Einsparung von Entwicklungskosten. Das TIA Portal umfasst STEP 7 für die Programmierung der S7-1200 und WinCC für den Entwurf von Projekten für Basic Panels.



Die kompakte Steuerung S7-1200 umfasst:

- Integriertes PROFINET
- Hochgeschwindigkeits-E/A für die Bewegungssteuerung, integrierte analoge Eingänge zur Minimierung des Platzbedarfs und des Nachrüstbedarfs an zusätzlichen E/A, 4 Impulsgeneratoren für Impulsfolge- und Impulsdaueranwendungen (Seite 77) und bis zu 6 schnelle Zähler
- In die CPU-Module integrierte E/A bieten 6 bis 14 Eingänge sowie 4 bis 10 Ausgänge.



Bereitstellung zusätzlicher E/A (Seite 18) durch Signalmodule für Gleichstrom-, Relais- oder analoge E/A sowie durch an der CPU-Vorderseite einzurastende innovative Signalboards.

Die SIMATIC HMI Basic Panels (Seite 23) wurden speziell für die S7-1200 konzipiert.

Dieses Easy Book bietet eine Einführung in die speicherprogrammierbare Steuerung S7-1200. Die folgenden Seiten enthalten einen Überblick über die zahlreichen Funktionen und Leistungsmerkmale der Geräte.

Beachten Sie für weitere Informationen das *S7-1200 Systemhandbuch*. Beachten Sie für weitere Informationen zu UL- und FM-Zulassung, CE-Kennzeichnung, C-Tick-Zulassung und anderen Normen die technischen Daten (Seite 311).

Dieses Handbuch beschreibt die folgenden Produkte:

- STEP 7 V13 SP1 Basic und Professional
- S7-1200 CPU Firmware-Version V4.0

Dokumentation und Information

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

- Das S7-1200 Systemhandbuch bietet spezielle Informationen zu Funktionsweise, Programmierung und technischen Daten der gesamten S7-1200-Produktfamilie. Neben dem Systemhandbuch bietet das S7-1200 Easy Book einen eher allgemeinen Überblick über die Fähigkeiten der S7-1200 Familie.

Sowohl das Systemhandbuch als auch das Easy Book stehen elektronisch (im PDF-Format) zur Verfügung. Die elektronischen Handbücher können von der Kundensupport-Website heruntergeladen werden. Sie finden sie außerdem auf den Datenträgern, die mit jeder S7-1200 CPU ausgeliefert werden.

- 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.
- Mit My Documentation Manager greifen Sie auf die elektronische Version (PDF) des SIMATIC-Dokumentationssatzes zu, der das Systemhandbuch, das Easy Book und Informationssystem von STEP 7 umfasst. Mit My Documentation Manager können Sie Themen aus verschiedenen Dokumenten per Drag & Drop selbst anordnen und so eigene benutzerspezifische Handbücher anlegen.

Im Kundensupport-Portal (<http://support.automation.siemens.com>) finden Sie einen Link auf My Documentation Manager unter mySupport.

- Die Kundensupport-Website bietet außerdem Podcasts, FAQs und weitere nützliche Dokumente für S7-1200 und STEP 7. Die Podcasts zeigen kurze Lernvideos zu speziellen Funktionen oder Szenarien, mit denen die Interaktionen, die praktische Bedienung und Effizienz von STEP 7 demonstriert werden. Die Podcast-Sammlung finden Sie auf folgenden Websites:
 - Webseite zu STEP 7 Basic (<http://www.automation.siemens.com/mcms/simatic-controller-software/de/step7/step7-basic/Seiten/Default.aspx>)
 - Webseite zu STEP 7 Professional (<http://www.automation.siemens.com/mcms/simatic-controller-software/de/step7/step7-professional/Seiten/Default.aspx>)
- Im technischen Forum unter Service & Support (<https://www.automation.siemens.com/WW/forum/guests/Conferences.aspx?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://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=257&Language=de&onlyInternet=False>)
 - Forum für STEP 7 Basic (<https://www.automation.siemens.com/WW/forum/guests/Conference.aspx?SortField=LastPostDate&SortOrder=Descending&ForumID=262&Language=de&onlyInternet=False>)

Service & Support

Zusätzlich zu unserem Dokumentations-Angebot bietet Siemens Ihnen im Internet technisches Know-how auf der Kundensupport-Website (<http://www.siemens.com/automation/>) 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.

Security-Hinweise

Siemens bietet Produkte und Lösungen mit Industrial Security-Funktionen an, die den sicheren Betrieb von Anlagen, Lösungen, Maschinen, Geräten und/oder Netzwerken unterstützen. Sie sind wichtige Komponenten in einem ganzheitlichen Industrial Security-Konzept. Die Produkte und Lösungen von Siemens werden unter diesem Gesichtspunkt ständig weiterentwickelt. Siemens empfiehlt, sich unbedingt regelmäßig über Produkt-Updates zu informieren.

Für den sicheren Betrieb von Produkten und Lösungen von Siemens ist es erforderlich, geeignete Schutzmaßnahmen (z. B. Zellschutzkonzept) zu ergreifen und jede Komponente in ein ganzheitliches Industrial Security-Konzept zu integrieren, das dem aktuellen Stand der Technik entspricht. Dabei sind auch eingesetzte Produkte von anderen Herstellern zu berücksichtigen. Weitergehende Informationen über Industrial Security finden Sie unter (<http://www.siemens.com/industrialsecurity>).

Um stets über Produkt-Updates informiert zu sein, melden Sie sich für unseren produktspezifischen Newsletter an. Weitere Informationen hierzu finden Sie unter (<http://support.automation.siemens.com>).

Inhaltsverzeichnis

	Vorwort	3
1	Einführung in die leistungsstarke und flexible S7-1200	15
1.1	Einführung in die S7-1200 SPS	15
1.2	Erweiterung der CPU-Funktionen	18
1.3	S7-1200 Module.....	21
1.4	Grundlegende HMI-Panels	23
1.5	Einbaumaße und notwendiger Freiraum	24
1.6	Neue Funktionen.....	29
2	STEP 7 vereinfacht Ihre Arbeit	33
2.1	Einfaches Einfügen von Anweisungen in Ihr Anwenderprogramm.....	34
2.2	Schneller Zugriff auf viel verwendete Operationen über die Funktionsleiste	35
2.3	Einfaches Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen	35
2.4	Erweiterbare Anweisungen	36
2.5	Einfaches Ändern des CPU-Betriebszustands	36
2.6	Ändern des Erscheinungsbilds und der Konfiguration von STEP 7	38
2.7	Projektbibliotheken und globale Bibliotheken für einfachen Zugriff	38
2.8	Einfache Auswahl einer Version einer Anweisung	39
2.9	Einfaches Drag & Drop zwischen Editoren	40
2.10	Aufruftyp eines DB ändern	41
2.11	Geräte vorübergehend vom Netzwerk trennen.....	42
2.12	Einfaches virtuelles "Abziehen" von Baugruppen ohne Verlust der Konfiguration	43
3	Erste Schritte	45
3.1	Projekt anlegen	45
3.2	Variablen für die E/A der CPU anlegen	46
3.3	Einfaches Netzwerk im Anwenderprogramm anlegen.....	48
3.4	Adressieren Sie die Anweisungen mithilfe der PLC-Variablen in der Variablentabelle.....	50
3.5	"Box"-Anweisung hinzufügen.....	51
3.6	Anweisung CALCULATE für komplexe mathematische Gleichungen verwenden.....	52
3.7	HMI-Gerät zum Projekt hinzufügen.....	54
3.8	Netzwerkverbindung zwischen CPU und HMI-Gerät herstellen	55
3.9	HMI-Verbindung zur gemeinsamen Nutzung von Variablen erstellen.....	55

3.10	HMI-Bild anlegen.....	56
3.11	PLC-Variable für das HMI-Element auswählen	57
4	SPS-Grundlagen leicht gemacht	59
4.1	Bei jedem Zyklus ausgeführte Arbeitsschritte.....	59
4.2	Betriebszustände der CPU.....	60
4.3	Ausführung des Anwenderprogramms	62
4.3.1	Bearbeitung des Zyklus im Betriebszustand RUN	62
4.3.2	OBs strukturieren Ihr Anwenderprogramm	63
4.3.3	Prioritäten und Warteschlange für die Ausführung von Ereignissen	63
4.4	Speicherbereiche, Adressierung und Datentypen	67
4.4.1	Von der S7-1200 unterstützte Datentypen.....	68
4.4.2	Adressierung der Speicherbereiche.....	71
4.4.3	Zugriff auf eine "Scheibe" eines Variablendatentyps	73
4.4.4	Zugriff auf eine Variable mit einer AT-Überlagerung	75
4.5	Impulsausgänge.....	77
5	Einfache Erstellung der Gerätekonfiguration	79
5.1	Konfiguration für eine nicht spezifizierte CPU erkennen	80
5.2	CPU zur Konfiguration hinzufügen.....	81
5.3	Ändern eines Geräts	82
5.4	Module zur Konfiguration hinzufügen	82
5.5	Konfigurieren des CPU- und Modulbetriebs	84
5.5.1	Systemmerker und Taktmerker bieten Standardfunktionen	86
5.6	Konfigurieren der IP-Adresse der CPU.....	89
5.7	Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach.....	91
5.7.1	Knowhow-Schutz	94
5.7.2	Kopierschutz	95
6	Programmierung leicht gemacht.....	97
6.1	Einfaches Entwerfen Ihres Anwenderprogramms	97
6.1.1	OBs für die Organisation Ihres Anwenderprogramms	99
6.1.2	FBs und FCs vereinfachen die Programmierung der modularen Aufgaben.....	101
6.1.3	Datenbausteine sorgen für die unkomplizierte Speicherung von Programmdateien	102
6.1.4	Neuen Codebaustein anlegen	103
6.1.5	Anlegen wiederverwendbarer Codebausteine	104
6.1.6	Einen Codebaustein aus einem anderen Codebaustein aufrufen	105
6.2	Anwenderfreundliche Programmiersprachen.....	105
6.2.1	Kontaktplan (KOP)	106
6.2.2	Funktionsplan (FUP)	107
6.2.3	Überblick über SCL	107
6.2.4	SCL-Programmiereditor	108
6.3	Leistungsstarke Anweisungen erleichtern die Programmierung.....	110
6.3.1	Bereitstellung der erwartbaren grundlegenden Operationen.....	110
6.3.2	Anweisungen für Vergleichen und Verschieben	113
6.3.3	Umwandlungsanweisungen	114

6.3.4	Mathematik ganz einfach mit der Anweisung Calculate	116
6.3.5	Zeiten	117
6.3.6	Zähler	122
6.3.7	Impulsdauermodulation (PWM)	125
6.4	Einfache Erstellung von Datenprotokollen	126
6.5	Einfaches Überwachen und Testen Ihres Anwenderprogramms	128
6.5.1	Beobachtungstabellen und Forcetabellen	128
6.5.2	Querverweis zum Anzeigen der Verwendung	129
6.5.3	Aufrufstruktur zur Prüfung der Aufrufhierarchie	130
6.5.4	Diagnoseanweisungen zur Überwachung der Hardware	131
6.5.4.1	Zustände der LEDs der CPU lesen	131
6.5.4.2	Anweisungen zum Lesen des Diagnosezustands der Geräte	132
6.6	Schneller Zähler (HSC)	133
6.6.1	Funktionsweise von schnellen Zählern	134
6.6.2	Konfiguration eines schnellen Zählers	141
7	Einfache Kommunikation zwischen Geräten	143
7.1	Netzwerkverbindung erstellen	144
7.2	Kommunikationsoptionen	145
7.3	Anzahl der asynchronen Kommunikationsverbindungen	147
7.4	PROFINET- und PROFIBUS-Anweisungen	147
7.5	PROFINET	149
7.5.1	Offene Benutzerkommunikation	149
7.5.1.1	Ad-hoc-Modus	150
7.5.1.2	Verbindungs-IDs für Anweisungen für die offene Benutzerkommunikation	150
7.5.1.3	Parameter für die PROFINET-Verbindung	153
7.5.2	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren	156
7.6	PROFIBUS	158
7.6.1	Kommunikationsdienste der PROFIBUS-CMs	160
7.6.2	Verweis auf die Benutzerhandbücher für PROFIBUS-CMs	161
7.6.3	CM 1243-5 (DP-Master) und DP-Slave hinzufügen	161
7.6.4	PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen	162
7.7	ASi	164
7.7.1	AS-i-Mastermodul CM 1243-2 und AS-i-Slave hinzufügen	165
7.7.2	Einem AS-i-Slave eine AS-i-Adresse zuweisen	166
7.8	S7-Kommunikation	169
7.8.1	Anweisungen GET und PUT	169
7.8.2	S7-Verbindung erstellen	170
7.8.3	Parametrieren der GET/PUT-Verbindung	171
7.9	GPRS	172
7.9.1	Anschluss an ein GSM-Netz	172
7.10	Kommunikationsprotokolle PtP, USS und Modbus	179
7.10.1	Punkt-zu-Punkt-Kommunikation	179
7.10.2	Mit den seriellen Kommunikationsschnittstellen arbeiten	179
7.10.3	PtP-Operationen	180
7.10.4	USS-Anweisungen	181

7.10.5	Modbus-Anweisungen.....	183
8	Einfache PID-Regelung.....	185
8.1	Anweisung PID und Technologieobjekt einfügen	187
8.2	Operation PID_Compact	189
8.3	ErrorBit-Parameter der Anweisung PID_Compact.....	193
8.4	Anweisung PID_3Step	195
8.5	ErrorBit-Parameter der Anweisung PID_3Step.....	202
8.6	PID-Regler konfigurieren.....	204
8.7	Inbetriebnahme des PID-Reglers.....	207
9	Webserver für einfachen Internetanschluss.....	209
9.1	Einfache Nutzung der Standard-Webseiten.....	210
9.2	Bedingungen, die sich auf die Nutzung des Webserver auswirken können	213
9.2.1	Funktionseinschränkungen bei Deaktivierung von JavaScript in den Internetoptionen	214
9.2.2	Eingeschränkte Funktionen, wenn Cookies in den Internetoptionen nicht erlaubt sind	215
9.3	Benutzerdefinierte Webseiten einfach anlegen	216
9.3.1	Eigene benutzerdefinierte Webseiten einfach anlegen	216
9.3.2	Einschränkungen bei benutzerdefinierten Webseiten	217
9.3.3	Konfiguration einer benutzerdefinierten Webseite	218
9.3.4	WWW-Anweisung verwenden	219
10	Einfache Bewegungssteuerung.....	221
10.1	Phasenlage	226
10.2	Konfigurieren eines Impulsgenerators	229
10.3	Achse konfigurieren	230
10.4	TO-Befehlstabelle PTO konfigurieren	234
10.5	Bewegungssteuerungsanweisungen	237
10.5.1	Übersicht MC-Anweisungen.....	237
10.5.2	Anweisung MC_Power (Achse freigeben/sperrern).....	238
10.5.3	Anweisung MC_Reset (Fehler bestätigen)	241
10.5.4	Anweisung MC_Home (Referenzpunktfahrt der Achse durchführen).....	242
10.5.5	Anweisung MC_Halt (Achse pausieren)	245
10.5.6	Anweisung MC_MoveAbsolute (Achse absolut positionieren).....	247
10.5.7	Anweisung MC_MoveRelative (Achse relativ positionieren).....	249
10.5.8	Anweisung MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen)	251
10.5.9	Anweisung MC_MoveJog (Achse im Tipbetrieb bewegen)	254
10.5.10	Anweisung MC_CommandTable (Achssteuersbefehle als Bewegungsfolge ausführen)	257
10.5.11	Anweisung MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern)	260
10.5.12	Anweisung MC_WriteParam (Parameter des Technologieobjekts schreiben)	262
10.5.13	Anweisung MC_ReadParam (Parameter eines Technologieobjekts lesen)	264
10.6	Funktionsweise der Bewegungssteuerung bei der S7-1200	266
10.6.1	Für die Bewegungssteuerung verwendete CPU-Ausgänge	266
10.6.2	Hardware- und Software-Endschalter für die Bewegungssteuerung.....	268
10.6.3	Referenzpunktfahrt.....	271

10.6.3.1	Referenzpunktfahrt für Achsen durchführen	271
10.6.3.2	Konfiguration der Parameter für die Referenzpunktfahrt	273
10.6.3.3	Abfolge bei der aktiven Referenzpunktfahrt	275
10.7	Inbetriebnahme	277
11	Einfaches Arbeiten mit den Online-Tools	285
11.1	Online-Verbindung mit einer CPU herstellen	285
11.2	Interaktion mit der Online-CPU	286
11.3	Online gehen, um die Werte in der CPU zu beobachten	287
11.4	Einfaches Anzeigen des Status des Anwenderprogramms	288
11.5	Beobachtungstabelle zur Überwachung der CPU verwenden	288
11.6	Arbeiten mit der Forcetable	290
11.7	Online-Werte eines DBs erfassen, um die Startwerte zurückzusetzen	293
11.8	Elemente des Projekts kopieren	294
11.9	Vergleichen von Offline- und Online-CPU's	295
11.10	Diagnoseereignisse anzeigen	296
11.11	Einstellen der IP-Adresse und der Uhrzeit	296
11.12	Auf Werkseinstellungen zurücksetzen	297
11.13	Firmware aktualisieren	298
11.14	Laden einer permanenten IP-Adresse in eine Online-CPU	299
11.15	Verwendung der "nicht spezifizierten CPU" zum Laden der Hardwarekonfiguration aus dem Gerät	299
11.16	Laden im Betriebszustand RUN	301
11.16.1	Ändern des Programms im Betriebszustand RUN	303
11.17	CPU-Daten bei Auslösebedingungen verfolgen und aufzeichnen	304
12	IO-Link – ganz einfach	305
12.1	Überblick über die IO-Link-Technologie	305
12.2	Komponenten eines IO-Link-Systems	305
12.3	Nach dem Anlauf	305
12.4	IO-Link-Protokoll	306
12.5	Konfiguration des Feldbusses	306
12.6	IO-Link und Ihr STEP 7-Programm	306
12.7	Der SM 1278 4xIO-Link-Master	307
A	Technische Daten	311
A.1	Allgemeine technische Daten	311
A.2	CPU-Module	319
A.3	Digitale Erweiterungsmodule	323
A.3.1	SB 1221, SB 1222 und SB 1223 Digitaleingabe/-ausgabe (DI, DO und DI/DO)	323

A.3.2	SM 1221 Digitaleingabe (DI)	326
A.3.3	SM 1222 Digitalausgabe (DO)	327
A.3.4	SM 1223 V-DC-Digitaleingabe/-ausgabe (DI/DO)	329
A.3.5	SM 1223 120/230 V-AC-Eingabe/Relaisausgang	330
A.4	Technische Daten der digitalen Eingänge und Ausgänge	331
A.4.1	24-V-DC-Digitaleingabe (DI)	331
A.4.2	120/230 V-AC-Digitaleingabe	333
A.4.3	Digitalausgänge (DO)	334
A.5	Analoge Erweiterungsmodule	336
A.5.1	SB 1231 und SB 1232 Analogeingabe (AI) und Analogausgabe (AO)	336
A.5.2	SM 1231 Analogeingabe (AI)	337
A.5.3	SM 1232 Analogausgabe (AO)	338
A.5.4	SM 1234 Analogeingabe/-ausgabe (AI/AO)	338
A.5.5	Schaltpläne SM 1231 (AI), SM 1232 (AO) und SM 1234 (AI/AO)	339
A.6	BB 1297 Batterieboard	340
A.7	Technische Daten der analogen E/A	341
A.7.1	Technische Daten der analogen Eingänge (CPU, SM und SB)	341
A.7.2	Eingangsmessbereiche (AI) für Spannung und Strom	342
A.7.3	Schrittantwort der analogen Eingänge (AI)	343
A.7.4	Abtastzeit und Aktualisierungszeiten der Analogeingänge	344
A.7.5	Technische Daten der analogen Ausgänge	345
A.7.6	Ausgangsmessbereiche (AO) für Spannung und Strom	346
A.8	RTD- und Thermoelementmodule	347
A.8.1	Technische Daten SB 1231 RTD und SB 1231 TC	348
A.8.2	Technische Daten SM 1231 RTD	350
A.8.3	Technische Daten SM 1231 TC	352
A.8.4	Technische Daten der analogen Eingänge für RTD und TC (SM und SB)	353
A.8.5	Thermoelementtyp	354
A.8.6	Filterauswahl und Aktualisierungszeiten beim Thermoelement	355
A.8.7	Auswahltabelle RTD-Gebertyp	356
A.8.8	Filterauswahl und Aktualisierungszeiten beim RTD	357
A.9	Kommunikationsschnittstellen	358
A.9.1	PROFIBUS-Master/Slave	358
A.9.1.1	CM 1242-5 PROFIBUS-Slave	358
A.9.1.2	CM 1243-5 PROFIBUS-Master	359
A.9.2	GPRS CP	361
A.9.2.1	Technische Daten des CP 1242-7 GPRS	361
A.9.3	Teleservice (TS)	364
A.9.4	RS485-, RS232- und RS422-Kommunikation	365
A.9.4.1	Technische Daten des CM 1241 RS485	365
A.9.4.2	Technische Daten des CM 1241 RS422/485	367
A.9.4.3	Technische Daten des CM 1241 RS232	368
A.10	Technologiemodule	370
A.10.1	SM 1278 4xIO-Link-Master SM	370
A.10.1.1	Technische Daten des Signalmoduls SM 1278 4xIO-Link-Master	370
A.10.1.2	SM 1278 4xIO-Link-Master SM, Schaltpläne	373
A.11	Zugehörige Produkte	374
A.11.1	PM 1207 Stromversorgungsmodul	374

A.11.2	CSM 1277 Compact Switch Module	374
A.11.3	CM CANopen-Modul	375
B	Austauschen einer V3.0-CPU durch eine V4.0-CPU.....	377
B.1	Austauschen einer V3.0-CPU durch eine V4.0-CPU	377
	Index.....	383

Einführung in die leistungsstarke und flexible S7-1200

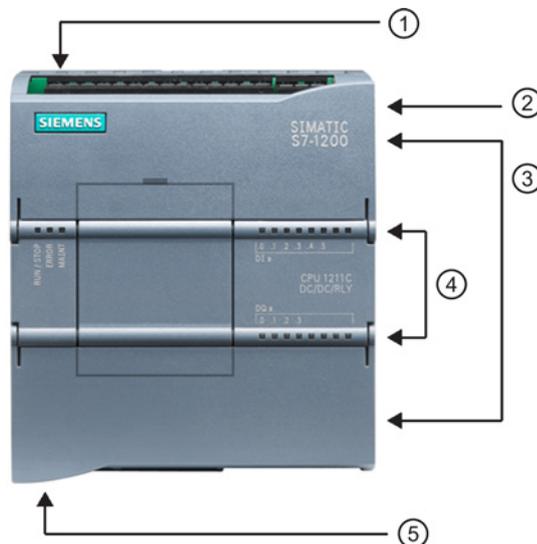
1

1.1 Einführung in die S7-1200 SPS

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

Die CPU umfasst einen Mikroprozessor, eine integrierte Spannungsversorgung, Eingangskreise und Ausgangskreise, integriertes PROFINET, Peripherie zur Bewegungssteuerung in Hochgeschwindigkeit sowie integrierte Analogeingänge in einem kompakten Gehäuse und bildet somit eine leistungsstarke Steuerung. 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 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 PROFIBUS-, GPRS-, RS485- oder RS232-Netzwerke stehen weitere Module zur Verfügung.



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

1.1 Einführung in die S7-1200 SPS

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

- Jede CPU ist mit einem Passwortschutz (Seite 91) ausgestattet, mit dem der Zugriff auf die CPU-Funktionen nach Bedarf eingerichtet werden kann.
- Sie können mit dem Knowhow-Schutz (Seite 94) den Code in einem bestimmten Baustein verbergen.
- Mit dem Kopierschutz (Seite 95) können Sie Ihr Programm mit einer bestimmten Memory Card oder CPU verknüpfen.

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	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Anwenderspeicher	Arbeitsspeicher	30 KB	50 KB	75 KB	100 KB	125 KB
	Ladespeicher	1 MB	1 MB	4 MB	4 MB	4 MB
	Remanent	10 KB	10 KB	10 KB	10 KB	10 KB
Integrierte E/A	Digital	6 Eingänge/4 Ausgänge	8 Eingänge/6 Ausgänge	14 Eingänge/10 Ausgänge	14 Eingänge/10 Ausgänge	14 Eingänge/10 Ausgänge
	Analog	2 Eingänge	2 Eingänge	2 Eingänge	2 Eingänge/2 Ausgänge	2 Eingänge/2 Ausgänge
Größe des Prozessabbilds	Eingänge (E)	1024 Byte	1024 Byte	1024 Byte	1024 Byte	1024 Byte
	Ausgang (A)	1024 Byte	1024 Byte	1024 Byte	1024 Byte	1024 Byte
Merker (M)		4096 Byte	4096 Byte	8192 Byte	8192 Byte	8192 Byte
Erweiterung: Signalmodul (SM)		Keine	2	8	8	8
Signalboard (SB), Batterieboard (BB) oder Kommunikationsboard (CB)		1	1	1	1	1
Kommunikationsmodule (CM) (Anbau links)		3	3	3	3	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	Ea.0 bis Ea.5	Ea.0 bis Ea.5	Ea.0 bis Ea.5	Ea.0 bis Ea.5
	30/120 kHz	--	Ea.6 bis Ea.7	Ea.6 bis Eb.5	Ea.6 bis Eb.5	Ea.6 bis Eb.1
Impulsausgänge ^{e 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.0 bis Aa.3	Aa.0 bis Aa.3	Aa.0 bis Aa.3	Aa.4 bis Ab.1
	20 kHz	--	Aa.4 bis Aa.5	Aa.4 bis Ab.1	Aa.4 bis Ab.1	--
Memory Card		SIMATIC Memory Card (optional)				
Pufferung Echtzeituhr		20 Tage, typ./min. 12 Tage bei 40 Grad C (wartungsfreier Hochleistungskondensator)				
PROFINET Ethernet-Kommunikationsport		1	1	1	2	2

Merkmal	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
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.

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 311).

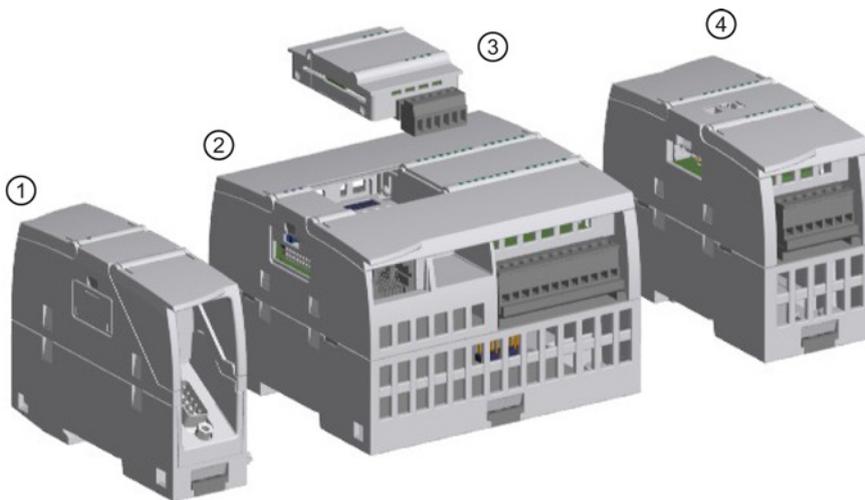
Tabelle 1-2 Von der S7-1200 unterstützte Bausteine, Zeiten und Zähler

Element	Beschreibung	
Bausteine	Typ	OB, FB, FC, DB
	Größe	30 KB (CPU 1211C) 50 KB (CPU 1212C) 64 KB (CPU 1214C, CPU 1215C und CPU 1217C)
	Anzahl	Bis 1024 Bausteine gesamt (OBs + FBs + FCs + DBs)
	Schachtelungstiefe	16 aus Zyklus- oder Anlauf-OBs 6 aus beliebigen Alarm-OBs
	Überwachung	Der Zustand von 2 Codebausteinen kann gleichzeitig überwacht werden.
	OBs	Programmzyklus
Anlauf		Multi
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
Tageszeit		Multi
Zustand		1
Update		1
Profil		1
Zeiten		Typ
	Anzahl	Nur durch die Speicherkapazität begrenzt
	Speicherung	Struktur im DB, 16 Bytes pro Zeit

Element		Beschreibung
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

1.2 Erweiterung der CPU-Funktionen

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 311).



- ① Kommunikationsmodul (CM) oder Kommunikationsprozessor (CP)
- ② CPU
- ③ Signalboard (SB), Kommunikationsboard (CB) oder Batterieboard (BB)
- ④ Signalmodul (SM)

Tabelle 1-3 Digitale Signalmodule und Signalboards

Typ	Nur Eingang	Nur Ausgang	Ein-/Ausgang kombiniert
③ digitales SB	<ul style="list-style-type: none"> • 4 x 24-V-DC-Eingänge, 200 kHz • 4 x 5-V-DC-Eingänge, 200 kHz 	<ul style="list-style-type: none"> • 4 x 24-V-DC-Ausgänge, 200 kHz • 4 x 5-V-DC-Ausgänge, 200 kHz 	<ul style="list-style-type: none"> • 2 x 24-V-DC-Eingänge/2 x 24-V-DC-Ausgänge • 2 x 24-V-DC-Eingänge/2 x 24-V-DC-Ausgänge 200 kHz • 2 x 5-V-DC-Eingänge/2 x 5-V-DC-Ausgänge 200 kHz
④ digitales SM	<ul style="list-style-type: none"> • 8 x 24-V-DC-Eingänge 	<ul style="list-style-type: none"> • 8 x 24-V-DC-Ausgänge • 8 x Relaisausgänge • 8 x Relaisausgänge (Umschaltung) 	<ul style="list-style-type: none"> • 8 x 24-V-DC-Eingänge/8 x 24-V-DC-Ausgänge • 8 x 24-V-DC-Eingänge/8 x Relaisausgänge • 8 x 120/230-V-AC-Eingänge/8 x Relaisausgänge
	<ul style="list-style-type: none"> • 16 x 24-V-DC-Eingänge 	<ul style="list-style-type: none"> • 16 x 24-V-DC-Ausgänge • 16 x Relaisausgänge 	<ul style="list-style-type: none"> • 16 x 24-V-DC-Eingänge/16 x 24-V-DC-Ausgänge • 16 x 24-V-DC-Eingänge/16 x Relaisausgänge

Tabelle 1-4 Analoge Signalmodule und Signalboards

Typ	Nur Eingang	Nur Ausgang	Ein-/Ausgang kombiniert
③ analoges SB	<ul style="list-style-type: none"> • 1 x 12-Bit-Analogeingang • 1 x 16-Bit-RTD • 1 x 16-Bit-Thermoelement 	<ul style="list-style-type: none"> • 1 x Analogausgang 	-
④ analoges SM	<ul style="list-style-type: none"> • 4 x Analogeingänge • 4 x Analogeingänge x 16 Bit • 8 x Analogeingänge • Thermoelement: <ul style="list-style-type: none"> – 4 x 16-Bit-TC – 8 x 16-Bit-TC • RTD: <ul style="list-style-type: none"> – 4 x 16-Bit-RTD – 8 x 16-Bit-RTD 	<ul style="list-style-type: none"> • 2 x Analogausgänge • 4 x Analogausgänge 	<ul style="list-style-type: none"> • 4 x Analogeingänge/2 x Analogausgänge

Tabelle 1-5 Kommunikationsschnittstellen

Modul	Typ	Beschreibung
① Kommunikationsmodul (CM)	RS232	Vollduplex
	RS422/485	Vollduplex (RS422) Halbduplex (RS485)
	PROFIBUS-Master	DPV1
	PROFIBUS-Slave	DPV1
	AS-i-Master (CM 1243-2)	AS-Interface
① Kommunikationsprozessor (CP)	Modem anschließbar	GPRS
③ Kommunikationsboard (CB)	RS485	Halbduplex
TeleService ¹	TS-Adapter IE Basic	Anschluss an CPU
	TS-Adapter GSM	GSM/GPRS
	TS-Adapter Modem	Modem
	TS-Adapter ISDN	ISDN
	TS-Adapter RS232	RS232

¹ Mit dem TS-Adapter IE Basic können Sie über ein Ethernet-Kabel verschiedene Kommunikationsschnittstellen an den PROFINET-Port der CPU anschließen. An den TS-Adapter IE Basic können Sie bis zu drei TS-Adaptermodule anschließen.

Tabelle 1-6 Technologiemodule

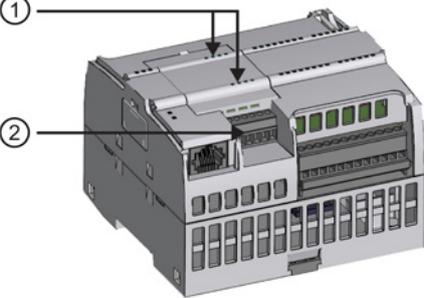
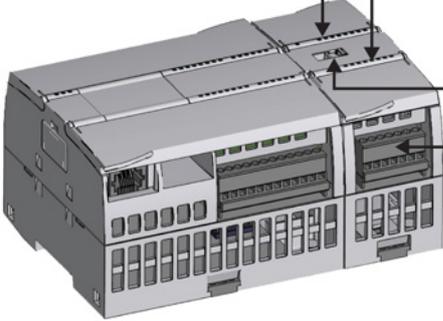
Modul	Typ	Beschreibung
④ IO-Link	SM 1278 4xIO-Link-Master	Unterstützt 4 IO-Link-Slaves

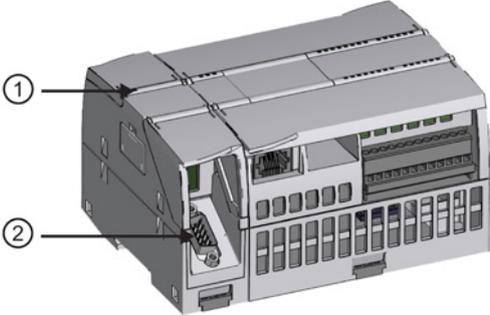
Tabelle 1-7 Andere Boards

Modul	Beschreibung
③ Batterieboard	Wird in die Schnittstelle für Erweiterungsboards auf der Vorderseite der CPU gesteckt. Bietet langfristige Pufferung der Echtzeituhr.

1.3 S7-1200 Module

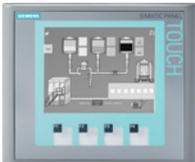
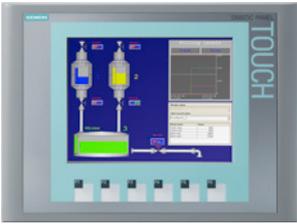
Tabelle 1- 8 S7-1200 Erweiterungsmodule

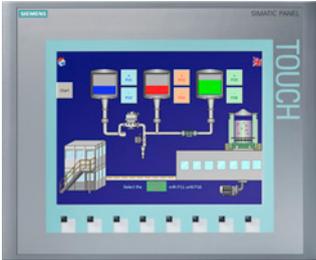
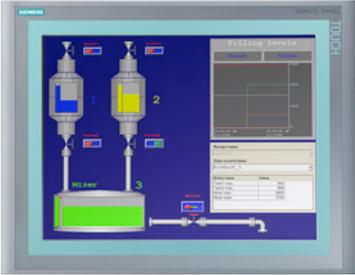
Art des Moduls	Beschreibung	
<p>Die CPU unterstützt ein steckbares Erweiterungsboard:</p> <ul style="list-style-type: none"> • Ein Signalboard (SB) stellt zusätzliche E/A für Ihre CPU bereit. Das SB wird auf der Vorderseite der CPU angeschlossen. • Mit einem Kommunikationsboard (CB) können Sie Ihre CPU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

Art des Moduls	Beschreibung					
<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. Ein CP bietet Möglichkeiten für andere Arten der Kommunikation, z. B. für den Anschluss der CPU über ein GPRS-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). 		<table border="1"> <tr> <td data-bbox="1091 344 1150 389">①</td> <td data-bbox="1155 344 1439 389">Status-LEDs</td> </tr> <tr> <td data-bbox="1091 396 1150 441">②</td> <td data-bbox="1155 396 1439 441">Kommunikationsstecker</td> </tr> </table>	①	Status-LEDs	②	Kommunikationsstecker
①	Status-LEDs					
②	Kommunikationsstecker					

1.4 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.

Grundlegendes HMI-Panel	Beschreibung	Technische Daten
 <p>KP 300 Basic PN</p>	<p>3,6"-Membrantastatur mit 10 frei konfigurierbaren taktilen Tasten</p> <ul style="list-style-type: none"> • Mono (STN, schwarz/weiß) • 87 mm x 31 mm (3,6") • Programmierte Hintergrundfarbe (weiß, grün, gelb oder rot) • Auflösung: 240 x 80 	<ul style="list-style-type: none"> • 250 Variablen • 50 Prozessbilder • 200 Meldungen • 25 Kurven • 40-KB-Rezeptspeicher • 5 Rezepte, 20 Datensätze, 20 Einträge
 <p>KTP 400 Basic PN</p>	<p>4"-Touchscreen mit 4 taktilen Tasten</p> <ul style="list-style-type: none"> • Mono (STN, Graustufen) • 76,79 mm x 57,59 mm (3,8") Hoch- oder Querformat • Auflösung: 320 x 240 	<ul style="list-style-type: none"> • 250 Variablen • 50 Prozessbilder • 200 Meldungen • 25 Kurven • 40-KB-Rezeptspeicher • 5 Rezepte, 20 Datensätze, 20 Einträge
 <p>KTP 600 Basic PN</p>	<p>6"-Touchscreen mit 6 taktilen Tasten</p> <ul style="list-style-type: none"> • Farbe (TFT, 256 Farben) oder Mono (STN, Graustufen) • 115,2 mm x 86,4 mm (5,7") Hoch- oder Querformat • Auflösung: 320 x 240 	<ul style="list-style-type: none"> • 500 Variablen • 50 Prozessbilder • 200 Meldungen • 25 Kurven • 40-KB-Rezeptspeicher • 5 Rezepte, 20 Datensätze, 20 Einträge

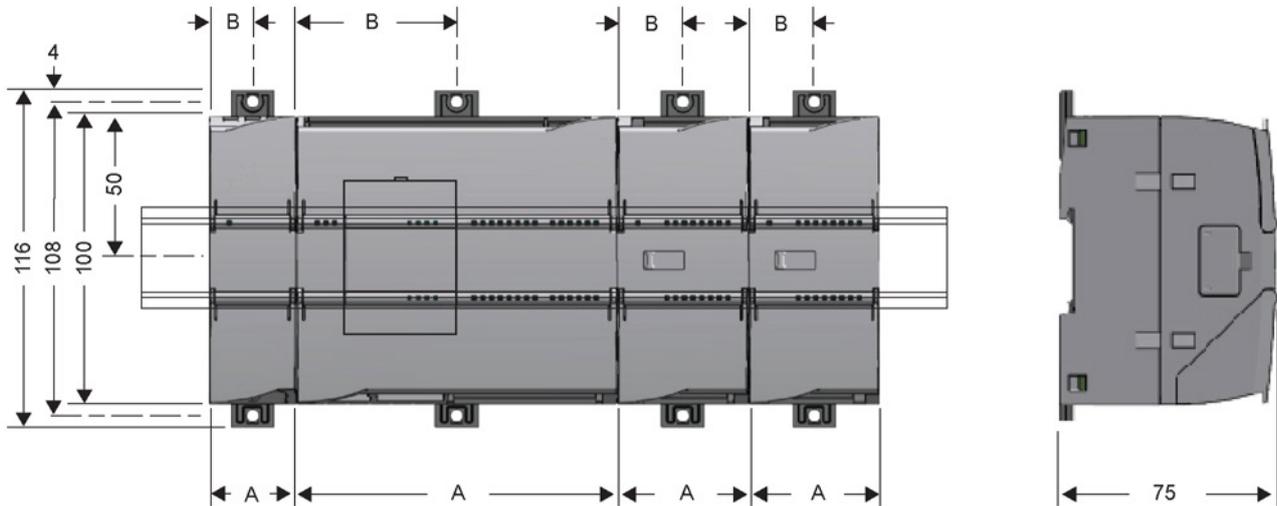
Grundlegendes HMI-Panel	Beschreibung	Technische Daten
 <p>KTP 1000 Basic PN</p>	<p>10"-Touchscreen mit 8 taktilem Tasten</p> <ul style="list-style-type: none"> • Farbe (TFT, 256 Farben) • 211,2 mm x 158,4 mm (10,4") • Auflösung: 640 x 480 	<ul style="list-style-type: none"> • 500 Variablen • 50 Prozessbilder • 200 Meldungen • 25 Kurven • 40-KB-Rezeptspeicher • 5 Rezepte, 20 Datensätze, 20 Einträge
 <p>TP 1500 Basic PN</p>	<p>15"-Touchscreen</p> <ul style="list-style-type: none"> • Farbe (TFT, 256 Farben) • 304,1 mm x 228,1 mm (15,1") • Auflösung: 1024 x 768 	<ul style="list-style-type: none"> • 500 Variablen • 50 Prozessbilder • 200 Meldungen • 25 Kurven • 40-KB-Rezeptspeicher (integrierter Flash) • 5 Rezepte, 20 Datensätze, 20 Einträge

1.5 Einbaumaße und notwendiger Freiraum

Die Steuerung S7-1200 wurde so ausgelegt, dass sie einfach einzubauen ist. Beim Schaltschrankbau ebenso wie bei der Montage auf einer Standard-Hutschiene gestattet die kompakte Baugröße eine optimale Platzausnutzung.

Spezielle Informationen zu den Einbauvoraussetzungen und weitere Hinweise zum Einbau finden Sie im *S7-1200 Systemhandbuch*.

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



CPU 1215C, CPU 1217C

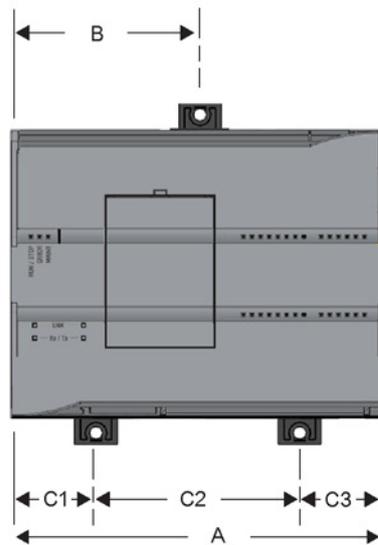


Tabelle 1-9 Abmessungen für die Montage (mm)

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

1.5 Einbaumaße und notwendiger Freiraum

S71200 Geräte		Breite A (mm)	Breite B (mm)	Breite C (mm)
	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	--
Kommunikationsschnittstellen	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	30	15	--
	TS (TeleService)-Adapter IE Basic ¹			
	TS-Adapter	30	15	--
	TS-Modul	30	15	--

¹ Bevor Sie den TS (TeleService)-Adapter 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.

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 S71200 kann horizontal oder vertikal eingebaut werden. Die kompakte Größe der S7-1200 macht eine effiziente Platzausnutzung möglich.

 **WARNUNG**

Einbaurichtlinien für S71200 PLCs

Bei den SIMATIC S7-1200 Automatisierungssystemen handelt es sich um offene Steuerungen. 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.

Die Nichteinhaltung dieser Einbaubedingungen kann zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Beachten Sie diese Maßnahmen beim Einbau der S7-1200 Automatisierungssysteme.

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.

Berücksichtigen Sie auch, wie Sie die Verdrahtung der Geräte in der Schalttafel verlegen. Vermeiden Sie es, Niederspannungssignalleitungen und Kommunikationskabel in der gleichen Kabelbahn wie AC-Versorgungsleitungen und schnellschaltende Hochgeschwindigkeits-DC-Leitungen 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**

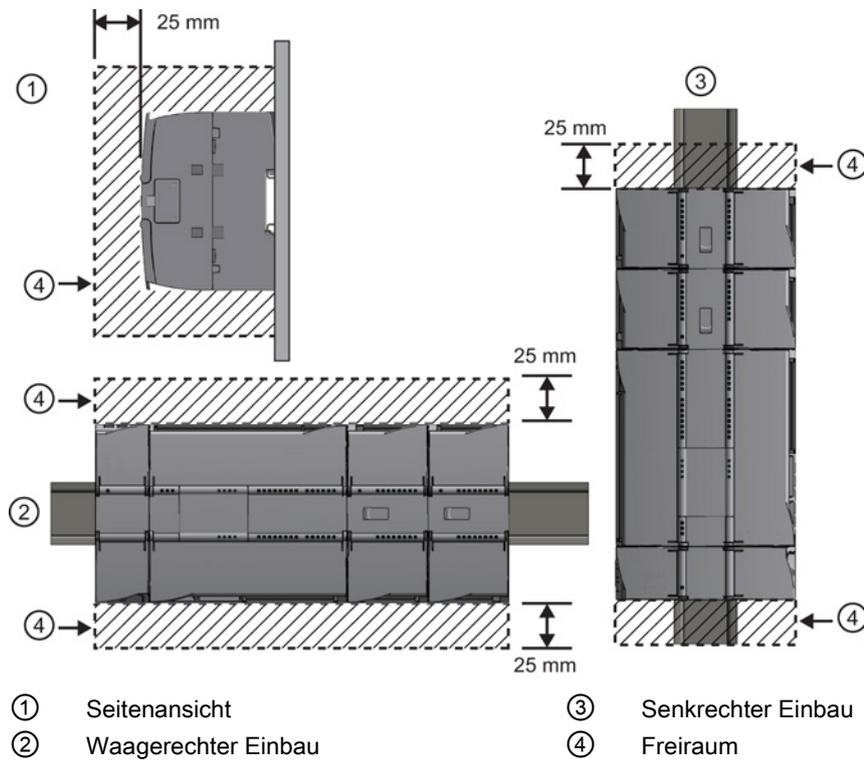
Bei vertikalem Einbau ist die maximal zulässige Umgebungstemperatur um 10 °C niedriger.

Richten Sie ein vertikal eingebautes S71200 System wie in der folgenden Abbildung gezeigt aus.

Stellen Sie sicher, dass das S7-1200 System richtig montiert wird.

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

1.5 Einbaumaße und notwendiger Freiraum



⚠️ 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.

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 tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Treffen Sie alle notwendigen Sicherheitsvorkehrungen und vergewissern Sie sich, dass vor dem Einbau bzw. Ausbau eines Geräts die Spannungsversorgung der S7-1200 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**

Korrektur Einbau der S7-1200 Module

Falscher Einbau eines S7-1200 Moduls kann zu unvorhersehbarem Verhalten des Programms der S7-1200 führen.

Wird ein S7-1200 Gerät durch eine andere Variante ersetzt, nicht richtig ausgerichtet oder in der falschen Reihenfolge eingebaut, so kann dies aufgrund von unerwartetem Betrieb der Geräte zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

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.

1.6 Neue Funktionen

Folgende Funktionen sind in dieser Version neu hinzugekommen:

- Die S7-1200 unterstützt neue Organisationsbausteine (OBs) (Seite 99) mit unterschiedlichen Prioritätsstufen und Alarmen.
- Der Webserver (Seite 209) unterstützt jetzt die Anzeige von Standard-Webseiten und benutzerdefinierten Webseiten auf einem Mobilgerät ebenso wie auf einem PC. Die Standard-Webseiten sind in dieser Version in Englisch, Deutsch, Französisch, Spanisch, Italienisch und vereinfachtem Chinesisch verfügbar.
- Die Funktion "Laden in Run" (Seite 301) unterstützt jetzt maximal zwanzig Bausteine, die Sie im Betriebszustand RUN ins Zielsystem laden können. Sie können auch Variablen hinzufügen und Variablen in bestehenden Datenbausteinen und Funktionsbausteinen ändern und die geänderten Datenbausteine im Betriebszustand RUN ins Zielsystem laden.
- Die Online- und Diagnose-Tools von STEP 7 bieten die Möglichkeit, ein Firmware-Update (Seite 298) Ihrer CPU, Signalmodule, Kommunikationsmodule und von angeschlossenen Signal- oder Kommunikationsboards durchzuführen.
- STEP 7 umfasst eine Nachverfolgungs- und Logikanalysefunktion (Seite 304), die Sie mit den S7-1200 CPUs der Version 4.0 einsetzen können. Mit dieser Funktion können Sie spezifische Daten konfigurieren, die Sie nachverfolgen und aufzeichnen möchten, wenn die CPU eine von Ihnen definierte Auslösebedingung erfüllt. Die CPU speichert die aufgezeichneten Daten und STEP 7 bietet Tools zum Abrufen und Analysieren der aufgezeichneten Daten.

- Neue Programmieranweisungen:
 - **Variable bei Signalfanke setzen:** R_TRIG, F_TRIG
 - **Ortszeit schreiben:** WR_LOC_T
 - **Maximale Zeichenkettenlänge:** MAX_LEN
 - **Uhrzeitalarme:** SET_TINTL, CAN_TINT, ACT_TINT, QRY_TINT
 - **Prozessrezepte:** RecipeExport, RecipeImport
 - **Adressbearbeitung:** LOG2GEO, RD_ADDR
 - **Bewegungssteuerung:** MC_WriteParam, MC_ReadParam
 - **Passwort aktivieren/deaktivieren:** ENDIS_PW
- HSC-Verbesserungen, die bewirken, dass jeder Eingang oder Ausgang einer HSC-Anweisung jedem integrierten oder SB-Digitaleingang zugewiesen werden kann
- PTO/PWM-Verbesserungen, die bewirken, dass jeder Eingang oder Ausgang einer PTO/PWM-Anweisung jedem integrierten oder SB-Digitalausgang zugewiesen werden kann
- Erweiterte Bibliotheksfunktionen einschließlich Versionierung

Neue Module für die S7-1200

Neue Module erweitern die Leistungsfähigkeit der S7-1200 CPU und bieten Ihnen die erforderliche Flexibilität für Ihren Automatisierungsbedarf:

- Neue CPU 1217C DC/DC/DC mit schnellen Differenz-E/A
- Neu und verbesserte S7-1200 Signalmodule. Die neuen Signalmodule (6ES7 2xx-xxx32-0XB0) ersetzen die bisherigen Signalmodule (6ES7 2xx-xxx30-0XB0). Die neuen Module bieten:
 - Analogeingabe- und -ausgabebaugruppen jetzt mit Stromsignal im Bereich 4-20 mA
 - Analogeingabebaugruppen bei Einsatz im Bereich 4-20 mA jetzt mit Drahtbruchererkennung
 - Die Codierung der Steckverbinder verhindert Anschlussfehler beim Anschluss von Feldverdrahtungssteckern, die jetzt für die Module mit Relaisausgängen verfügbar sind
 - Ersatzteilkompatibilität: Die überarbeiteten Module können ohne Änderungen anstelle der bisherigen Module verwendet werden
- Neue Ersatzteile für die S7-1200 CPUs
- Neuer CPU 1217C Eingangssimulator (6ES7 274-1XK30-0XA0)
- Der neue SM 1278 4xIO-Link-Master (6ES7 278-4BD32-0XB0) funktioniert sowohl als Signalmodul als auch als Kommunikationsmodul und ermöglicht den Anschluss von bis zu 4 IO-Link-Slaves (3-Leiter-Anschluss) oder 4 Standardaktoren oder Standarddrehgebern

- Neues S7-1200 Potentiometermodul (6ES7 274-1XA30-0XA0)
- Das neue CM CANopen für die S7-1200 ist ein steckbares Modul, mit dem Sie CANopen-Geräte mit einem S7-1200 PLC verbinden können. Das Modul kann als Master oder als Slave konfiguriert werden.

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

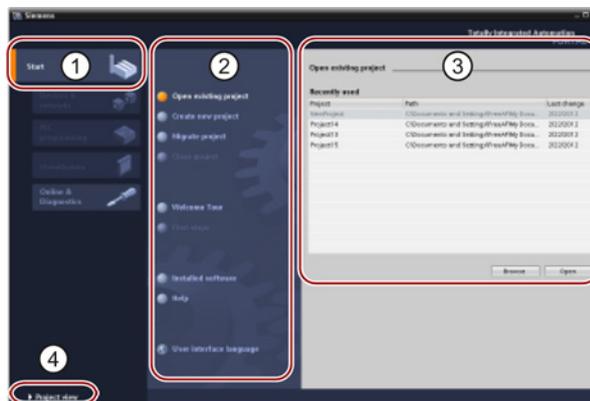
Wenn Sie eine S7-1200 V3.0-CPU durch eine S7-1200 V4.0-CPU austauschen, beachten Sie die dokumentierten Unterschiede zwischen den Versionen.

Siehe auch

Grundlegende HMI-Panels (Seite 23)

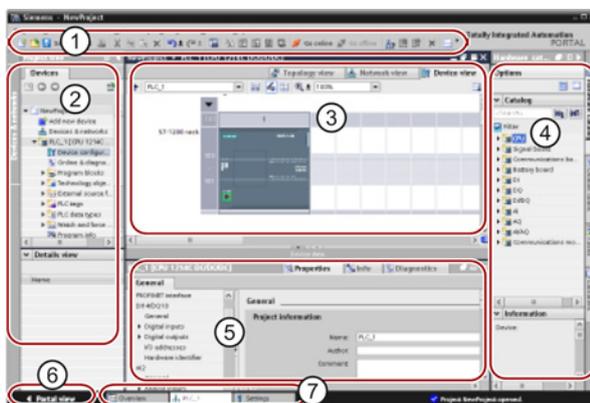
STEP 7 vereinfacht Ihre Arbeit

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 Mausclick 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

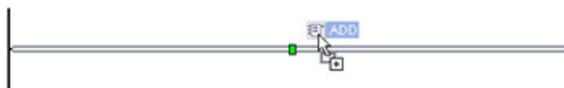
2.1 Einfaches Einfügen von Anweisungen in Ihr Anwenderprogramm

Da sich alle Komponenten an einer Stelle befinden, haben Sie schnellen Zugriff auf jeden Bereich Ihres Projekts. Beispielsweise zeigt das Inspektorfenster die Eigenschaften und weitere Informationen für das Objekt an, das im Arbeitsbereich ausgewählt wurde. 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.

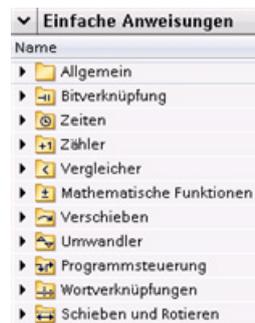
In der Editorleiste werden alle derzeit geöffneten Editoren angezeigt. Mit der Editorleiste arbeiten Sie so schneller und effizienter. Zum Umschalten zwischen geöffneten Editoren klicken Sie einfach auf den gewünschten Editor. Sie können auch zwei Editoren gleichzeitig anzeigen und diese vertikal oder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

2.1 Einfaches Einfügen von Anweisungen in Ihr Anwenderprogram

STEP 7 bietet Taskcards mit den Anweisungen für Ihr Programm. Die Anweisungen sind nach Funktionen gegliedert.



Um Ihr Programm anzulegen, ziehen Sie die Anweisungen von der Taskcard in ein Netzwerk.



2.2 Schneller Zugriff auf viel verwendete Operationen über die Funktionsleiste

STEP 7 enthält eine Funktionsleiste "Favoriten" für den schnellen Zugriff auf Anweisungen, die Sie häufig verwenden. Um eine Anweisung in Ihr Netzwerk einzufügen, genügt ein Klick auf das entsprechende Symbol.



(Um im Anweisungsverzeichnis die "Favoriten" aufzurufen, doppelklicken Sie auf das Symbol.)



Die Funktionsleiste "Favoriten" lässt sich problemlos anpassen und durch neue Anweisungen erweitern.

Fügen Sie die gewünschte Anweisung einfach mit "Drag & Drop" den "Favoriten" hinzu.

Für den Zugriff auf die Anweisung genügt jetzt ein Klick!



2.3 Einfaches Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen

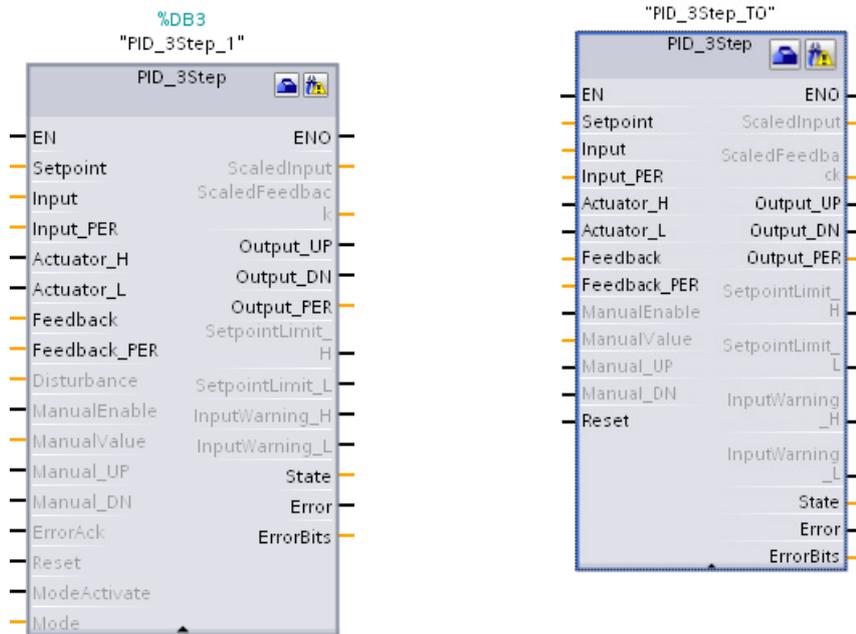


Bei einigen Anweisungen können Sie weitere Eingänge oder Ausgänge erstellen.

- Um einen Eingang oder Ausgang hinzuzufügen, klicken Sie auf das Symbol "Erstellen" oder an einem der vorhandenen Parameter IN oder OUT mit der rechten Maustaste auf den Eingangsanschluss und wählen den Befehl "Eingang einfügen".
- Um einen Eingang oder Ausgang zu löschen, klicken Sie bei einem der vorhandenen IN- oder OUTParameter mit der rechten Maustaste auf den Anschluss (sofern mehr als die zwei ursprünglichen Eingänge vorhanden sind) und wählen den Befehl "Löschen".

2.4 Erweiterbare Anweisungen

Einige der komplexeren Anweisungen sind erweiterbar und zeigen zunächst nur die wesentlichen Eingänge und Ausgänge an. Um alle Eingänge und Ausgänge vollständig anzuzeigen, klicken Sie auf den Pfeil im unteren Bereich der Anweisung.



2.5 Einfaches Ändern des CPU-Betriebszustands

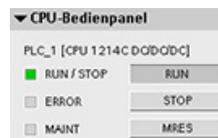
Die CPU verfügt nicht über einen physischen Schalter zum Ändern des Betriebszustands (STOP oder RUN).

Klicken Sie in der Funktionsleiste auf die Schaltfläche "CPU starten" oder "CPU stoppen", um den Betriebszustand der CPU zu ändern.



Beim Konfigurieren der CPU in der Gerätekonfiguration legen Sie das Anlaufverhalten der CPU über ihre Eigenschaften fest (Seite 84).

Im Portal "Online & Diagnose" steht Ihnen auch ein Bedienpanel zur Verfügung, in dem Sie den Betriebszustand der Online-CPU ändern können. Um das CPU-Bedienpanel verwenden zu können, müssen Sie mit der CPU online verbunden sein. Das in der Taskcard "Online-Tools" enthaltene Bedienpanel zeigt den Betriebszustand der Online-CPU an. Über das Bedienpanel können Sie auch den Betriebszustand der Online-CPU ändern.

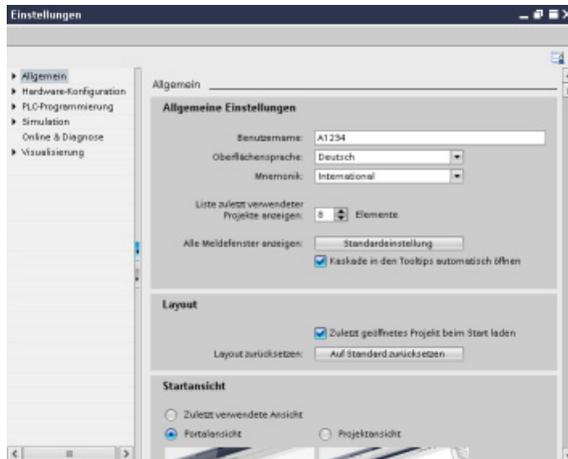


Mit der Schaltfläche auf dem Bedienpanel ändern Sie den Betriebszustand (STOP bzw. RUN). Außerdem enthält das Bedienpanel eine Schaltfläche MRES zum Umräumen des Speichers.

Der aktuelle Betriebszustand der CPU wird durch die Farbe der RUN/STOP-Anzeige angegeben. Gelb steht für den Betriebszustand STOP, Grün für RUN.

Unter Betriebszustände der CPU im S7-1200 Systemhandbuch finden Sie weitere Informationen zum Konfigurieren des standardmäßigen Betriebszustands beim Hochlauf.

2.6 Ändern des Erscheinungsbilds und der Konfiguration von STEP 7

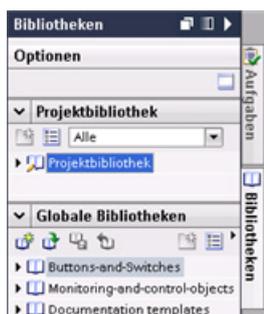


Sie haben zahlreiche Einstellmöglichkeiten; diese betreffen z. B. das Aussehen der Bedienoberfläche, die Sprache oder den Ordner zum Speichern Ihrer Arbeitsergebnisse.

Wählen Sie zum Ändern von Einstellungen im Menü "Optionen" den Befehl "Einstellungen".

2.7 Projektbibliotheken und globale Bibliotheken für einfachen Zugriff

Mit Hilfe der globalen Bibliothek und der Projektbibliothek können Sie gespeicherte Objekte innerhalb eines Projekts oder projektübergreifend wiederverwenden. So können Sie z. B. Bausteinvorlagen für verschiedene Projekte anlegen und die Vorlagen jeweils den besonderen Anforderungen der Automatisierungsaufgabe anpassen. Sie können unterschiedliche Objekte in Bibliotheken ablegen, so z. B. FCs, FBs, DBs, Gerätekonfigurationen, Datentypen, Beobachtungstabellen, Prozessbilder und Bildbausteine. Sie können außerdem die Komponenten der HMI-Geräte in Ihrem Projekt speichern.



Jedem Projekt ist eine Projektbibliothek zugeordnet, in der die während des Projekts mehrfach verwendeten Objekte abgelegt werden. Diese Projektbibliothek ist Teil des Projekts. Durch Öffnen oder Schließen des Projekts wird auch die Projektbibliothek geöffnet oder geschlossen. Durch Speichern des Projekts werden Änderungen in der Projektbibliothek gespeichert.

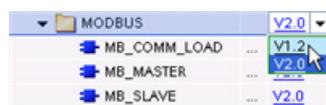
Sie können auch selbst eine globale Bibliothek anlegen, um die Objekte abzulegen, die für andere Projekte verfügbar sein sollen. Beim Anlegen einer neuen globalen Bibliothek speichern Sie diese Bibliothek an einem Speicherort auf Ihrem Computer oder Netzwerk.

2.8 Einfache Auswahl einer Version einer Anweisung

Durch die Entwicklung und die Ausgabezyklen bestimmter Befehlssätze (z. B. Modbus, PID und Bewegungssteuerung) ist es inzwischen zu mehreren freigegebenen Versionen dieser Anweisungen gekommen. Um die Kompatibilität und Migration mit älteren Projekten sicherzustellen, können Sie in STEP 7 auswählen, welche Anweisungsversion Sie in Ihr Anwenderprogramm einfügen.

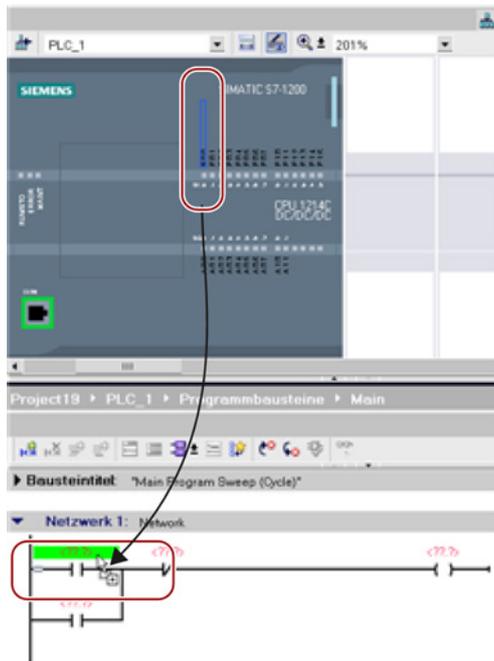


Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.



Um die Version einer Anweisung zu ändern, wählen Sie die entsprechende Version in der Klappliste aus.

2.9 Einfaches Drag & Drop zwischen Editoren



Damit Sie Aufgaben schnell und unkompliziert erledigen können, ermöglicht STEP 7 das Ziehen und Ablegen mit der Maus ("Drag & Drop") von Elementen zwischen den Editoren. So können Sie beispielsweise einen Eingang von der CPU an die Adresse einer Anweisung in Ihrem Anwenderprogramm ziehen.

Sie brauchen zum Auswählen der Eingänge oder Ausgänge der CPU mindestens den Zoomfaktor 200 %.

Beachten Sie, dass die Variablenamen nicht nur in der PLC-Variablen-tabelle, sondern auch in der CPU angezeigt werden.

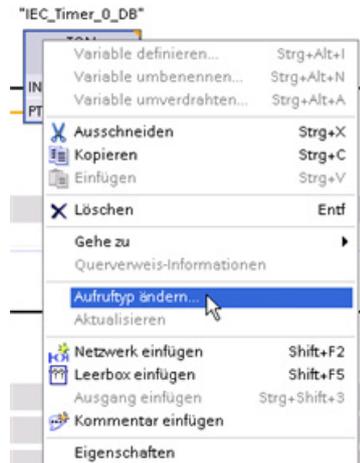
Um zwei Editoren gleichzeitig anzuzeigen, verwenden Sie den Menübefehl "Editor teilen" oder die entsprechende Schaltfläche in der Funktionsleiste.



Zum Umschalten zwischen den geöffneten Editoren klicken Sie auf die jeweiligen Symbole in der Editorleiste.



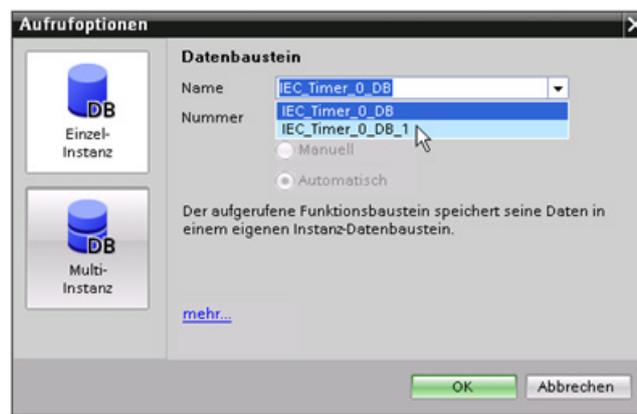
2.10 Aufruftyp eines DB ändern



In STEP 7 können Sie problemlos die Zuweisung einrichten oder ändern, die zwischen einem DB und einer Anweisung oder einem DB und einem FB besteht, der sich in einem FB befindet.

- Sie können die Zuweisung zwischen verschiedenen DBs umschalten.
- Sie können die Zuweisung zwischen einem Einzelinstanz-DB und einem Multiinstanz-DB wechseln.
- Sie können einen Instanz-DB erstellen (sofern ein Instanz-DB fehlt oder nicht verfügbar ist).

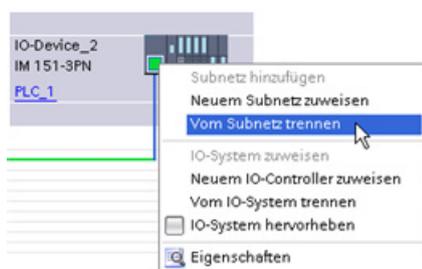
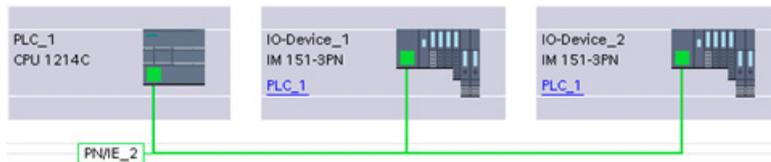
Sie können den Befehl "Aufruftyp ändern" auswählen, indem Sie entweder im Programmiereditor mit der rechten Maustaste auf die Anweisung oder den FB klicken oder indem Sie im Menü "Optionen" den Befehl "Baustein aufrufen" auswählen.



Im Dialog "Aufrufoptionen" können Sie einen Einzelinstanz- oder einen Multiinstanz-DB auswählen. Außerdem können Sie in einer Klappliste der verfügbaren DBs bestimmte DBs auswählen.

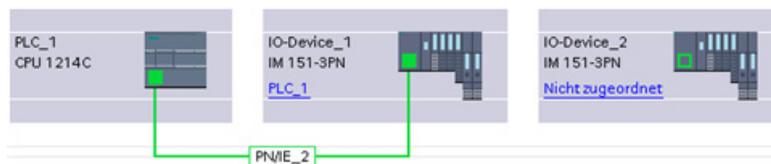
2.11 Geräte vorübergehend vom Netzwerk trennen

Sie können einzelne Netzwerkgeräte vom Subnetz trennen. Weil die Konfiguration des Geräts nicht aus dem Projekt entfernt wird, können Sie die Verbindung des Geräts mühelos wiederherstellen.



Klicken Sie mit der rechten Maustaste auf den Schnittstellenanschluss des Netzwerkgeräts und wählen Sie im Kontextmenü den Befehl "Vom Subnetz trennen".

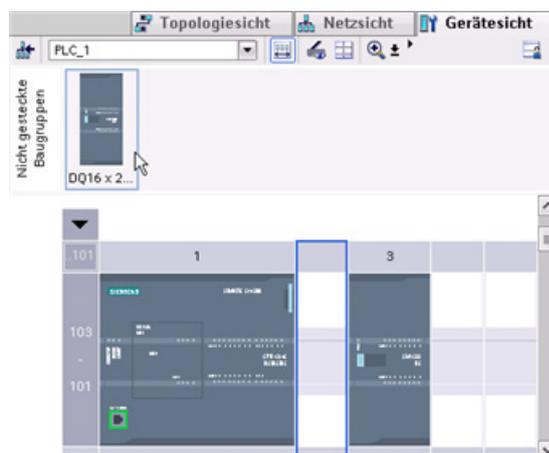
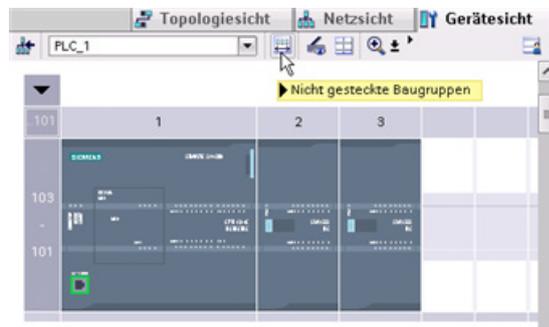
STEP 7 konfiguriert die Netzwerkverbindungen neu, entfernt das getrennte Gerät jedoch nicht aus dem Projekt. Die Netzwerkverbindung wird zwar gelöscht, doch die Schnittstellenadressen werden nicht verändert.



Wenn Sie die neuen Netzwerkverbindungen laden, müssen Sie die CPU in den Betriebszustand STOP versetzen.

Um das Gerät wieder anzuschließen, erstellen Sie einfach eine neue Netzwerkverbindung mit dem Anschluss des Geräts.

2.12 Einfaches virtuelles "Abziehen" von Baugruppen ohne Verlust der Konfiguration



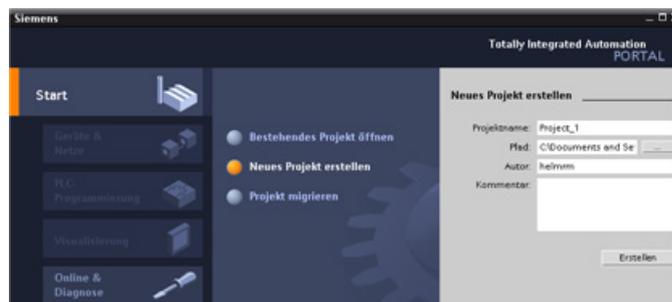
STEP 7 bietet Ihnen eine virtuelle Ablage für "nicht gesteckte" Baugruppen. Sie können eine Baugruppe vom Baugruppenträger "abziehen" und dabei die Konfiguration der Baugruppe speichern. Diese abgezogenen Baugruppen werden mit dem Projekt zusammen gespeichert; dadurch können Sie sie später wieder einsetzen, ohne die Parameter erneut konfigurieren zu müssen.

Diese Funktion wird z. B. bei kurzfristigen Wartungsarbeiten eingesetzt. Denken Sie beispielsweise an einen Fall, in dem Sie auf eine bestimmte Baugruppe warten, die ausgetauscht werden soll, und eine andere Baugruppe vorübergehend als Ersatz verwenden wollen. In diesem Fall können Sie die konfigurierte Baugruppe vom Baugruppenträger in die "Ablage nicht gesteckter Baugruppen" ziehen und anschließend die vorübergehende Ersatzbaugruppe einsetzen.

Erste Schritte

3.1 Projekt anlegen

Das Arbeiten mit STEP 7 ist einfach! Erfahren Sie selbst, wie schnell Sie ein Projekt anlegen können.



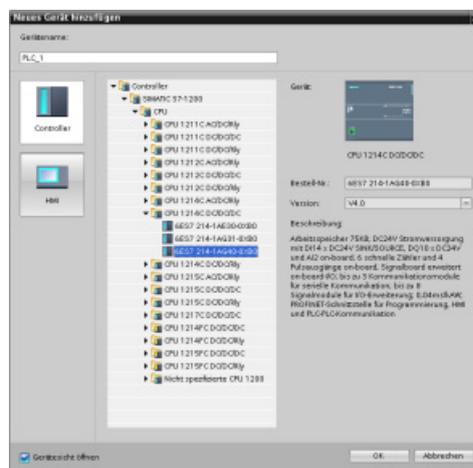
Klicken Sie im Portal "Start" auf die Aufgabe "Neues Projekt erstellen".

Geben Sie einen Projektnamen an und klicken Sie auf die Schaltfläche "Erstellen".



Wählen Sie nach dem Anlegen des Projekts das Portal "Geräte & Netze".

Klicken Sie auf "Neues Gerät hinzufügen".



Wählen Sie die in das Projekt aufzunehmende CPU:

1. Wählen Sie im Dialog "Neues Gerät hinzufügen" die Schaltfläche "SIMATIC PLC".
2. Wählen Sie in der Liste eine CPU.
3. Um die CPU dem Projekt hinzuzufügen, klicken Sie auf die Schaltfläche "Hinzufügen".

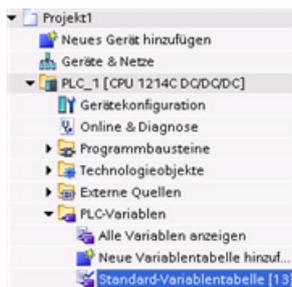
Beachten Sie, dass die Option "Gerätesicht öffnen" ausgewählt ist. Bei Auswahl dieser Option öffnen Sie durch Klicken auf "Hinzufügen" die "Gerätekonfiguration" der Projektansicht.



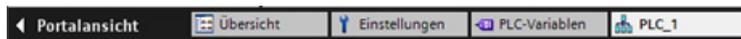
In der Gerätesicht wird die hinzugefügte CPU angezeigt.

3.2 Variablen für die E/A der CPU anlegen

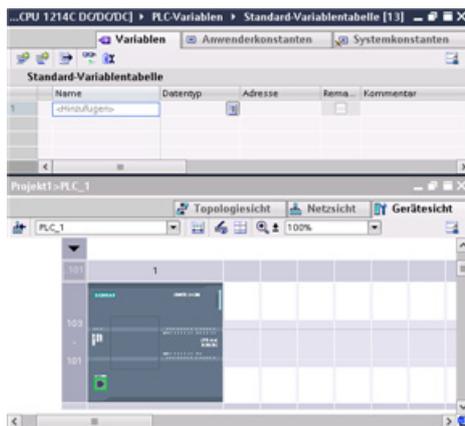
"PLC-Variablen" sind die symbolischen Namen für E/A und Adressen. Wenn Sie eine PLC-Variable anlegen, speichert STEP 7 die Variable in einer Variablen-tabelle. Der Zugriff auf die Variablen-tabelle kann über alle Editoren erfolgen (Programmiereditor, Geräteeditor, Visualisierungseditor und Beobachtungstabelleneditor).



Öffnen Sie bei geöffnetem Geräteeditor eine Variablen-tabelle. Die geöffneten Editoren werden in der Editorleiste angezeigt.



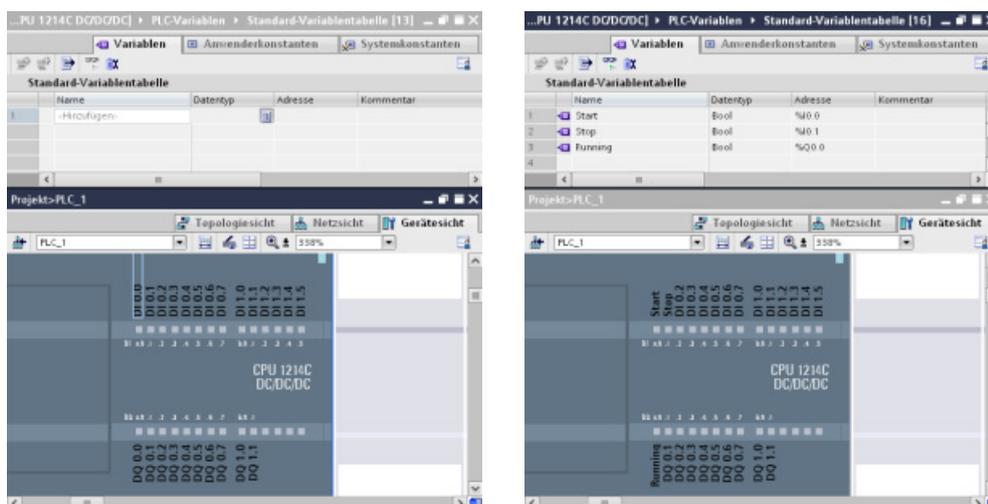
Klicken Sie in der Funktionsleiste auf die Schaltfläche zum horizontalen Teilen des Editorbereichs.



STEP 7 zeigt sowohl die Variablen-tabelle als auch den Geräteeditor an.

Zeigen Sie die Gerätekonfiguration mit einem Zoomfaktor über 200 % an, sodass die E/A lesbar und auswählbar sind. Ziehen Sie die Eingänge und Ausgänge aus der CPU in die Variablen-tabelle:

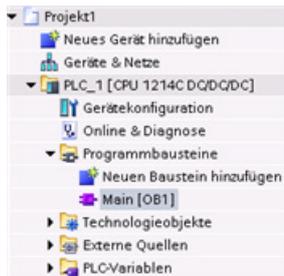
1. Wählen Sie "E0.0" und ziehen den Eingang in die erste Zeile der Variablen-tabelle.
2. Ändern Sie den Variablennamen von "E0.0" in "Start".
3. Ziehen Sie E0.1 in die Variablen-tabelle und ändern Sie den Namen in "Stop".
4. Ziehen Sie A0.0 (im unteren Bereich der CPU) in die Variablen-tabelle und ändern Sie den Namen in "Running".



Nachdem die Variablen in die PLC-Variablen-tabelle eingetragen sind, stehen sie Ihrem Anwenderprogramm zur Verfügung.

3.3 Einfaches Netzwerk im Anwenderprogramm anlegen

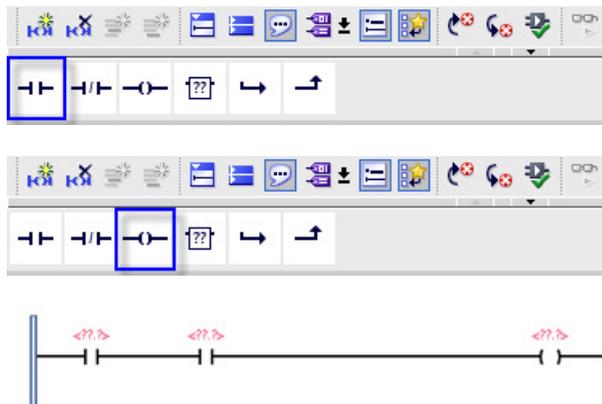
Ihr Programmcode besteht aus Anweisungen, die von der CPU der Reihe nach ausgeführt werden. Legen Sie in diesem Beispiel den Programmcode im Kontaktplan (KOP) an. Das KOP-Programm besteht aus einer Folge von Netzwerken, die den Strompfaden eines Schaltplans ähneln.



Um den Programmiereditor zu öffnen, gehen Sie folgendermaßen vor:

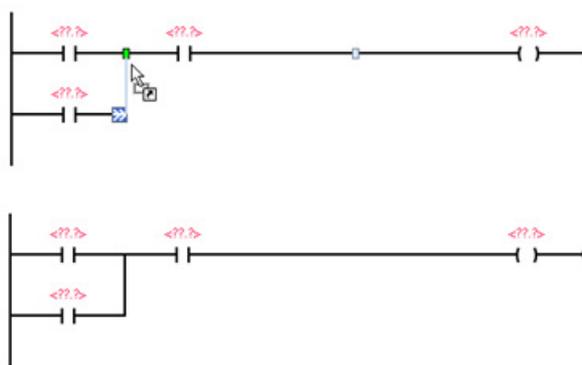
1. Erweitern Sie den Ordner "Programmbausteine" in der Projektnavigation zur Anzeige des Bausteins "Main [OB1]".
 2. Doppelklicken Sie auf den Baustein "Main [OB1]".
- Der Programmiereditor öffnet den Programmbaustein (OB1).

Mit den Schaltflächen in der Funktionsleiste "Favoriten" können Sie Kontakte und Spulen in Ihr Netzwerk einfügen.



1. Um einen Kontakt in das Netzwerk einzufügen, klicken Sie in der Funktionsleiste "Favoriten" auf die Schaltfläche "Schließerkontakt".
2. Fügen Sie in diesem Beispiel einen zweiten Kontakt ein.
3. Zum Einfügen einer Spule klicken Sie auf "Ausgangsspule".

Die "Favoriten" enthalten auch eine Schaltfläche zum Anlegen einer Verzweigung.

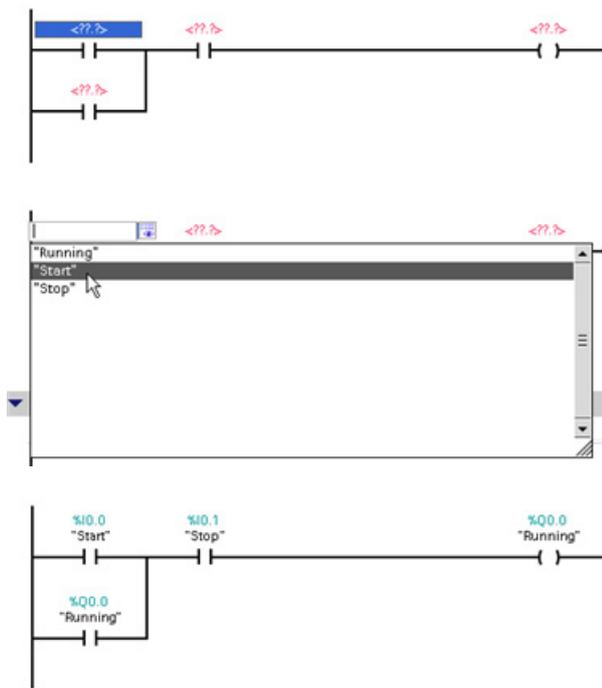


1. Wählen Sie die linke Schiene, um die Schiene für die Verzweigung auszuwählen.
2. Um eine Verzweigung zur Schiene des Netzwerks hinzuzufügen, klicken Sie auf das Symbol für "Verzweigung öffnen".
3. Fügen Sie einen weiteren Schließer in die geöffnete Verzweigung ein
4. Ziehen Sie den Pfeil mit zwei Spitzen zu einem Verbindungspunkt (dem grünen Quadrat auf dem Strompfad) zwischen den zwei Kontakten auf dem ersten Strompfad.

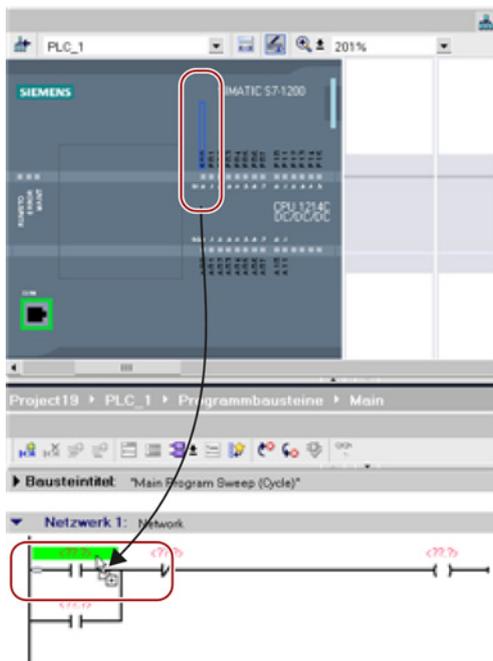
Um das Projekt zu speichern, klicken Sie in der Funktionsleiste auf "Projekt speichern". Sie können auch dann speichern, wenn die Bearbeitung des Strompfads noch nicht abgeschlossen ist. Sie können jetzt den Variablennamen die entsprechenden Anweisungen zuordnen.

3.4 Adressieren Sie die Anweisungen mithilfe der PLC-Variablen in der Variablen-tabelle

Die Variablen-tabelle beschleunigt das Eingeben der PLC-Variablen für die Adressen der Kontakte und Spulen erheblich.



1. Doppelklicken Sie auf die Standardadresse <???.?> über dem ersten Öffner.
2. Klicken Sie auf das Symbol für den Auswahlvorgang rechts neben der Adresse, um die Variablen in der Variablen-tabelle zu öffnen.
3. Wählen Sie in der Klappliste "Start" für den ersten Kontakt.
4. Wiederholen Sie für den zweiten Kontakt die obigen Schritte und wählen Sie die Variable "Stop".
5. Für die Spule und den Selbsthaltekontakt wählen Sie die Variable "Running".



Sie können die E/A-Adressen auch direkt von der CPU ziehen. Dazu müssen Sie nur den Arbeitsbereich der Projektansicht teilen (Seite 40).

Zeigen Sie zum Auswählen der E/A die CPU mindestens mit Zoomfaktor 200 % an.

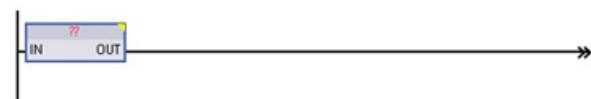
So können Sie die E/A der CPU aus der "Gerätekonfiguration" in die KOP-Anweisung im Programmiereditor ziehen und damit nicht nur die Adresse für die Anweisung, sondern gleichzeitig einen Eintrag in der PLC-Variablen-tabelle anlegen.

3.5 "Box"-Anweisung hinzufügen

Der Programmiereditor bietet eine allgemeine "Box"-Anweisung. Nach dem Einfügen dieser Box-Anweisung können Sie die Art der Anweisung, z. B. eine Anweisung ADD, aus einer Klappliste auswählen.



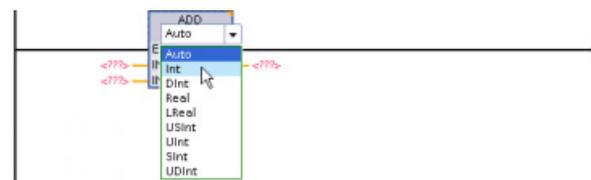
Klicken Sie in der Funktionsleiste "Favoriten" auf die allgemeine "Box"-Anweisung.



Unter der allgemeinen "Box"-Anweisung sind eine Reihe von Anweisungen verfügbar. Für dieses Beispiel erstellen Sie eine Anweisung ADD:



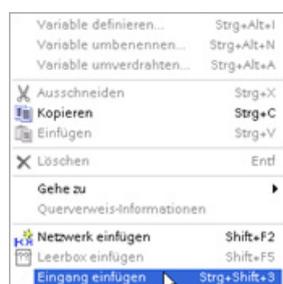
1. Öffnen Sie die Klappliste mit Anweisungen, indem Sie auf die gelbe Ecke der "Box"-Anweisung klicken.
2. Blättern Sie in der Liste nach unten und wählen Sie die Anweisung ADD aus.



3. Klicken Sie auf die gelbe Ecke neben dem "?", um den Datentyp für Eingaben und Ausgaben zu wählen.



Jetzt können Sie die Variablen (oder Speicheradressen) für die Werte eingeben, die mit der Anweisung ADD verwendet werden sollen.



Sie können außerdem für bestimmte Anweisungen zusätzliche Eingänge angeben:

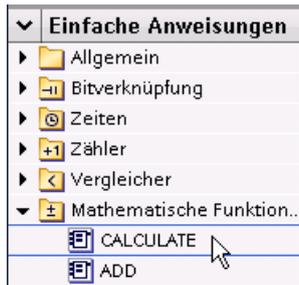
1. Klicken Sie auf einen der Eingänge in der Box.
2. Klicken Sie mit der rechten Maustaste, um das Kontextmenü aufzurufen, und wählen Sie "Eingang einfügen".



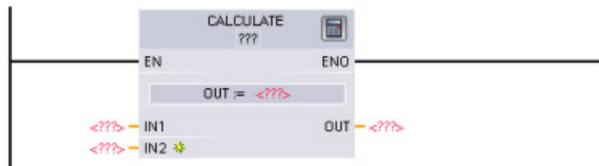
Die Anweisung ADD verwendet nun drei Eingänge.

3.6 Anweisung CALCULATE für komplexe mathematische Gleichungen verwenden

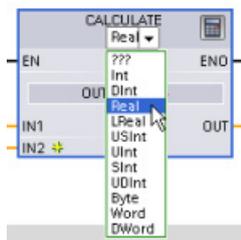
Mit der Anweisung Calculate können Sie eine mathematische Funktion erstellen, die mehrere Eingangsparameter verarbeitet und das Ergebnis entsprechend der von Ihnen vorgegebenen Gleichung ausgibt.



Erweitern Sie im Basic-Anweisungsverzeichnis den Ordner der mathematischen Funktionen. Doppelklicken Sie auf die Anweisung Calculate, um sie in Ihr Anwenderprogramm einzufügen.



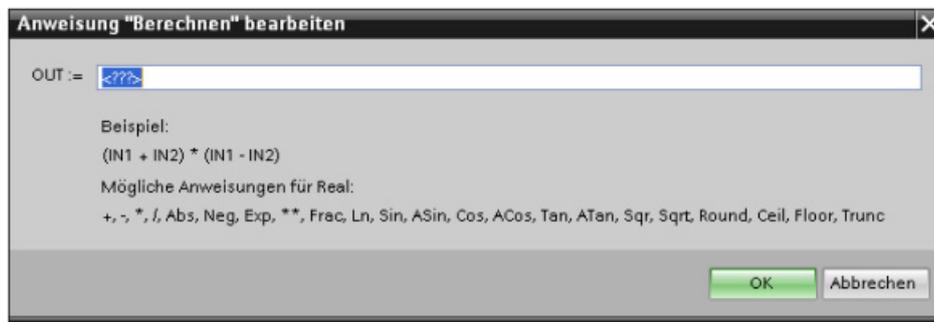
Die nicht konfigurierte Anweisung Calculate bietet zwei Eingangsparameter und einen Ausgangsparameter.



Klicken Sie auf "???" und wählen Sie die Datentypen für die Eingangs- und Ausgangsparameter aus. (Alle Eingangs- und Ausgangsparameter müssen denselben Datentyp haben.) Wählen Sie für dieses Beispiel den Datentyp "Real" aus.



Klicken Sie auf das Symbol "Gleichung bearbeiten", um die Gleichung einzugeben.



3.6 Anweisung CALCULATE für komplexe mathematische Gleichungen verwenden

Geben Sie in diesem Beispiel die folgende Gleichung zum Skalieren eines Rohanalogwerts ein. (Die Bezeichnungen "In" und "Out" entsprechen den Parametern der Anweisung Calculate.)

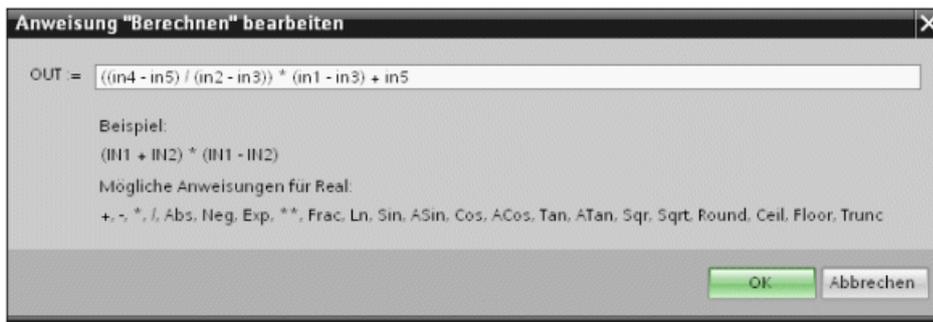
$$\text{Out}_{\text{value}} = ((\text{Out}_{\text{high}} - \text{Out}_{\text{low}}) / (\text{In}_{\text{high}} - \text{In}_{\text{low}})) * (\text{In}_{\text{value}} - \text{In}_{\text{low}}) + \text{Out}_{\text{low}}$$

$$\text{Out} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$

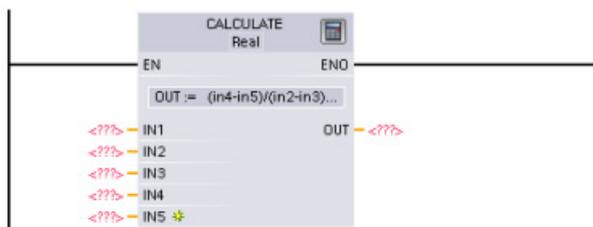
Erläuterung:	Out _{value}	(Out)	Skalierter Ausgangswert
	In _{value}	(in1)	Analogeingangswert
	In _{high}	(in2)	Oberer Grenzwert für den skalierten Eingangswert
	In _{low}	(in3)	Unterer Grenzwert für den skalierten Eingangswert
	Out _{high}	(in4)	Oberer Grenzwert für den skalierten Ausgangswert
	Out _{low}	(in5)	Unterer Grenzwert für den skalierten Ausgangswert

Geben Sie im Feld "Calculate bearbeiten" die Gleichung mit den Parameternamen ein:

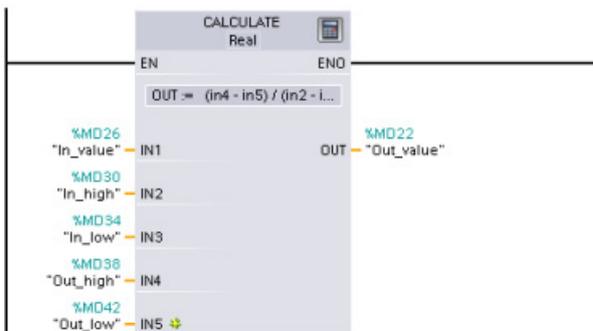
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



Wenn Sie auf "OK" klicken, erstellt die Anweisung Calculate die für die Anweisung erforderlichen Eingänge.



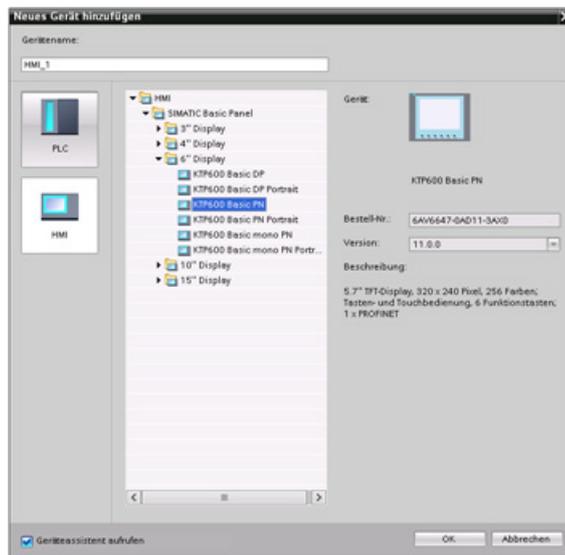
Geben Sie die Variablennamen für die Werte ein, die den Parametern entsprechen.



3.7 HMI-Gerät zum Projekt hinzufügen



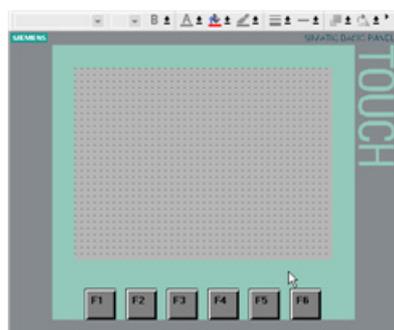
Das Hinzufügen eines HMI-Geräts zum Projekt ist einfach!



1. Doppelklicken Sie auf das Symbol für "Neues Gerät hinzufügen".
2. Wählen Sie im Dialog "Neues Gerät hinzufügen" die Schaltfläche "SIMATIC HMI".
3. Wählen Sie das gewünschte HMI-Gerät aus der Liste aus.

Sie können die Bilder für das HMI-Gerät auch mit Hilfe des Geräteassistenten konfigurieren.

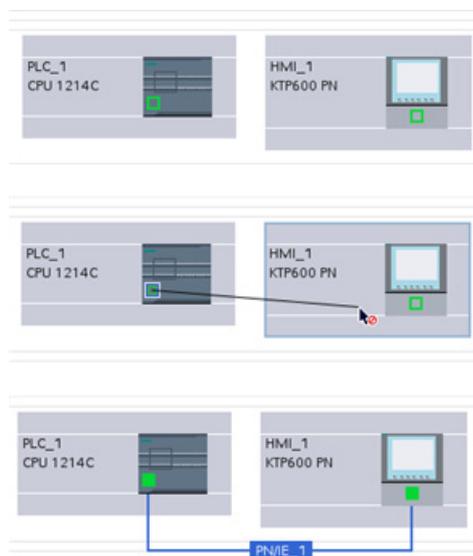
4. Klicken Sie auf "OK", um das HMI-Gerät zum Projekt hinzuzufügen.



Das TIA Portal fügt das HMI-Gerät dem Projekt hinzu. Das TIA Portal bietet Ihnen einen HMI-Assistenten zur einfacheren Konfiguration aller Bilder und der Struktur Ihres HMI-Geräts.

Wenn Sie den Geräteassistenten nicht ausführen, legt das TIA Portal ein einfaches HMI-Standardbild an. Später können Sie weitere Bilder oder Objekte in Bildern hinzufügen.

3.8 Netzwerkverbindung zwischen CPU und HMI-Gerät herstellen

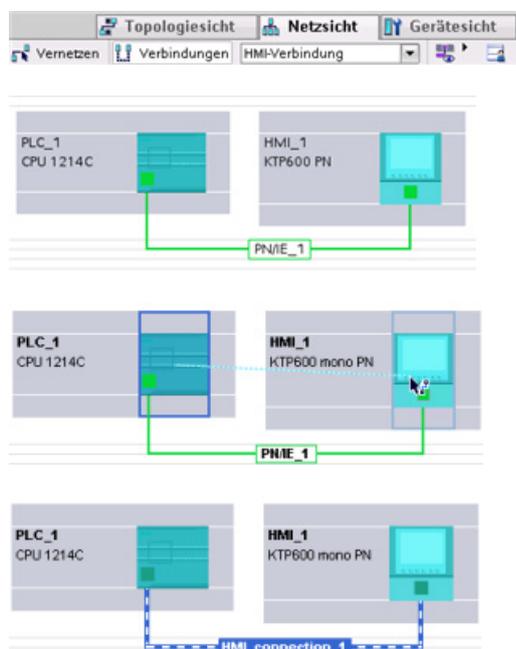


Das Erstellen einer Netzwerkverbindung ist einfach!

- Navigieren Sie zu "Geräte & Netze" und wählen Sie die Netzsicht, um CPU und HMI-Gerät anzuzeigen.
- Um ein PROFINET-Netzwerk zu erstellen, ziehen Sie eine Linie von dem grünen Quadrat (Ethernet-Port) auf dem Gerät zu dem grünen Quadrat des anderen Geräts.

Für die beiden Geräte wird eine Netzwerkverbindung hergestellt.

3.9 HMI-Verbindung zur gemeinsamen Nutzung von Variablen erstellen



Wenn Sie eine HMI-Verbindung zwischen den beiden Geräten herstellen, können Sie anschließend problemlos Variablen mit beiden Geräten gemeinsam nutzen.

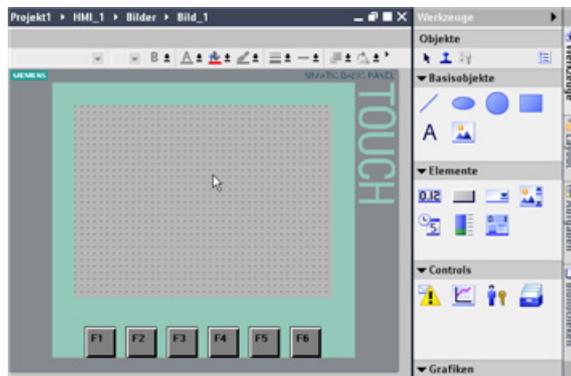
- Klicken Sie bei ausgewählter Netzwerkverbindung auf die Schaltfläche "Verbindungen" und wählen Sie dann in der Klappliste "HMI-Verbindung" aus.
- Hierdurch erscheinen die beiden Geräte in blauer Anzeigefarbe.
- Wählen Sie die CPU aus und ziehen Sie die Linie zum HMI-Gerät.
- Das Konfigurieren der Variablen für diese HMI-Verbindung geschieht durch Auswählen aus einer Liste von PLC-Variablen.

Sie können eine HMI-Verbindung auch auf andere Arten erstellen:

- Wenn Sie eine PLC-Variable aus der PLC-Variablen-tabelle, aus dem Programmiereditor oder dem Gerätekonfigurationseditor in den Editor für das HMI-Bild ziehen, wird dadurch automatisch eine HMI-Verbindung erstellt.
- Wenn Sie im HMI-Assistenten nach dem PLC-Gerät suchen, wird automatisch eine HMI-Verbindung erstellt.

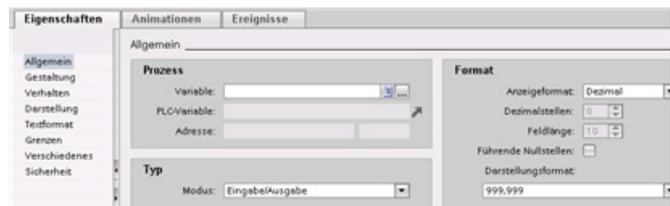
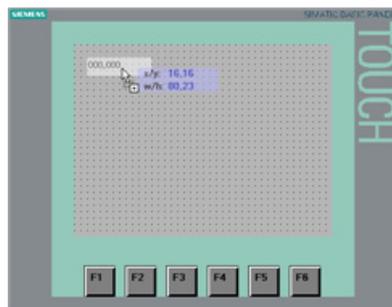
3.10 HMI-Bild anlegen

Auch ohne Einsatz des Geräteassistenten ist das Konfigurieren eines HMI-Bildes einfach.



STEP 7 stellt standardmäßig eine Gruppe von Bibliotheken bereit, aus denen Grundformen, interaktive Elemente und Standardgrafiken eingefügt werden können.

Um ein Element einzufügen, brauchen Sie das Element nur auf das Bild zu ziehen und dort abzulegen. Das Konfigurieren von Aussehen und Verhalten des Elements erfolgt über die Elementeigenschaften im Inspektorfenster.



Sie können die Elemente auf Ihrem Bild auch dadurch erstellen, dass Sie PLC-Variablen mit der Maus entweder aus der Projektnavigation oder aus dem Programmiereditor in das HMI-Bild ziehen. Die PLC-Variable wird dann zu einem Element in dem Bild. Sie können die Parameter für dieses Element anhand der Eigenschaften ändern.

3.11 PLC-Variable für das HMI-Element auswählen

Nachdem Sie das Element in Ihrem Bild erstellt haben, weisen Sie dem Element über dessen Eigenschaften eine PLC-Variable zu. Wenn Sie auf die Auswahl Schaltfläche neben dem Variablenfeld klicken, werden die PLC-Variablen der CPU angezeigt.



Sie können PLC-Variablen auch mit der Maus aus der Projektnavigation in das HMI-Bild ziehen. Rufen Sie die PLC-Variablen in der Ansicht "Details" in der Projektnavigation auf und ziehen Sie die Variable dann mit der Maus in das HMI-Bild.

SPS-Grundlagen leicht gemacht

4.1 Bei jedem Zyklus ausgeführte Arbeitsschritte

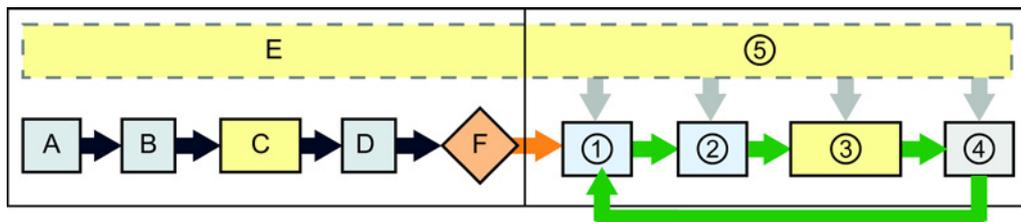
Jeder Zyklus umfasst das Schreiben der Ausgänge, das Lesen der Eingänge, das Bearbeiten der Anweisungen des Anwenderprogramms und die Durchführung der Systemwartung oder Hintergrundverarbeitung.



Dieser Zyklus wird als Abtastzyklus oder Abtastung bezeichnet. Unter Standardbedingungen werden alle digitalen und analogen Ein- und Ausgänge synchron zum Zyklus mit einem internen Speicherbereich, dem so genannten Prozessabbild, aktualisiert. Das Prozessabbild enthält ein Momentabbild der physischen Ein- und Ausgänge von CPU, Signalboard und Signalmodulen.

- Die CPU liest die physischen Eingänge unmittelbar vor der Ausführung des Anwenderprogramms und speichert die Eingangswerte im Prozessabbild der Eingänge. Dadurch wird sichergestellt, dass diese Werte während der Ausführung der Anwenderanweisungen konsistent bleiben.
- Die CPU führt die Logik der Anwenderanweisungen durch und aktualisiert die Ausgangswerte im Prozessabbild der Ausgänge, statt in die tatsächlichen physischen Ausgänge zu schreiben.
- Nach Ausführung des Anwenderprogramms schreibt die CPU die resultierenden Ausgänge aus dem Prozessabbild der Ausgänge in die physischen Ausgänge.

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.



STARTUP

- A Löscht den Speicherbereich für Eingänge ("E")
- B Die Ausgänge werden mit dem letzten Wert oder dem Ersatzwert initialisiert
- C Die Anlauf-OBs werden ausgeführt
- D Der Zustand der physischen Eingänge wird in den Speicherbereich E kopiert
- E Alle Alarmereignisse werden in der Warteschlange für die Verarbeitung im Betriebszustand RUN gespeichert
- F Das Schreiben des Speicherbereichs für die Ausgänge ("A") in die physischen Ausgänge wird freigegeben

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

Sie können das Standardverhalten eines Moduls ändern, indem Sie dieses aus der automatischen E/A-Aktualisierung herausnehmen. Sie können ferner sofort bei der Ausführung einer Anweisung digitale und analoge E/A-Werte lesen und schreiben. Durch das direkte Lesen der physischen Eingänge wird das Prozessabbild der Eingänge nicht verändert. Durch das direkte Schreiben in die physischen Ausgänge werden das Prozessabbild der Ausgänge und der physische Ausgang geändert.

4.2 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 und Sie können ein Projekt laden. Die RUN/STOP-LED leuchtet dauerhaft gelb.
- Im Betriebszustand STARTUP führt die CPU eine Anlauflogik (sofern vorhanden) aus. Die CPU verarbeitet im Betriebszustand STARTUP keine Alarmereignisse. Die RUN/STOP-LED blinkt abwechselnd grün und gelb.
- Im Betriebszustand RUN wird der Zyklus wiederholt ausgeführt. Alarmereignisse können auftreten und von der CPU an beliebigen Punkten innerhalb der Programmzyklusphase abgearbeitet werden. Im Betriebszustand RUN können einige Teile eines Projekts geladen werden. Die RUN/STOP-LED leuchtet dauerhaft grün.

Die CPU unterstützt den Warmstart, um in den Betriebszustand RUN zu gehen. Während eines Warmstarts wird kein Umräumen ausgeführt, Sie können jedoch ein Umräumen über STEP 7 auslösen. Beim Umräumen 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 IP-Adressen werden beim Umräumen nicht gelöscht. Ein Warmstart initialisiert alle nicht remanenten System- und Anwenderdaten.

Die Einstellung für das Anlaufverhalten der CPU nach NETZ-EIN kann einschließlich der Anlaufart vollständig über STEP 7 konfiguriert werden. 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. Bei der Systeminitialisierung löscht die CPU den gesamten Speicherbereich mit nicht remanenten Merkern und setzt alle nicht remanenten DBs auf ihre Ausgangswerte zurück. Dann schaltet die CPU in die jeweilige Anlaufart. Bestimmte Fehler verhindern, dass die CPU in den Betriebszustand RUN geht. Die CPU unterstützt die folgenden Anlaufarten: Betriebszustand STOP, "Wechsel in Betriebszustand RUN nach Warmstart" und "Wechsel in vorhergehenden Betriebszustand nach Warmstart".

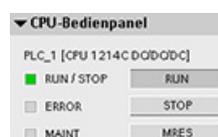
ACHTUNG

Konfiguration des Warmstarts

Die CPU kann aufgrund von behebbaren Fehlern in den Betriebszustand STOP gehen. Dies können z.B. der Ausfall eines ersetzbaren Signalmoduls oder temporäre Fehler sein wie Störungen der Netzleitung oder ein unvorhersehbares Anlaufereignis.

Wenn für die CPU die Einstellung "Warmstart - Betriebsart vor NETZ-AUS" konfiguriert ist, kehrt sie nach Behebung des Fehlers erst dann in den Betriebszustand RUN zurück, wenn sie einen neuen Befehl von STEP 7 erhält, um in den Betriebszustand RUN zu gehen. Ohne einen neuen Befehl behält die CPU den Betriebszustand STOP als den Zustand vor NETZ-AUS bei.

Für CPUs, die für den von einer STEP 7-Verbindung unabhängigen Betrieb vorgesehen sind, sollte üblicherweise die Einstellung "Warmstart - RUN" konfiguriert sein, damit die CPU nach Behebung eines Fehlerzustands durch Aus- und Wiedereinschalten wieder in den Betriebszustand RUN versetzt werden kann.



Die CPU verfügt nicht über einen physischen Schalter zum Ändern des Betriebszustands (STOP bzw. RUN). Um den Betriebszustand der CPU zu ändern, bietet STEP 7 die folgenden Tools:

- Schaltflächen "STOP" und "RUN" in der Symbolleiste von STEP 7
- CPU-Bedienpanel in den Online-Tools

Sie können ferner eine STP-Anweisung 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.

4.3 Ausführung des Anwenderprogramms

Die CPU unterstützt die folgenden Bausteinararten 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 eigenen Startereignissen (Seite 63) 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 (Instanz-DB) für die Speicherung von Zustandswerten während der Ausführung, die von anderen Bausteinen im Programm verwendet werden können.
- Datenbausteine (DBs) speichern Daten, die von den Programmbausteinen verwendet werden können.

Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU (Seite 15) begrenzt. Die Anzahl der einzelnen OBs, FCs, FBs und DBs ist nicht begrenzt. Die Gesamtzahl der Bausteine darf jedoch 1024 nicht überschreiten.

4.3.1 Bearbeitung 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.

4.3.2 OBs strukturieren Ihr Anwenderprogramm

OBs steuern die Ausführung des Anwenderprogramms. Die Ausführung eines Organisationsbausteins wird durch bestimmte Ereignisse in der CPU angestoßen. OBs können sich nicht gegenseitig aufrufen und sie können auch nicht aus einer FC oder einem FB aufgerufen werden. Nur ein Ereignis wie zum Beispiel ein Diagnosealarm oder ein Zeitintervall kann die Ausführung eines OB anstoßen. Die CPU bearbeitet die OBs dann entsprechend ihren Prioritätsklassen, wobei OBs mit höheren Prioritäten zuerst ausgeführt werden. Die niedrigste Prioritätsklasse ist 1 (für den Hauptprogrammzyklus), die höchste ist 24.

4.3.3 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 Zeitverzögerungereignisse 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). In 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 werden in der Reihenfolge der Priorität der Ereignisse ausgeführt, die die OBs auslösen. Bei Version 4.0 der S7-1200 CPUs 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.

Nachfolgend sind zwei Fälle beschrieben, in denen Alarmereignisse einen Weckalarm-OB und einen Zeitverzögerungs-OB auslösen. In beiden Fällen ist der Zeitverzögerungs-OB (OB201) keinem Teilprozessabbild zugeordnet und wird mit Priorität 4 ausgeführt. Der Weckalarm-OB (OB200) ist Teilprozessabbild PIP1 zugeordnet und wird mit Priorität 2 ausgeführt. Die folgenden Abbildungen zeigen die unterschiedliche Ausführung unterbrechbarer und nicht unterbrechbarer OBs:

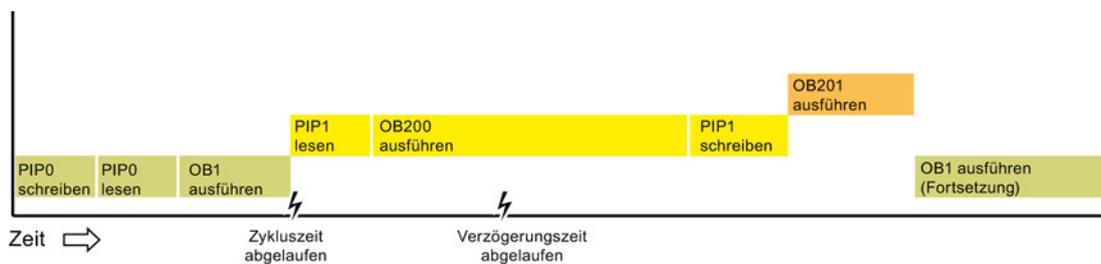


Bild 4-1 Fall 1: Nicht unterbrechbare OB-Ausführung

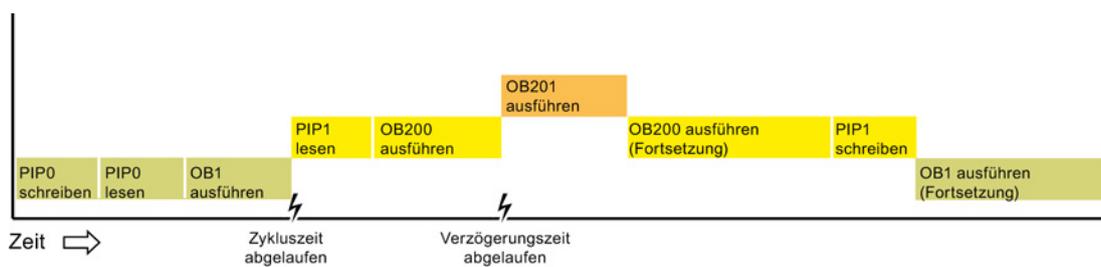


Bild 4-2 Fall 2: Unterbrechbare OB-Ausführung

Hinweis

Wenn Sie die OB-Ausführungsart als nicht unterbrechbar festlegen, kann ein Zeitfehler-OB lediglich Programmzyklus-OBs unterbrechen. In den Versionen vor V4.0 der S7-1200 CPU konnte ein Zeitfehler-OB jede OB-Ausführung unterbrechen. Ab V4.0 muss die OB-Ausführung als unterbrechbar konfiguriert werden, wenn ein Zeitfehler-OB (oder ein anderer OB mit höherer Priorität) die Ausführung von OBs, die keine Programmzyklus-OBs sind, unterbrechen soll.

Wissenswertes zu Prioritäten und Warteschlange für die Ausführung von Ereignissen

Die Zahl anstehender Ereignisse aus einer einzigen Quelle kann begrenzt werden, 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. Weitere Informationen zum Warteschlangenüberlauf finden Sie im Thema zu Zeitfehler-OBs.

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 4- 1 OB-Ereignisse

Ereignis	Zulässige Anzahl	OB-Standardpriorität
Programmzyklus	1 Programmzyklusereignis Mehrere OBs zulässig	1 ⁴
Anlauf	1 Anlaufereignis ¹ Mehrere OBs zulässig	1 ⁴
Verzögerung	Bis zu 4 Zeitereignisse 1 OB je Ereignis	3
Weckalarm	Bis zu 4 Ereignisse 1 OB je Ereignis	8
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
Ziehen oder Stecken von Modulen	1 Ereignis	6
Fehler bei Baugruppenträger oder Station	1 Ereignis	6
Tageszeit	Bis zu 2 Ereignisse	2
Zustand	1 Ereignis	4

4.3 Ausführung des Anwenderprogramms

Ereignis	Zulässige Anzahl	OB-Standardpriorität
Update	1 Ereignis	4
Profil	1 Ereignis	4

- 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 V4.0-CPU ist 22. Wenn Sie eine V3.0-CPU durch eine V4.0 CPU 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 hierzu finden Sie unter "Austauschen Ihrer V3.0-CPU durch eine V4.0-CPU".

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 4-2 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.
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.
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	Falls der Baustein mit dem Fehler über eine Fehlerbehandlungsroutine verfügt, aktualisiert er die Fehlerstruktur; falls nicht, schreibt die CPU den Fehler in den Diagnosepuffer 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.

Siehe auch

Online gehen, um die Werte in der CPU zu beobachten (Seite 287)

4.4 Speicherbereiche, Adressierung und Datentypen

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

- Der Ladespeicher ist ein nichtflüchtiger Speicher für Anwenderprogramm, Daten und Konfiguration. Beim Laden eines Projekts in die CPU wird das Projekt zunächst im Ladespeicher abgelegt. Dieser Speicher befindet sich entweder auf einer Memory Card (sofern vorhanden) oder in der CPU. Der nicht-flüchtige Ladespeicher bleibt auch bei einem Spannungsausfall erhalten. Sie können die Kapazität des für Datenprotokolle verfügbaren Ladespeichers mit Hilfe einer Memory Card erhöhen.
- Der Arbeitsspeicher ist ein flüchtiger Speicher für einige Elemente des Anwenderprojekts während der Bearbeitung des Anwenderprogramms. Die CPU kopiert einige Elemente des Projekts aus dem 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. Der remanente Speicherbereich dient zum Speichern der Werte ausgewählter Adressen des Anwenderspeichers bei Spannungsausfall. Kommt es zu einer geplanten Spannungsunterbrechung oder einem Spannungsausfall, werden diese remanenten Werte beim Einschalten von der CPU wiederhergestellt.



Mit der optional erhältlichen SIMATIC Memory Card verfügen Sie über eine alternative Speichereinrichtung für Ihr Anwenderprogramm oder zum Übertragen des Programms. Wenn Sie die Memory Card nutzen, führt die CPU das Programm aus der Memory Card und nicht aus dem Speicher der CPU aus.

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

Sie können die optionale SIMATIC Memory Card als Programmkarte, als Übertragungskarte, zum Erfassen von Datenprotokolldateien oder zur Durchführung eines Firmware-Updates nutzen.

- Bei Verwendung als Übertragungskarte können Sie damit Ihr Projekt unter Umgehung von STEP 7 in mehrere CPUs kopieren. Das auf der Übertragungskarte gespeicherte Projekt wird in den Speicher der CPU kopiert. Sie müssen die Übertragungskarte nach dem Kopieren des Programms in die CPU ziehen.
- Die Programmkarte nimmt die Stelle des CPU-Speichers ein. Alle CPU-Funktionen werden von der Programmkarte gesteuert. Wenn Sie die Programmkarte stecken, wird der interne Ladespeicher der CPU gelöscht (auch das Anwenderprogramm und ggf. geforcte E/A). Die CPU führt das Anwenderprogramm dann von der Programmkarte aus.
- Auf der Programmkarte können Sie auch Datenprotokolldateien speichern (Seite 126). Die Programmkarte bietet eine größere Kapazität als der interne Speicher der CPU. Mit der Webserver-Funktion (Seite 209) der CPU können Sie die Datenprotokolldateien auf einen Computer herunterladen.
- Sie können eine Memory Card auch zur Durchführung eines Firmware-Updates verwenden. Weitere Informationen finden Sie im S7-1200 Systemhandbuch.

Hinweis

Die Programmkarte **muss** in der CPU gesteckt bleiben. Wenn Sie die Programmkarte ziehen, geht die CPU in den Betriebszustand STOP.

4.4.1 Von der S7-1200 unterstützte 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.

Tabelle 4-3 Von der S7-1200 unterstützte Datentypen

Datentypen	Beschreibung
Bit- und Bitfolge-Datentypen	<ul style="list-style-type: none"> • Bool ist ein Boolescher Wert bzw. ein Bitwert. • Byte ist ein 8-Bit-Wert. • Word ist ein 16-Bit-Wert. • DWord ist ein 32-Bit-Wert (Doppelwort).
Ganzzahlige Datentypen	<ul style="list-style-type: none"> • USInt (vorzeichenlose 8-Bit-Ganzzahl) und SInt (vorzeichenbehaftete 8-Bit-Ganzzahl) sind "kurze" Ganzzahlen (8 Bit bzw. 1 Byte im Speicher), die mit oder ohne Vorzeichen sein können. • UInt (vorzeichenlose 16-Bit-Ganzzahl) und Int (vorzeichenbehaftete 16-Bit-Ganzzahl) sind Ganzzahlen (16 Bit bzw. 1 Wort im Speicher), die mit oder ohne Vorzeichen sein können. • UDInt (vorzeichenlose 32-Bit-Ganzzahl) und DInt (vorzeichenbehaftete 32-Bit-Ganzzahl) sind doppelte Ganzzahlen (32 Bit bzw. 1 Doppelwort im Speicher), die mit oder ohne Vorzeichen sein können.
Realzahl-Datentypen	<ul style="list-style-type: none"> • Real ist eine 32-Bit-Realzahl bzw. ein Gleitpunktwert. • LReal ist eine 64-Bit-Realzahl bzw. ein Gleitpunktwert.

Datentypen	Beschreibung
Datum- und Uhrzeit-Datentypen	<ul style="list-style-type: none"> • Date ist ein 16-Bit-Datumswert (ähnlich wie UInt) mit der Anzahl von Tagen seit dem 1. Januar 1990. Der maximale Datumswert ist 65378 (16#FF62), dies entspricht dem 31. Dezember 2168. Alle möglichen Date-Werte sind gültig. • DTL (Datum und Uhrzeit lang) ist eine Struktur aus 12 Bytes, in der Informationen zum Datum und zur Uhrzeit in einer vordefinierten Struktur gespeichert werden. <ul style="list-style-type: none"> – Jahr (UInt): 1970 bis 2554 – Monat (UInt): 1 bis 12 – Tag (UInt): 1 bis 31 – Wochentag (UInt): 1 (Sonntag) bis 7 (Samstag) – Stunden (UInt): 0 bis 23 – Minuten (UInt): 0 bis 59 – Sekunden (UInt): 0 bis 59 – Nanosekunden (UDInt): 0 bis 999999999 • Time ist ein 32-Bit-IEC-Zeitwert (ähnlich wie Dint), der die Anzahl Millisekunden speichert (von 0 bis 24 Tage 20 Stunden 31 Minuten 23 Sekunden und 647 ms). Alle möglichen Time-Werte sind gültig. Time-Werte können für Berechnungen verwendet werden, negative Zeiten sind möglich. • TOD (Tageszeit) ist ein 32-Bit-Uhrzeitwert (ähnlich wie Dint), der die Anzahl Millisekunden seit Mitternacht enthält (von 0 bis 86399999).
Zeichen- und Zeichenfolge-Datentypen	<ul style="list-style-type: none"> • Char ist ein 8-Bit-Einzelzeichen. • String ist eine Zeichenfolge variabler Länge mit maximal 254 Zeichen.
Array- und Struktur-Datentypen	<ul style="list-style-type: none"> • Array enthält mehrere Elemente desselben Datentyps. Arrays können in der Bausteinschnittstelle von OB, FC, FB und DB angelegt werden. Im PLC-Variableneditor können Sie kein Array erstellen. • Struct definiert eine Struktur von Daten, die aus anderen Datentypen bestehen. Der Datentyp Struct kann genutzt werden, um eine Gruppe zusammengehöriger Prozessdaten als eine Dateneinheit zu behandeln. Sie deklarieren den Namen und die interne Datenstruktur für den Datentyp Struct im Datenbausteineditor oder in einem Bausteinschnittstelleneditor. <p>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.</p>

Datentypen	Beschreibung
PLC-Datentypen	<p>Der Datentyp PLC ist eine vom Anwender angelegte Datenstruktur, die eine benutzerspezifische Datenstruktur vorgibt, die Sie in Ihrem Programm mehrmals verwenden können. Wenn Sie einen PLC-Datentyp anlegen, erscheint der neue PLC-Datentyp in der Auswahl-Klappliste im DB-Editor und im Codebaustein-Schnittstelleneditor.</p> <p>PLC-Datentypen können direkt als Datentyp in einer Codebausteinschnittstelle oder in Datenbausteinen verwendet werden.</p> <p>PLC-Datentypen können als Vorlage für die Erstellung von mehreren globalen Datenbausteinen mit der gleichen Datenstruktur verwendet werden.</p>
Pointer-Datentypen	<ul style="list-style-type: none"> • Pointer bietet einen indirekten Verweis auf die Adresse einer Variablen. Der Datentyp belegt 6 Bytes (48 Bits) im Speicher und kann die folgenden Informationen zu einer Variable enthalten: DB-Nummer (oder 0, wenn die Daten nicht in einem DB gespeichert werden), Speicherbereich in der CPU und die Adresse im Speicher. • Any bietet einen indirekten Verweis auf den Anfang eines Datenbereichs und gibt dessen Länge an. Der Pointer Any belegt 10 Bytes im Speicher und kann die folgenden Informationen enthalten: Datentyp der Datenelemente, Anzahl der Datenelemente, Speicherbereich oder DB-Nummer und die "Byte.Bit"-Anfangsadresse der Daten. • Variant bietet einen indirekten Verweis auf Variablen verschiedener Datentypen oder Parameter. Der Pointer Variant erkennt Strukturen und einzelne Strukturkomponenten. Der Datentyp Variant belegt keinen Platz im Speicher.

Ferner werden die folgenden BCD-Zahlenformate (binärcodierte Dezimalzahlen) von den Konvertierungsanweisungen unterstützt, obwohl sie nicht als Datentypen zur Verfügung stehen.

- BCD16 ist ein 16-Bit-Wert (-999 bis 999).
- BCD32 ist ein 32-Bit-Wert (-9999999 bis 9999999).

4.4.2 Adressierung der Speicherbereiche

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.
- **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.

Beim Forcen wird ein fester Wert nur für einen physischen Eingang (Ex.y:P) oder einen physischen Ausgang (Ax.y:P) übernommen. Um einen Eingang oder Ausgang zu forcen, hängen Sie an die PLC-Variable oder die Adresse den Code ":P" an. Beachten Sie für weitere Informationen den Abschnitt "Variablen in der CPU forcen" (Seite 290).

Tabelle 4-4 Speicherbereiche

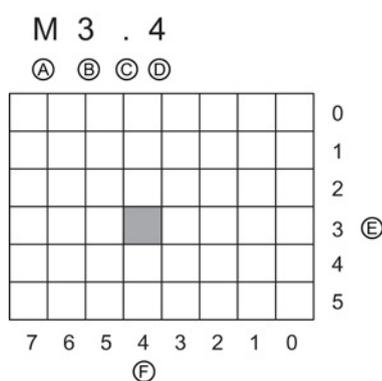
Speicherbereich	Beschreibung	Forcen	Remanent
E Prozessabbild der Eingänge	Wird zu Beginn des Zyklus aus den physischen Eingängen kopiert	Nein	Nein
E_:P ¹ (physischer Eingang)	Direktes Lesen der physischen Eingänge von CPU, SB oder SM	Ja	Nein
A Prozessabbild der Ausgänge	Wird zu Beginn des Zyklus in die physischen Ausgänge kopiert	Nein	Nein
A_:P ¹ (physischer Ausgang)	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 auf die physischen Eingänge und Ausgänge (oder um diese zu forcen) 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:

- Speicherbereich (wie E, A oder M)
- Größe der Daten, auf die zugegriffen werden soll (wie "B" für Byte oder "W" für Word)
- Adresse der Daten (wie Byte 3 oder Word 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).

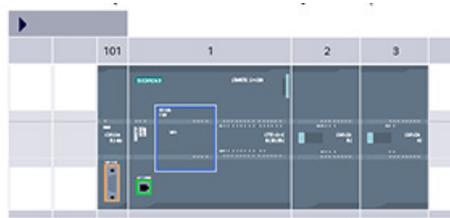


Absolute Adresse eines Speicherbereichs:

- 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.

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



Geräteübersicht						
Baugruppe	Steckp.	E-Adresse	A-Adresse	Typ	Bestel	
	103					
	102					
RS485_1	101			CM 1241 (RS485)	6ES7	
PLC_1	1			CPU 1214C DCD/	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 Ihren Konfigurationsbildschirm einfügen, werden E- und A-Adressen automatisch zugewiesen. Sie können die voreingestellte Adressierung ändern, indem Sie in der Gerätekonfiguration das Adressfeld auswählen und neue Zahlen eingeben.

- Digitale Eingänge und Ausgänge werden 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 Gruppen zu je 2 Ein- bzw. Ausgängen zugewiesen (4 Byte).

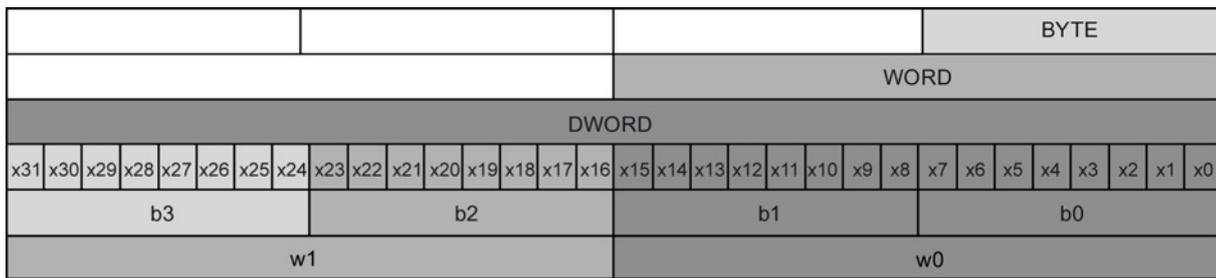
Das Bild zeigt ein Beispiel für eine CPU 1214C mit zwei SMs und einem SB. In diesem Beispiel können Sie in der Adresse des Moduls DE8 die 8 in 2 ändern. Das Werkzeug unterstützt Sie, indem Adressbereiche, die die falsche Größe haben oder mit anderen Adressen in Konflikt stehen, geändert werden.

4.4.3 Zugriff auf eine "Scheibe" 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 Datenscheibe 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 mit Doppelwortgröße kann über die Bits 0-31, Bytes 0-3 oder die Wörter 0-1 zugegriffen werden. Auf eine Variable mit Wortgröße kann über die Bits 0-15, Bytes 0-1 oder das Wort 0 zugegriffen werden. Auf eine Variable mit Bytegröße kann über die Bits 0-7 oder das Byte 0 zugegriffen werden. Bit-, Byte- und Wort-Scheiben können überall dort verwendet werden, wo Bits, Bytes oder Wörter erwartete Operanden sind.

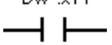
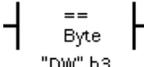
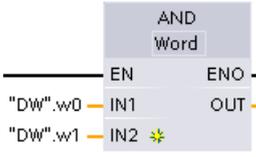
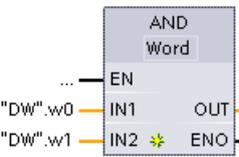


Hinweis

Auf folgende Datentypen kann über Scheiben 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 Scheiben zugegriffen werden, auf Datenbausteinvariablen vom Typ Real jedoch nicht.

Beispiele

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

	KOP	FUP	SCL
Bitzugriff	<p>"DW".x11</p> 		<pre>IF "DW".x11 THEN ... END_IF;</pre>
Bytezugriff	<p>"DW".b2</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>

4.4.4 Zugriff auf eine Variable mit einer AT-Überlagerung

Mit Hilfe der AT-Variablenüberlagerung können Sie mit einer überlagerten Deklaration eines unterschiedlichen Datentyps auf eine bereits deklarierte Variable eines Standardzugriffsbausteins zugreifen. Sie können beispielsweise die einzelnen Bits einer Variable vom Datentyp Byte, Word oder DWord mit einem Bool-Array adressieren.

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
▼	AT	AT "B1" Array [0..7] of Bool
■	AT[0]	Bool
■	AT[1]	Bool
■	AT[2]	Bool
■	AT[3]	Bool
■	AT[4]	Bool
■	AT[5]	Bool
■	AT[6]	Bool
■	AT[7]	Bool

Tabelle 4- 5 Überlagerung eines Bytes mit einem Booleschen Array

7	6	5	4	3	2	1	0
AT[0]	AT[1]	AT[2]	AT[3]	AT[4]	AT[5]	AT[6]	AT[7]

Ein weiteres Beispiel ist eine Variable vom Typ DWord, die mit einer Struktur überlagert wird:

■	DW1	DWord
▼	DW1_Struct	AT "DW1" Struct
■	S1	Word
■	S2	Byte
■	S3	Byte

Die Überlagerungstypen können in der Programmlogik direkt angesprochen werden:

KOP	FUP	SCL
		<pre>IF #AT[1] THEN ... END_IF;</pre>
		<pre>IF (#DW1_Struct.S1 = W#16#000C) THEN ... END_IF;</pre>
		<pre>out1 := #DW1_Struct.S2;</pre>

Regeln

- Die Überlagerung von Variablen ist nur in FB- und FC-Bausteinen mit Standardzugriff möglich.
- Sie können Parameter für alle Bausteinarten 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" gekennzeichnet sein.

4.5 Impulsausgänge

Die CPU oder das Signalboard (SB) kann mit vier Impulsgeneratoren für die Steuerung schneller Impulsausgänge konfiguriert werden, dabei sind die Impulsdauermodulation (PWM) oder die Impulsfolge (PTO) möglich. Die grundlegenden Bewegungssteuerungsanweisungen nutzen PTO-Ausgänge. Sie können jeden Impulsgenerator entweder PWM oder PTO zuordnen, jedoch nicht beiden gleichzeitig.



Impulsausgänge können nicht von anderen Anweisungen im Anwenderprogramm verwendet werden. Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren 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

Überschreiten Sie nicht die maximale Impulsfrequenz.

Die maximale Impulsfrequenz der Impulsgeneratoren beträgt 1 MHz bei der CPU 1217C und 100 kHz bei den CPUs 1211C, 1212C, 1214C und 1215C, 20 kHz (bei einem Standard-SB) bzw. 200 kHz (bei einem schnellen SB).

Den vier Impulsgeneratoren sind standardmäßig E/A zugewiesen, sie können jedoch für jeden Digitalausgang an der CPU oder am SB konfiguriert werden. Dezentralen E/A können die Impulsgeneratoren an der CPU nicht zugewiesen werden.

Beachten Sie beim Konfigurieren der grundlegenden Bewegungssteuerungsanweisungen, dass Sie von STEP 7 **nicht** gewarnt werden, wenn Sie eine Achse mit einer maximalen Geschwindigkeit oder Frequenz, die diese Hardwareeinschränkung überschreitet, konfigurieren. Dies kann zu Problemen in Ihrer Anwendung führen. Stellen Sie also stets sicher, dass Sie die maximale Impulsfrequenz der Hardware nicht überschreiten.

Sie können integrierte CPU-Ausgänge oder die Ausgänge eines optionalen Signalboards nutzen. In der folgenden Tabelle sind 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.

Den vier Impulsgeneratoren sind standardmäßig E/A zugewiesen, sie können jedoch für jeden Digitalausgang an der CPU oder am SB konfiguriert werden. Signalmodulen oder dezentralen E/A können die Impulsgeneratoren an der CPU nicht zugewiesen werden.

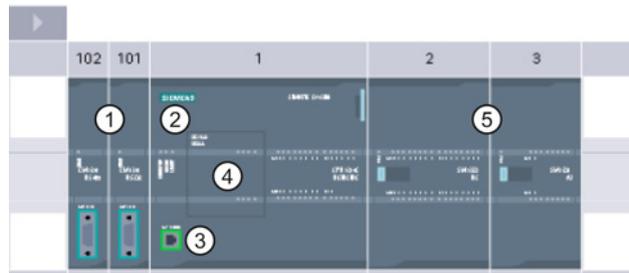
Tabelle 4- 6 Standard-Ausgangszuweisungen der Impulsgeneratoren

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
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.

Einfache Erstellung der Gerätekonfiguration

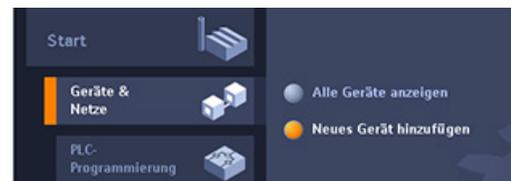
Sie können die Gerätekonfiguration für Ihr PLC-Gerät durch Hinzufügen einer CPU und weiterer Module zu Ihrem Projekt erstellen.



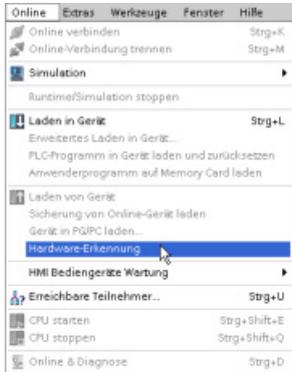
- ① Kommunikationsmodul (CM) oder Kommunikationsprozessor (CP): bis zu 3, in Steckplätzen 101, 102 und 103
- ② CPU: Steckplatz 1
- ③ Ethernet-Anschluss der CPU
- ④ Signalboard (SB), Kommunikationsboard (CB) oder Batterieboard (BB): max. 1, in CPU gesteckt
- ⑤ Signalmodul (SM) für digitale oder analoge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

Um die Gerätekonfiguration anzulegen, fügen Sie Ihrem Projekt zunächst ein Gerät hinzu.

- Wählen Sie in der Portalansicht das Portal "Geräte & Netze" und klicken Sie auf "Gerät hinzufügen".
- Doppelklicken Sie in der Projektansicht unter dem Projektnamen auf "Neues Gerät hinzufügen".

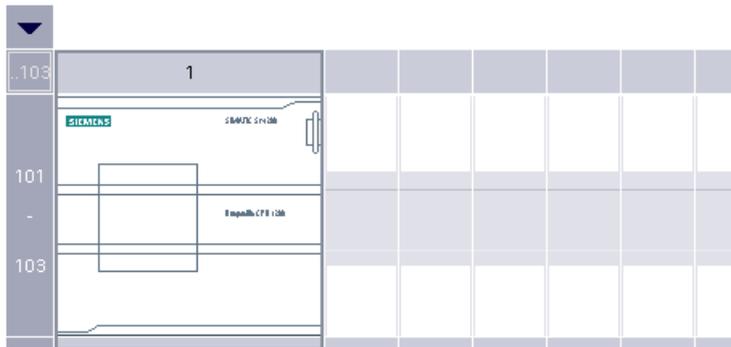


5.1 Konfiguration für eine nicht spezifizierte CPU erkennen



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".

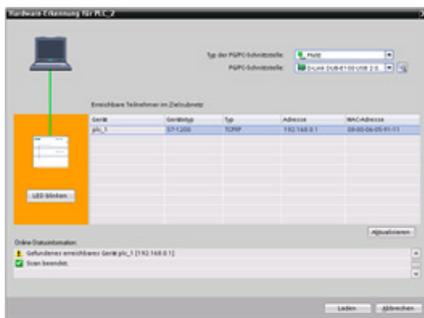
Wählen Sie im Gerätekonfigurationseditor die Option zum Erkennen der Konfiguration des angeschlossenen Geräts.



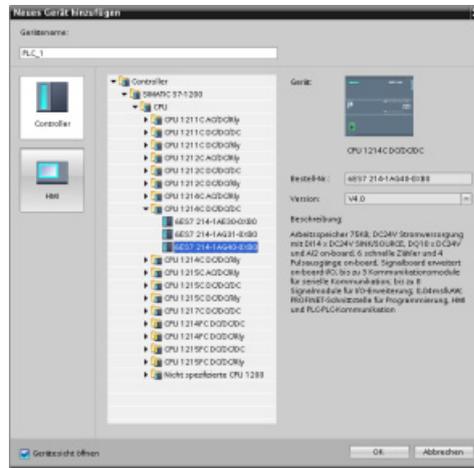
Das Gerät ist nicht spezifiziert.

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

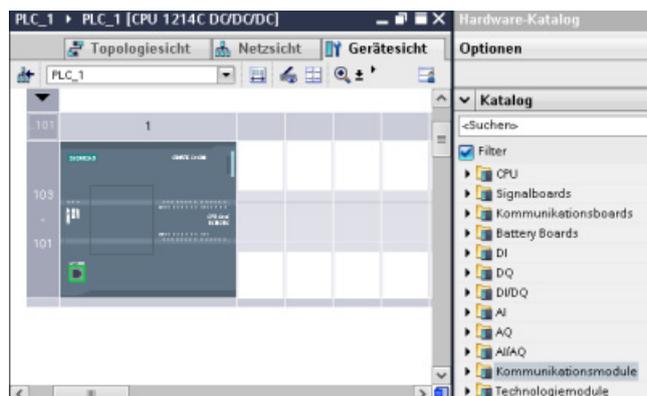
Nachdem Sie im Online-Dialog die CPU ausgewählt und auf die Schaltfläche zum Laden geklickt haben, lädt STEP 7 die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Sie können dann die Parameter für die CPU und die Module (Seite 84) konfigurieren.



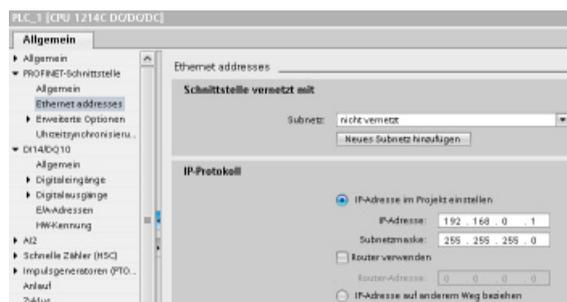
5.2 CPU zur Konfiguration hinzufügen



Zum Erstellen Ihrer Gerätekonfiguration fügen Sie eine CPU in Ihr Projekt ein. Wählen Sie im Dialog "Neues Gerät hinzufügen" die CPU aus und klicken Sie zum Hinzufügen der CPU zum Projekt auf "OK".



In der Gerätesicht wird die hinzugefügte CPU mit Baugruppenträger angezeigt.



Nach der Auswahl der CPU in der Gerätesicht werden die Eigenschaften der CPU im Inspektorfenster angezeigt. Über diese Eigenschaften können Sie die Betriebsparameter der CPU konfigurieren (Seite 84).

Hinweis

Die CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher bei der Gerätekonfiguration manuell eine IP-Adresse zuweisen. Ist Ihre CPU an einen Router im Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.

5.3 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 Kompatibilitätswarnungen für die beiden betroffenen Geräte angezeigt.

Hinweis

Geräteaustausch: Ersetzen einer V3.0-CPU durch eine V4.0-CPU

Sie können ein STEP 7 V12-Projekt in STEP 7 V13 öffnen und V3.0-CPU durch V4.0-CPU ersetzen. Sie können keine CPUs mit Vorgängerversionen vor V3.0 ersetzen. Berücksichtigen Sie beim Ersetzen einer V3.0-CPU durch eine V4.0-CPU die Unterschiede (Seite 377) in den Funktionen und im Verhalten zwischen den beiden Versionen und zwischen den von Ihnen zu ergreifenden Maßnahmen.

Wenn Sie mit einem Projekt für eine CPU-Version vor V3.0 arbeiten, können Sie das Projekt zunächst auf V3.0 hochrüsten und dann das V3.0-Projekt auf V4.0 hochrüsten.

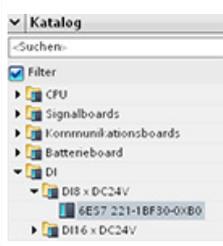
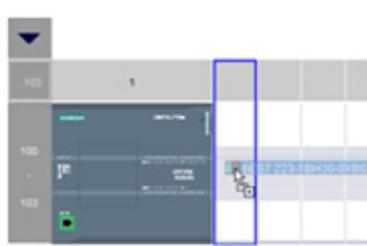
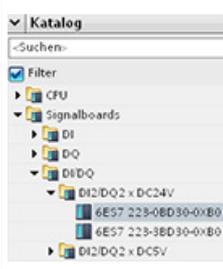
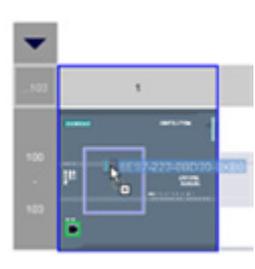
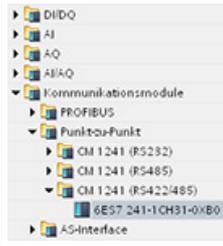
5.4 Module zur Konfiguration hinzufügen

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.
- 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.

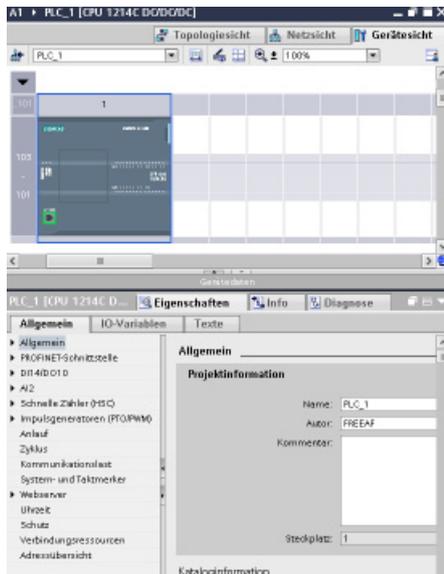
Um ein Modul in die Gerätekonfiguration einzufügen, wählen Sie das Modul im Hardwarekatalog aus und klicken doppelt darauf oder ziehen Sie es in den markierten Steckplatz. Sie müssen die Module in die Gerätekonfiguration aufnehmen und die Hardwarekonfiguration in die CPU laden, damit die Module funktionsfähig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

Tabelle 5- 1 Modul zur Gerätekonfiguration hinzufügen

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

5.5 Konfigurieren des CPU- und Modulbetriebs

Um die Betriebsparameter der CPU zu konfigurieren, wählen Sie die CPU in der Gerätesicht aus und öffnen im Inspektorfenster das Register "Eigenschaften".



Sie können die folgenden CPU-Eigenschaften festlegen:

- PROFINET-IP-Adresse und Uhrzeitsynchronisation für die CPU
- Anlaufverhalten der CPU nach einem Wechsel von AUS nach EIN
- Integrierte digitale und analoge E/A, schnelle Zähler (HSC) und Impulsgeneratoren
- Systemuhr (Uhrzeit, Zeitzone und Sommer-/Winterzeit)
- Lese-/Schreibschutz und Passwort für den Zugriff auf die CPU
- Maximale Zykluszeit oder feste Mindestzykluszeit und Kommunikationslast
- Webserver-Eigenschaften

Konfigurieren des CPU-Betriebs für den Wechsel von STOP nach RUN

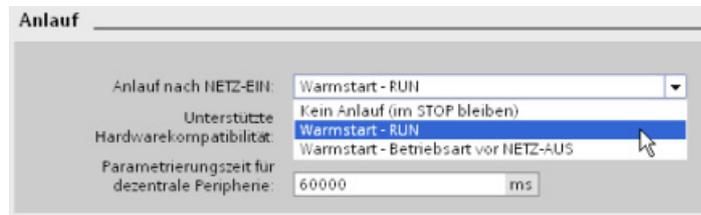
Immer wenn der Betriebszustand von STOP nach RUN wechselt, löscht die CPU das Prozessabbild der Eingänge, initialisiert das Prozessabbild der Ausgänge und verarbeitet die Anlauf-OBs. (Deshalb ergeben alle Lesezugriffe auf das Prozessabbild der Eingänge von Anweisungen im Anlauf-OB den Wert Null und nicht den aktuellen Wert des physischen Eingangs.) Um den aktuellen Zustand eines physischen Eingangs beim Anlauf zu lesen, müssen Sie den Eingang direkt auslesen. Dann werden die Anlauf-OBs und zugehörige FCs und FBs ausgeführt. Sind mehrere Anlauf-OBs vorhanden, so werden diese fortlaufend nach OB-Nummer, beginnend mit der niedrigsten OB-Nummer, abgearbeitet.

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.

Über die CPU-Eigenschaften konfigurieren Sie das Anlaufverhalten der CPU nach einem Neustart.



- Im Betriebszustand STOP
- Im Betriebszustand RUN
- Im vorherigen Betriebszustand (vor dem Neustart)

Die CPU führt einen Warmstart durch, bevor sie nach RUN wechselt. Bei einem Warmstart werden alle nicht remanenten Speicher auf die voreingestellten Startwerte zurückgesetzt. Die CPU puffert jedoch die aktuellen Werte des remanenten Speichers.

Hinweis

Die CPU führt nach dem Laden immer einen Neustart durch

Wenn Sie ein Element Ihres Projekts in die CPU laden (z. B. einen Programmbaustein, einen Datenbaustein, die Hardwarekonfiguration), führt die CPU beim nächsten Wechsel in den Betriebszustand RUN einen Neustart durch. Neben dem Löschen der Eingänge, Initialisieren der Ausgänge und Initialisieren des nicht remanenten Speichers werden beim Neustart auch die remanenten Speicherbereiche initialisiert.

Nach dem Neustart, der auf einen Ladevorgang folgt, wird bei jedem nachfolgenden Wechsel von STOP in RUN ein Warmstart durchgeführt (dabei wird der remanente Speicher nicht initialisiert).

5.5.1 Systemmerker und Taktmerker bieten Standardfunktionen

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 "Diagnosediagramm geändert" 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- 2 Systemspeicher

7	6	5	4	3	2	1	0
Reserviert Wert 0				Immer ausgeschaltet Wert 0	Immer eingeschaltet Wert 1	Diagnosestatusanzeige • 1: Änderung • 0: Keine Änderung	Anzeige erster Zyklus • 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:

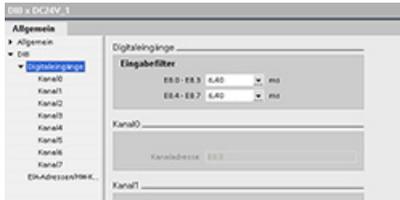
Tabelle 5- 3 Taktmerker

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.

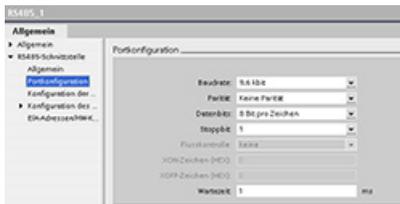
Konfigurieren des Betriebs der E/A und der Kommunikationsmodule

Um die Betriebsparameter der Signalmodule (SM), der Signalboards (SB) oder der Kommunikationsmodule (CM) zu konfigurieren, wählen Sie das Modul in der Gerätesicht aus und öffnen im Inspektorfenster das Register "Eigenschaften".



Signalmodul (SM) und Signalboard (SB)

- Digitale E/A: Konfigurieren der einzelnen Eingänge, z. B. für die Flankenerkennung und den Impulsabgriff (Eingang bleibt nach einem High- oder Low-Impuls einen Zyklus lang ein- oder ausgeschaltet) Konfigurieren der Ausgänge für die Verwendung eines eingefrorenen Werts oder eines Ersatzwerts bei einem Wechsel von RUN nach STOP
- Analoge E/A: Konfigurieren der Parameter für einzelne Eingänge (z. B. Spannung/Strom, Bereich und Glättung) und Freigabe der Unter- oder Überlaufdiagnose Konfigurieren der Parameter für einzelne Analogausgänge und Freigabe der Diagnose, z. B. von Kurzschlüssen (bei Spannungsausgängen) oder Überlaufwerten
- E/A-Adressen: Konfigurieren der Startadresse für die Ein- und Ausgänge des Moduls



Kommunikationsmodul (CM) und Kommunikationsboard (CB)

- Schnittstellenkonfiguration: Konfigurieren der Kommunikationsparameter wie Baudrate, Parität, Datenbits, Stoppbits und Wartezeit
- Meldung aus Zwischenspeicher übertragen und Meldung in Zwischenspeicher empfangen Konfigurieren der Optionen zum Senden und Empfangen von Daten (z. B. die Parameter für Meldungsbeginn und Meldungsende)

Diese Konfigurationsparameter können auch mit dem Anwenderprogramm geändert werden.

5.6 Konfigurieren der IP-Adresse der CPU

Da die CPU keine vorkonfigurierte IP-Adresse besitzt, müssen Sie manuell eine IP-Adresse zuweisen. Sie konfigurieren die IP-Adresse und die übrigen Parameter für die PROFINET-Schnittstelle bei der Konfiguration der Eigenschaften für die CPU.

- In einem PROFINET-Netzwerk ist jedem Gerät zur Identifikation eine eindeutige MAC-Adresse (Media-Access-Control-Adresse) vom Hersteller zugewiesen. Jedes Gerät benötigt außerdem eine IP-Adresse.
- Ein Subnetz ist eine logische Gruppierung miteinander verbundener Netzwerkgeräte. Eine Maske (die Subnetz- oder Netzmaske) legt die Grenzen eines Subnetzes fest. Die einzige Verbindung zwischen verschiedenen Subnetzen läuft über einen Router. Router fungieren als Bindeglieder zwischen LANs und benötigen IP-Adressen zum Senden und Empfangen von Datenpaketen.

Vergewissern Sie sich vor dem Laden einer IP-Adresse in die CPU, dass die IP-Adresse Ihrer CPU mit der IP-Adresse Ihres Programmiergeräts kompatibel ist.

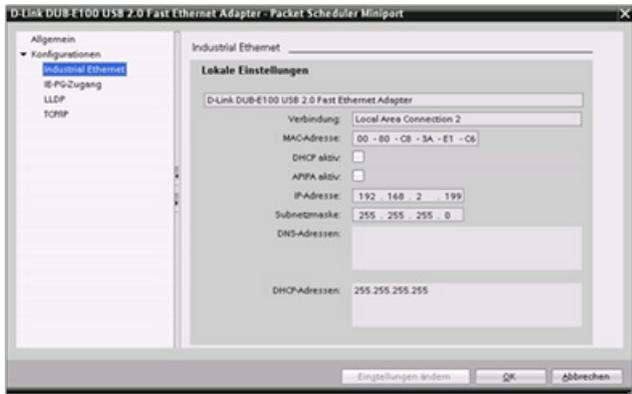
Die IP-Adresse des Programmiergeräts können Sie mithilfe von STEP 7 feststellen:

1. Erweitern Sie in der Projektnavigation den Ordner "Online-Zugänge", um Ihre Netzwerke anzuzeigen.
2. Wählen Sie das Netzwerk für die Verbindung zur CPU aus.
3. Klicken Sie mit der rechten Maustaste auf das spezifische Netzwerk, um das Kontextmenü aufzurufen.
4. Wählen Sie "Eigenschaften".

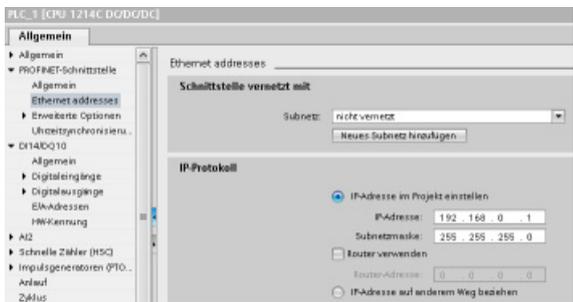
Hinweis

Die IP-Adresse der CPU muss mit der IP-Adresse und Subnetzmaske des Programmiergeräts kompatibel sein. Die geeignete IP-Adresse und Subnetzmaske für Ihre CPU erfahren Sie von Ihrem Netzwerkspezialisten.

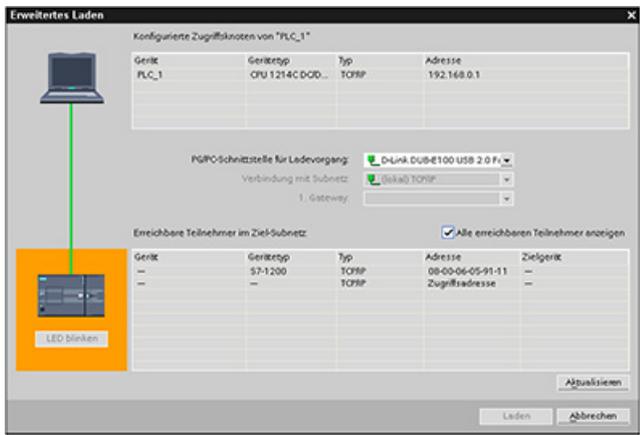
Einfache Erstellung der Gerätekonfiguration
 5.6 Konfigurieren der IP-Adresse der CPU



Im Fenster "Eigenschaften" werden die Einstellungen für das Programmiergerät angezeigt.



Geben Sie nach Ermittlung der IP-Adresse und Subnetzmaske für die CPU die IP-Adresse für die CPU und den Router (falls zutreffend) ein. Beachten Sie für weitere Informationen das *S7-1200 Systemhandbuch*.



Nach Abschluss der Konfiguration laden Sie das Projekt in die CPU. Die IP-Adressen für die CPU und ggf. den Router werden beim Laden des Projekts eingerichtet.

5.7 Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach

Die CPU bietet vier 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 Voreinstellung der CPU ist "ohne Einschränkung" und "ohne Passwortschutz". Um den Zugang zu einer CPU zu schützen, müssen Sie die Eigenschaften der CPU einrichten und das Passwort eingeben.

Wenn Sie ein Netzpasswort eingeben, dann 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 5-4 Schutzstufen der CPU

Schutzstufe	Zugangsbeschränkungen
Vollzugriff (kein Schutz)	Uneingeschränkter Zugriff ohne Passwortschutz.
Lesezugriff	HMI-Zugriff und uneingeschränkte Kommunikation zwischen CPUs ohne Passwortschutz. Ein Passwort ist für Änderungen (Schreibzugriffe) in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich.
HMI-Zugriff	HMI-Zugriff und uneingeschränkte Kommunikation zwischen CPUs ohne Passwortschutz. Ein Passwort ist zum Lesen der Daten in der CPU, für Änderungen (Schreiben) in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich.
Kein Zugriff (kompletter Schutz)	Kein Zugriff ohne Passwortschutz. Ein Passwort ist für den HMI-Zugriff, zum Lesen der Daten in der CPU, für das Ändern (Schreiben) von Daten in der CPU und für den Wechsel des Betriebszustands der CPU (RUN/STOP) erforderlich.

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", um die Schutzstufe auszuwählen und Passwörter einzugeben.

Zugriffsstufe	Zugriff			Zugriffserlaubnis	
	HMI	Lesen	Schreiben	Passwort	Bestätigung
<input type="radio"/> Vollzugriff (kein Schutz)	✓	✓	✓	*****	*****
<input type="radio"/> Lesezugriff	✓	✓		*****	*****
<input checked="" type="radio"/> HMI-Zugriff	✓				
<input type="radio"/> Kein Zugriff (kompletter Schutz)					

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, 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.

! WARNUNG

Nicht berechtigter Zugriff auf eine geschützte CPU

Anwender mit vollständigen CPU-Zugriffsrechten 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 tödlichen oder schweren Verletzungen und/oder zu Sachschaden 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 durch starke Passwörter. Starke Passwörter sind mindestens acht Zeichen lang, bestehen aus Buchstaben, Zahlen und Sonderzeichen, sind keine Wörter, die in einem Wörterbuch gefunden werden können, und sind keine Namen oder Kennungen, die sich aus persönlichen Daten ableiten lassen. Halten Sie das Passwort geheim und ändern Sie es häufig.
- 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.

Verbindungsmechanismen

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. Dies gilt z. B. für:

- PUT/GET-, FETCH/WRITE- oder FTP-Zugriff 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.

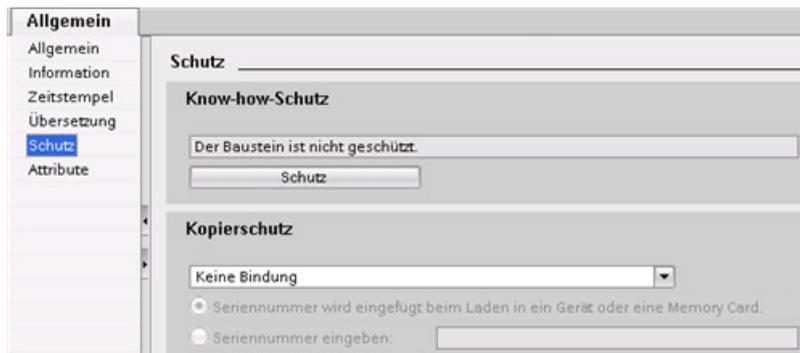
5.7.1 Knowhow-Schutz

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".

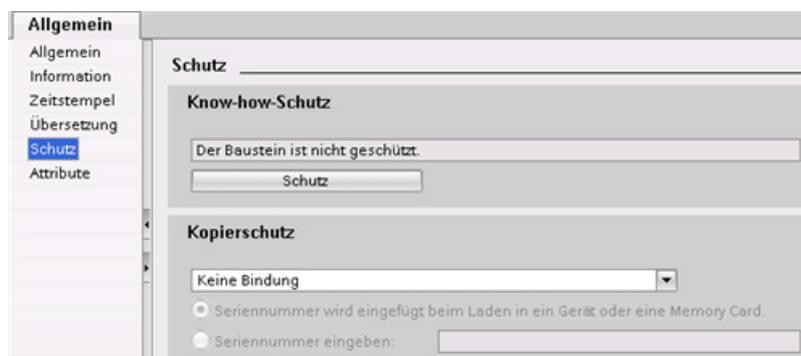


5.7.2 Kopierschutz

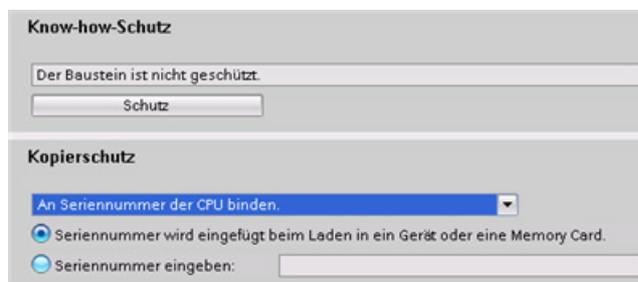
Eine weitere Sicherheitsfunktion ermöglicht Ihnen, Programmbausteine mit einer bestimmten Memory Card oder CPU zu verknüpfen. Diese Funktion ist vor allem zum Schutz geistigen Eigentums nützlich. Wird ein Programmbaustein mit einem bestimmten Gerät verknüpft, so ist die Verwendung des Programms oder Codebausteins nur in Verbindung mit einer bestimmten Memory Card oder CPU möglich. Mit dieser Funktion kann ein Programm oder ein Codebaustein elektronisch (zum Beispiel über Internet oder E-Mail) oder durch Versenden eines Speichermoduls verschickt werden. Kopierschutz ist verfügbar für OBs, FBs und FCs.

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 Option aus, um den Codebaustein mit einer Memory Card oder einer bestimmten CPU zu verknüpfen.



3. Wählen Sie die Art des Kopierschutzes aus und geben Sie die Seriennummer der Memory Card oder CPU ein.

Hinweis

Bei der Seriennummer ist die Groß- und Kleinschreibung zu beachten.

Programmierung leicht gemacht

6.1 Einfaches Entwerfen Ihres Anwenderprogramms

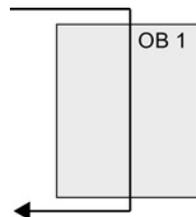
Beim Erstellen eines Anwenderprogramms für Automatisierungslösungen fügen Sie die Anweisungen des Programms in Codebausteine (OB, FB oder FC) ein.

Art der Struktur für das Anwenderprogramm wählen

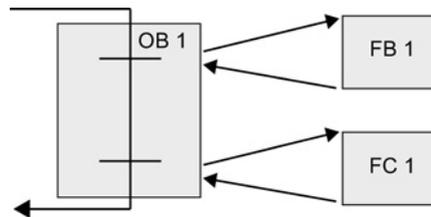
Je nach den Anforderungen Ihrer Anwendung können Sie eine lineare oder eine modulare Struktur für Ihr Anwenderprogramm wählen.

- Ein lineares Programm führt alle Anweisungen für Ihre Automatisierungsaufgaben nacheinander aus. Typischerweise werden bei einem linearen Programm alle Programmanweisungen in einem einzigen Programmzyklus-OB (wie OB 1) abgelegt, anhand dessen das Programm zyklisch abgearbeitet wird.
- 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 funktionalen Aufgaben 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:

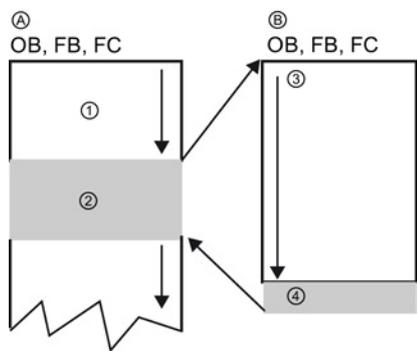


Modulare Struktur:



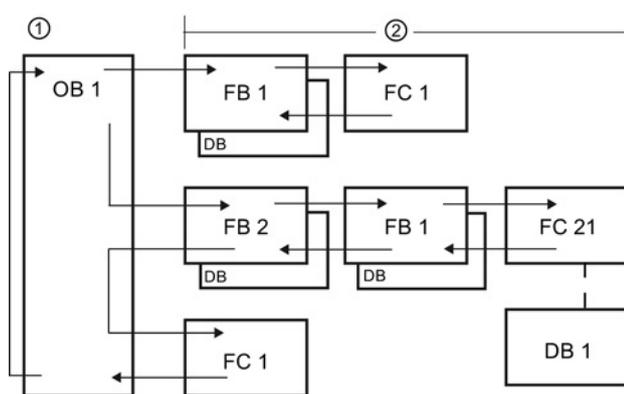
Modulare Codebausteine erstellen Sie durch den Entwurf von FBs und FCs für die Ausführung allgemeiner Aufgaben. Anschließend strukturieren Sie Ihr Programm, indem andere Codebausteine wiederum die wiederverwendbaren Module aufrufen. Der aufrufende Baustein gibt gerätespezifische Parameter an den aufgerufenen 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.

Sie können einen OB auch einem Unterbrechungsereignis zuordnen. Wenn das Ereignis auftritt, führt die CPU den Programmcode im zugehörigen OB aus. Nach der vollständigen Ausführung des OBs nimmt die CPU die Ausführung des Anwenderprogramms an dem Punkt wieder auf, an dem das Unterbrechungsereignis auftrat. Dies kann an jeder beliebigen Stelle im Zyklus sein.



- A Aufrufender Baustein (oder unterbrochener Baustein)
- B Aufgerufener FB oder FC (oder unterbrechender OB)
- ① Programmausführung
- ② Anweisung (oder unterbrechendes Ereignis), die die Ausführung eines anderen Bausteins auslöst
- ③ Programmausführung
- ④ Bausteinende (Rückkehr zum aufrufenden Baustein)

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.



- ① Zyklusbeginn
- ② Schachtelungstiefe

Durch das Anlegen allgemeiner Codebausteine, die im Anwenderprogramm mehrfach genutzt werden können, lassen sich Entwurf und Implementierung des Anwenderprogramms vereinfachen.

- 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 vereinfachen und beschleunigen auch Anpassungen oder Änderungen des Programmcodes.
- 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.
- Ein modularer Aufbau, der sich an bestimmten funktionalen Aufgaben orientiert, verkürzt normalerweise die Zeit bis zur Inbetriebnahme der fertiggestellten Anwendung.

6.1.1 OBs für die Organisation Ihres Anwenderprogramms

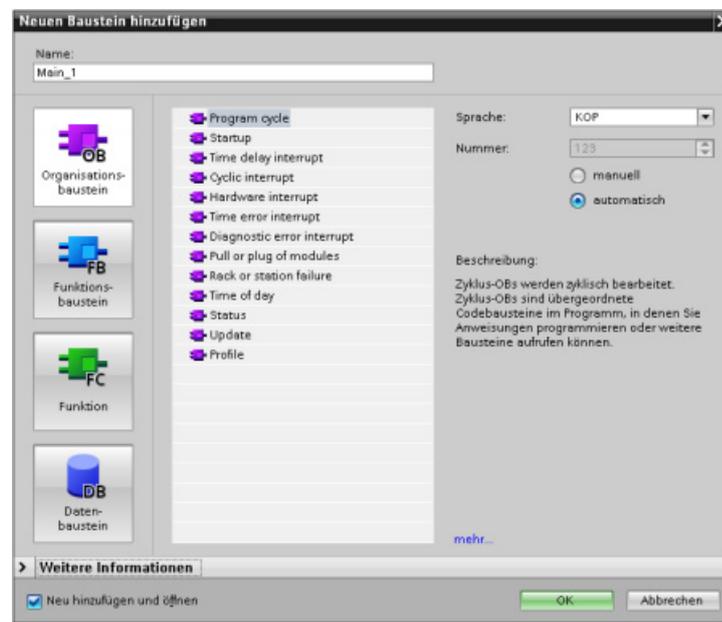
Organisationsbausteine dienen zur Strukturierung Ihres Programms. Sie bilden die Schnittstelle zwischen dem Betriebssystem und dem Anwenderprogramm. OBs sind ereignisgesteuert. Die Ausführung eines OBs durch das Zielsystem wird von einem Ereignis, wie z. B. einem Diagnosealarm oder einem Zeitintervall, angestoßen. Einige OBs haben voreingestellte Startereignisse und Anlaufverhalten.

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.

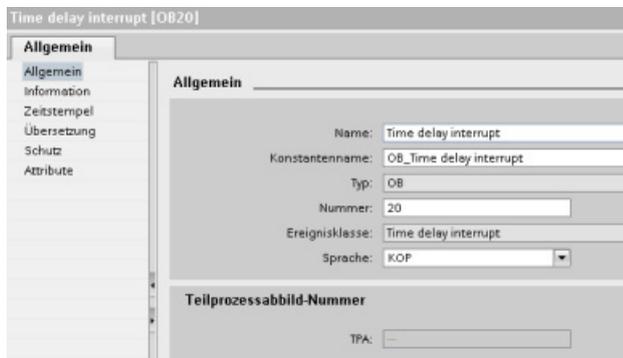
Die CPU ermittelt die Reihenfolge für die Bearbeitung von Alarmereignissen anhand der Prioritäten. Sie können einer Prioritätsklasse mehrere Alarmereignisse zuweisen. Weitere Informationen finden Sie in den Themen zu Organisationsbausteinen (Seite 63) und zur Ausführung des Anwenderprogramms (Seite 62).

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 62)".

6.1.2 FBs und FCs vereinfachen die Programmierung der modularen Aufgaben

Eine Funktion (FC) ist eine Art Unterprogramm. Eine FC ist ein Codebaustein, der typischerweise eine bestimmte Anweisung mit einer Anzahl von Eingangswerten durchführt. Die FC speichert die Ergebnisse dieser Operation an bestimmten Speicheradressen. Mit FCs können Sie die folgenden Tätigkeiten ausführen:

- Standardoperationen und mehrfach verwendbare Operationen, z. B. arithmetische Berechnungen
- Durchführung funktionaler Aufgaben, z. B. für einzelne Steuerungsvorgänge mit Bitverknüpfungen

Eine FC kann auch mehrmals an verschiedenen Stellen eines Programms aufgerufen werden. Diese Wiederverwendung vereinfacht die Programmierung häufig wiederkehrender Aufgaben.

Im Unterschied zu einem FB ist einer FC kein Instanz-DB zugeordnet. Die FC verwendet den temporären Speicher (L) für die Daten, die zur Berechnung der Operation benötigt werden. Die temporären Daten werden nicht gespeichert. Um Daten für die Verwendung nach der Ausführung der FC zu speichern, muss der Ausgangswert einem globalen Speicherplatz, wie z. B. dem Merkerspeicher, oder einem globalen DB zugewiesen werden.

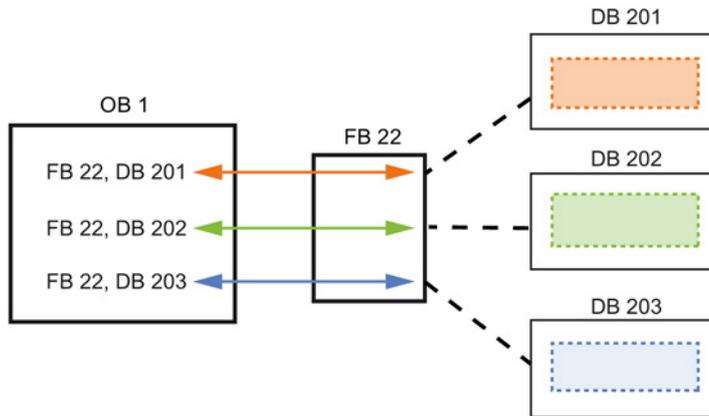
Ein Funktionsbaustein (FB) ist eine Art Unterprogramm mit Speicher. Ein Funktionsbaustein (FB) ist ein Codebaustein, dessen Aufrufe mit Bausteinparametern programmiert werden können. Ein FB speichert die Eingangs- (IN), Ausgangs- (OUT) und Durchgangparameter (IN_OUT) in einem variablen Speicherbereich, der sich in einem Datenbaustein (DB) oder "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.

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 öffnen Sie gleichzeitig einen Instanz-DB, in dem die Werte der Bausteinparameter und die statischen Lokaldaten für diesen Aufruf oder diese "Instanz" des FBs gespeichert sind. Diese Werte werden nach dem Beenden des FBs im Instanz-DB gespeichert.

Sie können den Parametern in der FB-Schnittstelle Startwerte zuweisen. Diese Werte werden an den zugehörigen Instanz-DB übertragen. Weisen Sie keine Parameter zu, so werden die im Instanz-DB gespeicherten Werte verwendet. In einigen Fällen müssen Sie die Parameter zuweisen.

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. 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.

Die folgende Abbildung zeigt einen OB, der einen FB drei Mal 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.

6.1.3 Datenbausteine sorgen für die unkomplizierte Speicherung von Programmdaten

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).

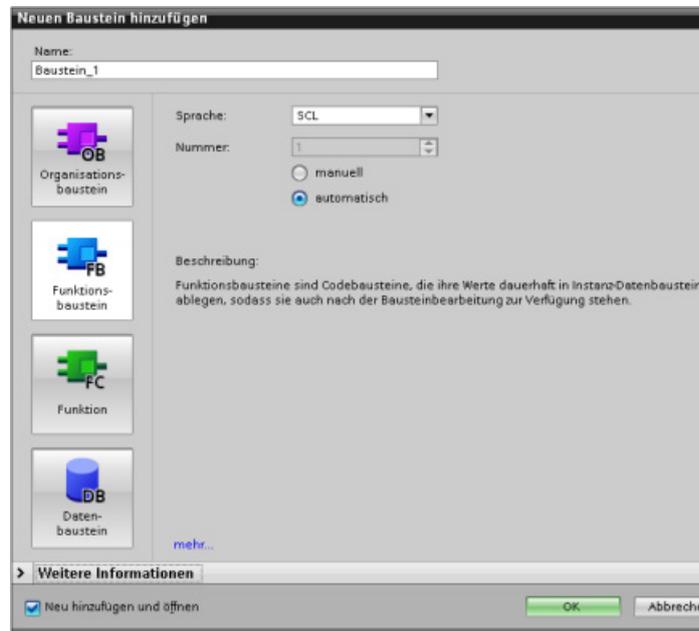
Ihr Anwenderprogramm kann Daten in den speziellen Speicherbereichen der CPU, z. B. für Eingänge (E), Ausgänge (A) und Merker (M), speichern. Außerdem können Sie einen Datenbaustein (DB) für den schnellen Zugriff auf die im Programm selbst gespeicherten Daten verwenden.

Die in einem DB gespeicherten Daten werden beim Schließen des Datenbausteins oder wenn die Ausführung des zugehörigen Codebausteins beendet ist, 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 nicht im Instanz-DB gespeichert.

Obwohl der Instanz-DB die Daten für einen bestimmten FB enthält, kann jeder Codebaustein auf die Daten eines Instanz-DBs zugreifen.

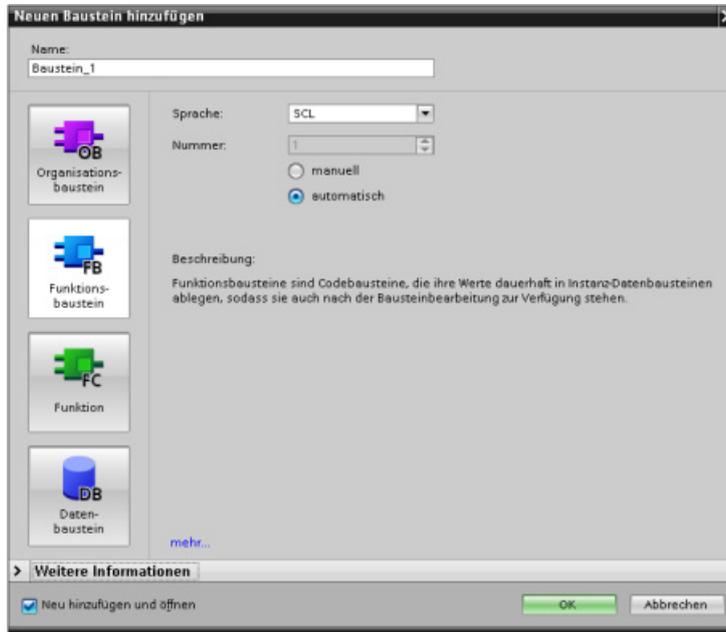
6.1.4 Neuen Codebaustein anlegen



1. Öffnen Sie den Ordner "Programmbausteine".
2. Doppelklicken Sie auf "Neuen Baustein hinzufügen".
3. Wählen Sie im Dialog "Neuen Baustein hinzufügen" den Typ des hinzuzufügenden Bausteins. Klicken Sie z. B. auf das Symbol "Funktion (FC)", um eine FC hinzuzufügen.
4. Wählen Sie die Programmiersprache für den Codebaustein durch Auswahl von "KOP" in der Klappliste.
5. Klicken Sie auf "OK", um den Baustein zum Projekt hinzuzufügen.

Bei Auswahl des Optionskästchens "Neu hinzufügen und öffnen" (Standardeinstellung) wird der Codebaustein in der Projektansicht geöffnet.

6.1.5 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.

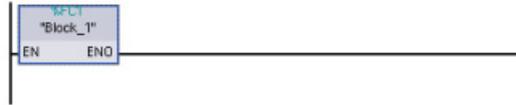
Bei Auswahl des Kontrollkästchens "Neu hinzufügen und öffnen" (Standardeinstellung) wird der Codebaustein in der Projektansicht geöffnet.

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.

6.1.6 Einen Codebaustein aus einem anderen Codebaustein aufrufen



Es ist ohne großen Aufwand möglich, mit einem Codebaustein (OB, FB oder FC) im Anwenderprogramm einen FB oder FC aufzurufen.

1. Öffnen Sie den Codebaustein, der den anderen Baustein aufrufen soll.
2. Selektieren Sie in der Projektnavigation den aufzurufenden Codebaustein.
3. Ziehen Sie den Baustein mit der Maus in das ausgewählte Netzwerk, um einen Aufruf des Codebausteins zu erstellen.

Hinweis

Mit dem Anwenderprogramm können Sie keinen OB aufrufen, da OBs ereignisgesteuert (Seite 63) sind. Die Ausführung eines OBs wird von der CPU in Reaktion auf den Empfang eines Ereignisses gestartet.

6.2 Anwenderfreundliche Programmiersprachen

STEP 7 bietet die folgenden Standardprogrammiersprachen für die S7-1200:

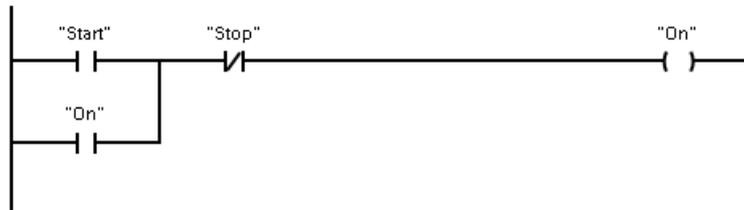
- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen.
- FUP (Funktionsplan) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra basiert.
- SCL (Structured Control Language) ist eine textbasierte, höhere Programmiersprache.

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.

6.2.1 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.

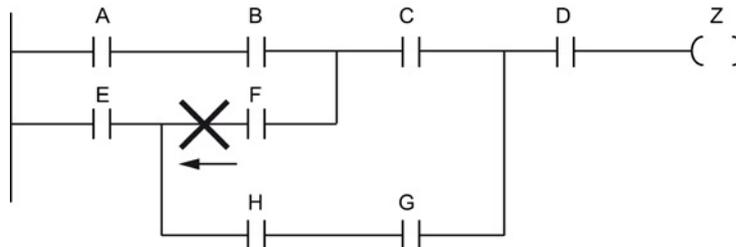
STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem KOP-Netzwerk nicht.

Hinweis

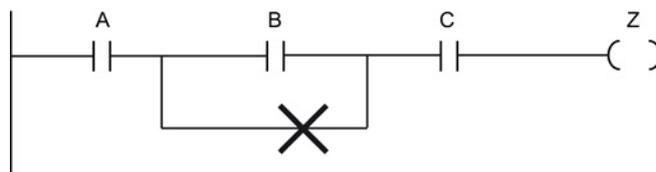
Jedes KOP-Netzwerk muss mit einer Spule oder einer Box abgeschlossen werden.

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.

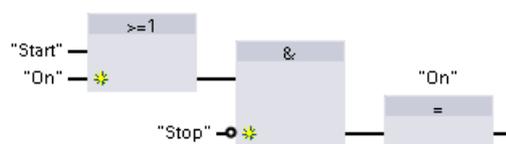


- Sie können keine Verzweigung anlegen, die einen Kurzschluss verursachen würde.



6.2.2 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.

6.2.3 Überblick über SCL

SCL (Structured Control Language) ist eine höhere, auf PASCAL basierende Programmiersprache für die SIMATIC S7-CPU's. SCL unterstützt die Baueinstructur von STEP 7. Sie können in SCL geschriebene Programmbausteine mit in KOP und FUP geschriebenen Programmbausteinen kombinieren.

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 arbeitet mit standardmäßigen PASCAL-Programmsteuerungsoperationen wie z. B. IF-THEN-ELSE, CASE, REPEAT-UNTIL, GOTO und RETURN. Für syntaktische Elemente der Programmiersprache SCL können Sie beliebige PASCAL-Verweise verwenden. Viele der anderen Anweisungen für SCL wie Zeiten und Zähler entsprechen den Anweisungen in KOP und FUP.

Da SCL wie PASCAL Steuerstrukturen für bedingte Verarbeitung, Schleifen und Verschachtelung bietet, lassen sich komplexe Algorithmen in SCL leichter realisieren als in KOP oder FUP.

Die folgenden Beispiele zeigen verschiedene Ausdrücke für verschiedene Einsatzzwecke:

```
"C" := #A+#B;
```

Weist einer Variablen zwei lokale Variablen zu.

```
"Data_block_1".Tag := #A;
```

Zuweisung zu einer Datenbausteinvariablen

```
IF #A > #B THEN "C" := #A;
```

Bedingung für die IF-THEN -Anweisung

```
"C" := SQRT (SQR (#A) + SQR (#B));
```

Parameter für die SQRT -Anweisung

Als höhere Programmiersprache nutzt SCL Standardanweisungen für grundlegende Aufgaben:

- Zuweisungsanweisung: :=
- Mathematische Funktionen: +, -, * und /
- Adressierung globaler Variablen: "<Variablenname>" (in doppelte Anführungszeichen eingeschlossener Variablenname oder Datenbausteinname)
- Adressierung lokaler Variablen: #<Variablenname> (Variablenname mit vorangestelltem "#" -Symbol)
- Absolute Adressierung: %<absolute Adresse>, zum Beispiel %E0.0 oder %MW10

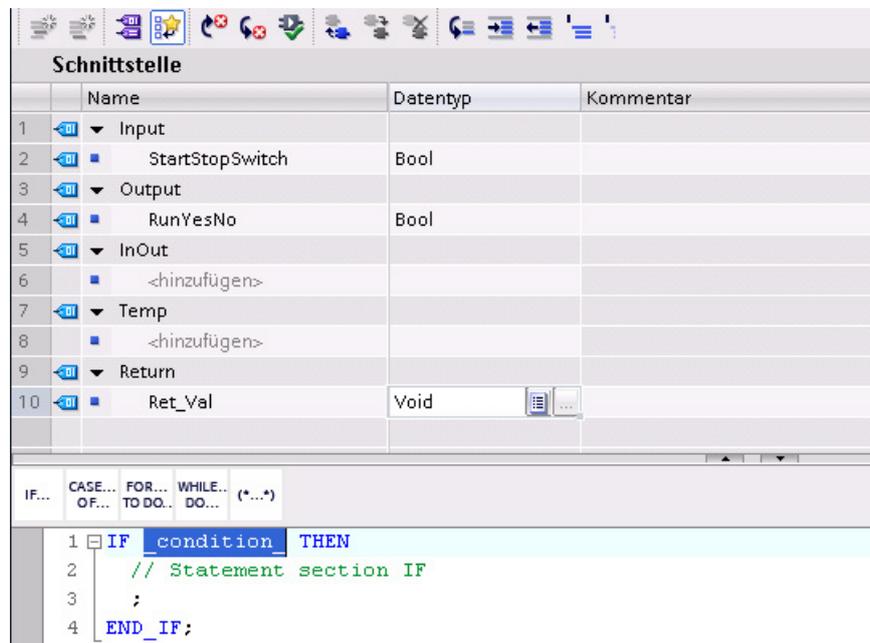
Arithmetische Operatoren können verschiedene numerische Datentypen verarbeiten. Der Datentyp des Ergebnisses wird vom Datentyp des höchstwertigen Operanden festgelegt. Beispiel: Eine Multiplikationsoperation mit einem Operanden vom Typ INT und einem Operanden vom Typ REAL ergibt einen Wert vom Typ REAL als Ergebnis.

6.2.4 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

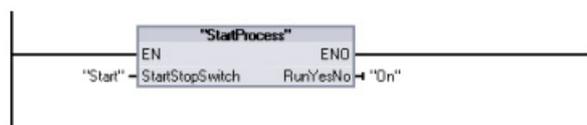
Sie geben den SCL-Code für Ihre Anweisung direkt in den Codeabschnitt ein. Der Editor enthält Schaltflächen für gängige Codekonstruktionen und Kommentare. Um komplexere Anweisungen anzulegen, ziehen Sie die SCL-Anweisungen einfach mit der Maus aus dem Anweisungsverzeichnis in Ihr Programm. Ferner können Sie in jedem Texteditor ein SCL-Programm anlegen und die Datei dann in STEP 7 importieren.



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.

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.

6.3 Leistungsstarke Anweisungen erleichtern die Programmierung

6.3.1 Bereitstellung der erwartbaren grundlegenden Operationen

Bitverknüpfungsanweisungen

Die Grundlage von Bitverknüpfungsanweisungen sind Kontakte und Spulen. Kontakte lesen den Status eines Bits aus, während Spulen den Status der Operation in ein Bit schreiben.



Kontakte dienen zum Testen des Binärstatus des Bits. Ergebnis dieses Tests ist entweder "Signalfluss" bei Ein (1) oder "kein Signalfluss" bei Aus (0).

Der Zustand der Spule zeigt den Zustand der vorherigen Verknüpfung an.

Wenn Sie eine Spule mit derselben Adresse an mehreren Stellen im Programm verwenden, so hängt vom Ergebnis der letzten Berechnung im Anwenderprogramm der Zustand des Werts ab, der während der Aktualisierung der Ausgänge in den physikalischen Ausgang geschrieben wird.

Schließerkontakt	Öffnerkontakt	Der Schließer ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 1 ist.
"IN"	"IN"	Der Öffner ist geschlossen (EIN), wenn der zugewiesene Bitwert gleich 0 ist.

Die Grundstruktur einer Bitverknüpfungsoperation ist entweder eine UND- oder eine ODER-Verknüpfung. In Reihe geschaltete Kontakte bilden logische UND-Verknüpfungen. Parallel geschaltete Kontakte bilden logische ODER-Verknüpfungen.

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 des PLC-Geräts verschaltet. Die CPU fragt die verschalteten Eingangssignale ab und aktualisiert die entsprechenden Zustandswerte im Prozessabbild der Eingänge.

Sie geben das direkte Lesen eines physischen Eingangs an, indem Sie nach der Variablen für einen Eingang den Code ":P" eingeben (Beispiel: "Motor_Start:P" oder "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.

Ausgangsspule Invertierte
Ausgangsspule



Beachten Sie die folgenden Ausgangsergebnisse für Signalfloss durch Ausgangs- und invertierte Ausgangsspulen:

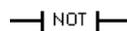
- Ist ein Signalfloss durch eine Ausgangsspule vorhanden, so wird das Ausgangsbit auf 1 gesetzt.
- Ist kein Signalfloss durch eine Ausgangsspule vorhanden, so wird das Ausgangsbit auf 0 gesetzt.
- Ist ein Signalfloss durch eine invertierte Ausgangsspule vorhanden, so wird das Ausgangsbit auf 0 gesetzt.
- Ist kein Signalfloss durch eine invertierte Ausgangsspule vorhanden, so wird das Ausgangsbit auf 1 gesetzt.

Die Anweisung für den Spulenausgang schreibt einen Wert in ein Ausgangsbit. Nutzt das angegebene Ausgangsbit die Speichererkennung A, so schaltet die CPU das Ausgangsbit im Prozessabbildregister ein oder aus und setzt das angegebene Bit jeweils entsprechend dem Signalfloss. Die Ausgangssignale für Ihre Steuerstellglieder werden mit den Ausgangsklemmen der PLC verschaltet. Im Betriebszustand RUN fragt die CPU die Eingangssignale ab, verarbeitet die Eingangszustände gemäß der Programmlogik und reagiert dann, indem sie die neuen Ausgangswerte im Prozessabbild der Ausgänge setzt. Nach jeder Programmausführung überträgt die CPU die im Prozessabbild gespeicherte neue Reaktion auf den Ausgangszustand zu den verschalteten Ausgangsklemmen.

Sie geben das direkte Schreiben in einen physischen Ausgang an, indem Sie nach der Variablen für einen Ausgang den Code `":P"` angeben (Beispiel: `"Motor_On:P"` oder `"A3.4:P"`). Beim direkten Schreiben werden die Bitdatenwerte in den Ausgang des Prozessabbilds und direkt in den physischen Ausgang geschrieben.

Spulen sind nicht auf den Abschluss eines Netzwerks beschränkt. Sie können eine Spule in einem Strompfad des KOP-Netzwerks, zwischen Kontakten oder anderen Anweisungen einfügen.

Kontaktinvertierer
NOT (KOP)



UND-Box mit einem
invertierten Logikeingang
(FUP)



UND-Box mit invertiertem
Logikeingang und -ausgang (FUP)

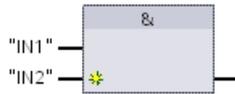


Der KOP-Kontakt NOT invertiert den logischen Zustand des Signalflosseingangs.

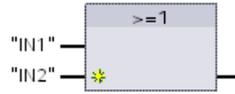
- Ist kein Signalfloss zum NOT-Kontakt vorhanden, so steht ein Signalfloss am Ausgang an.
- Ist ein Signalfloss zum NOT-Kontakt vorhanden, so steht kein Signalfloss am Ausgang an.

Bei der FUP-Programmierung können Sie die Funktion "Binäreingang invertieren" aus der Funktionsleiste "Favoriten" oder dem Anweisungsverzeichnis auf einen Eingang oder einen Ausgang ziehen, um einen Logikinvertierer für diesen Box-Anschluss zu erstellen.

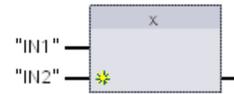
UND-Box (FUP)



ODER-Box (FUP)



XOR-Box (FUP)



- Damit der Ausgang WAHR ist, müssen alle Eingänge einer UND-Box WAHR sein
- Damit der Ausgang WAHR ist, muss ein beliebiger Eingang einer ODER-Box WAHR sein
- Damit der Ausgang WAHR ist, muss eine ungerade Anzahl der Eingänge einer XOR-Box WAHR sein

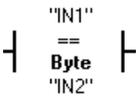
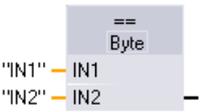
Bei der FUP-Programmierung werden die KOP-Netzwerke aus Kontakten durch die Box-Netzwerke UND (&), ODER (>=1) und Exklusiv ODER (x) dargestellt, in denen Sie Bitwerte für die Ein- und -Ausgänge der Box angeben können. Sie können ferner Verschaltungen mit anderen Logik-Boxen herstellen und so Ihre eigene Verschaltungslogik erstellen. Nachdem die Box in Ihrem Netzwerk platziert ist, können Sie die Funktion "Binäreingang 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.

6.3.2 Anweisungen für Vergleichen und Verschieben

Mit den Vergleichsanweisungen können Sie zwei Werte des gleichen Datentyps miteinander vergleichen.

Tabelle 6- 1 Vergleichsanweisungen

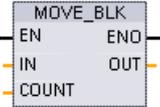
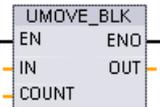
Anweisung	SCL	Beschreibung
<p>KOP:</p> 	<pre>out := in1 = in2; out := in1 <> in2; out := in1 >= in2; out := in1 <= in2; out := in1 > in2; out := in1 < in2;</pre>	<ul style="list-style-type: none"> • Gleich (==): Der Vergleich ist wahr, wenn IN1 gleich IN2 ist. • Ungleich (<>): Der Vergleich ist wahr, wenn IN1 nicht gleich IN2 ist. • Größer als oder gleich (>=): Der Vergleich ist wahr, wenn IN1 größer als oder gleich IN2 ist. • Kleiner als oder gleich (<=): Der Vergleich ist wahr, wenn IN1 kleiner als oder gleich IN2 ist. • Größer als (>): Der Vergleich ist wahr, wenn IN1 größer als IN2 ist. • Kleiner als (<): Der Vergleich ist wahr, wenn IN1 kleiner als IN2 ist.
<p>FUP:</p> 		

¹ In KOP und FUP: Der Kontakt wird aktiviert (KOP) bzw. der Box-Ausgang ist WAHR (FUP), wenn der Vergleich WAHR ist.

Mit den Anweisungen zum Verschieben kopieren Sie Datenelemente in eine neue Adresse im Speicher und können die Daten von einem Datentyp in einen anderen umwandeln. Die Quelldaten werden dadurch nicht verändert.

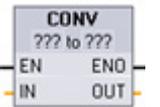
- MOVE kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse. Um einen weiteren Ausgang hinzuzufügen, klicken Sie auf das Symbol neben dem Parameter OUT1.
- MOVE_BLK (Bereich kopieren) und UMOVE_BLK (Bereich ununterbrechbar kopieren) kopieren einen Baustein mit Datenelementen in eine neue Adresse. 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-Variablen-tabelle zugewiesen ist.

Tabelle 6-2 Anweisungen MOVE, MOVE_BLK und UMOVE_BLK

KOP/FUP	SCL	Beschreibung
	<pre>out1 := in;</pre>	<p>Kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse oder in mehrere Adressen. Um einen weiteren Ausgang in KOP oder FUP hinzuzufügen, klicken Sie auf das Symbol neben dem Ausgangsparameter. In SCL verwenden Sie mehrere Zuweisungsanweisungen. Sie können auch eine der Schleifenkonstruktionen verwenden.</p>
	<pre>MOVE_BLK(in:=_variant_in, count:=_uint_in, out=>_variant_out);</pre>	<p>Unterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.</p>
	<pre>UMOVE_BLK(in:=_variant_in, count:=_uint_in out=>_variant_out);</pre>	<p>Ununterbrechbare Übertragung, die einen Bereich mit Datenelementen in eine neue Adresse kopiert.</p>

6.3.3 Umwandlungsanweisungen

Tabelle 6-3 Umwandlungsanweisungen

KOP/FUP	SCL	Beschreibung
	<pre>out := <data type in>_TO_<data type out>(in);</pre>	<p>Konvertiert ein Datenelement von einem Datentyp in einen anderen Datentyp.</p>

- 1 In KOP und FUP: Klicken Sie unterhalb des Box-Namens und wählen Sie in der Klappliste einen Datentyp aus. Nach Auswahl des Quelldatentyps (Konvertieren aus) wird in der Klappliste eine Reihe möglicher Umwandlungen (Konvertieren in) angezeigt.
- 2 In SCL: Konstruieren Sie die Umwandlungsanweisung durch Angabe des Datentyps für den Eingangsparameter (in) und den Ausgangsparameter (out). Beispiel: DWORD_TO_REAL konvertiert einen Doppelwortwert in einen Realzahlenwert.

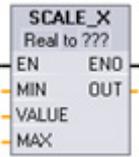
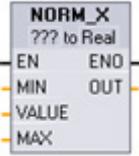
Tabelle 6- 4 Anweisungen Round und Truncate

KOP/FUP	SCL	Beschreibung
	<pre>out := ROUND (in);</pre>	<p>konvertiert eine Realzahl (Real oder LReal) in eine Ganzzahl. Die Anweisung rundet die Realzahl auf den nächsten ganzzahligen Wert (IEEE - runden). Wenn die Zahl genau die Hälfte der Spanne zwischen zwei Ganzzahlen ist (z. B. 10,5), rundet die Anweisung die Zahl auf die gerade Ganzzahl. Beispiel:</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12 <p>In KOP/FUP klicken Sie im Anweisungsfeld auf die "???", um den Datentyp für den Ausgang auszuwählen, z. B. "DInt". In SCL ist der Standard-Ausgangsdatentyp DINT. Um auf einen anderen Datentyp zu runden, geben Sie den Anweisungsnamen mit dem expliziten Namen des Datentyps ein, z. B. ROUND_REAL oder ROUND_LREAL.</p>
	<pre>out := TRUNC (in);</pre>	<p>konvertiert eine Realzahl (Real oder LReal) in eine Ganzzahl. Die Nachkommastellen der Realzahl werden auf Null verkürzt (IEEE- runden auf Null).</p>

Tabelle 6- 5 Anweisungen CEIL und Floor

KOP/FUP	SCL	Beschreibung
	<pre>out := CEIL (in);</pre>	<p>Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die größer oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf +unendlich).</p>
	<pre>out := FLOOR (in);</pre>	<p>Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die kleiner oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf -unendlich).</p>

Tabelle 6-6 Anweisungen SCALE_X und NORM_X

KOP/FUP	SCL	Beschreibung
	<pre>out := SCALE_X(min, := _undef_in_ value := _real_in_, max := _undef_in_);</pre>	<p>Skaliert den normalisierten Realparameter VALUE ($0,0 \leq \text{VALUE} \leq 1,0$) in den mit den Parametern MIN und MAX vorgegebenen Datentyp und Wertebereich: $\text{OUT} = \text{VALUE} (\text{MAX} - \text{MIN}) + \text{MIN}$</p>
	<pre>out := NORM_X(min := _undef_in_ value := _undef_in_, max := _undef_in_);</pre>	<p>Normalisiert den Parameter VALUE innerhalb des von den Parametern MIN und MAX angegebenen Wertebereichs: $\text{OUT} = (\text{VALUE} - \text{MIN}) / (\text{MAX} - \text{MIN})$, dabei ist ($0,0 \leq \text{OUT} \leq 1,0$)</p>

¹ Entsprechung in SCL: `out := value (max-min) + min;` ² Entsprechung in SCL: `out := (value-min)/(max-min);`

6.3.4 Mathematik ganz einfach mit der Anweisung Calculate

Tabelle 6-7 Anweisung CALCULATE

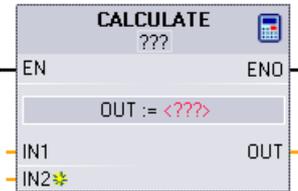
KOP/FUP	SCL	Beschreibung
	<p>Verwenden Sie die herkömmlichen mathematischen SCL-Ausdrücke, um die Gleichung zu erstellen.</p>	<p>Mit der Anweisung CALCULATE können Sie eine arithmetische Funktion erstellen, die an Eingängen ausgeführt wird (IN1, IN2, .. INn) und das Ergebnis an OUT zurückgibt, entsprechend der von Ihnen definierten Gleichung.</p> <ul style="list-style-type: none"> Wählen Sie zunächst einen Datentyp aus. Alle Eingänge und der Ausgang müssen denselben Datentyp haben. Um einen weiteren Eingang hinzuzufügen, klicken Sie auf das Symbol am letzten Eingang.

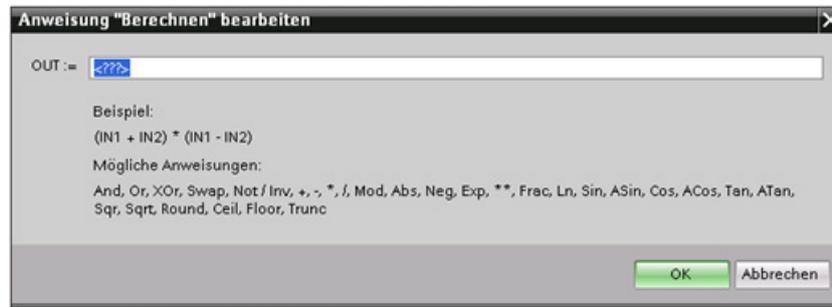
Tabelle 6-8 Datentypen für die Parameter

Parameter	Datentyp ¹
IN1, IN2, ..INn	SInt, Int, DInt, USInt, UInt, UDIInt, Real, LReal, Byte, Word, DWord
OUT	SInt, Int, DInt, USInt, UInt, UDIInt, 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.

Ein Beispiel und eine Liste der möglichen mathematischen Operationen, die Sie verwenden können, wird im unteren Bereich des Editors gezeigt.



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, dann ist ENO = 1. Andernfalls ist ENO = 0.

6.3.5 Zeiten

Die S7-1200 unterstützt die folgenden Zeiten

- Die Zeit TP erzeugt einen Impuls mit einer voreingestellten Dauer.
- Die Zeit TON setzt den Ausgang (Q) nach einer voreingestellten Zeitverzögerung auf EIN.
- Die Zeit TOF setzt den Ausgang (Q) auf EIN und nach einer voreingestellten Zeitverzögerung wieder auf AUS.
- Die Zeit TONR setzt den Ausgang (Q) nach einer voreingestellten Zeitverzögerung auf EIN. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis der Rücksetzeingang (R) der abgelaufenen Zeit angestoßen wird.

In KOP und FUP sind diese Anweisungen entweder als Box-Anweisung oder als Ausgangsspule verfügbar. STEP 7 bietet zudem die folgenden Zeitspulen für KOP und FUP:

- Die Spule PT (voreingestellte Zeit) lädt einen neuen voreingestellten Zeitwert in die angegebene Zeit.
- Die Spule RT (Zeit rücksetzen) setzt die angegebene Zeit zurück.

Die Anzahl der Zeiten, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Jede Zeit belegt 16 Byte im Speicher.

Jede Zeit nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten der Zeit zu speichern. In SCL müssen Sie für die einzelne Zeitanweisung zunächst einen DB erstellen, damit Sie ihn referenzieren können. In KOP und FUP erstellt STEP 7 automatisch den DB, wenn Sie die Anweisung einfügen.

Wenn Sie den DB erstellen, können Sie auch einen Multiinstanz-DB verwenden. Weil sich die Zeitdaten in einem einzigen DB befinden und nicht für jede Zeit ein eigener DB erforderlich ist, verkürzt sich die Verarbeitungszeit der Zeiten. Zwischen den Datenstrukturen der Zeiten im gemeinsam genutzten Multiinstanz-DB gibt es keine Wechselwirkungen.

Tabelle 6- 9 TP (Impuls)

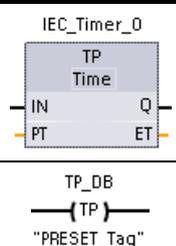
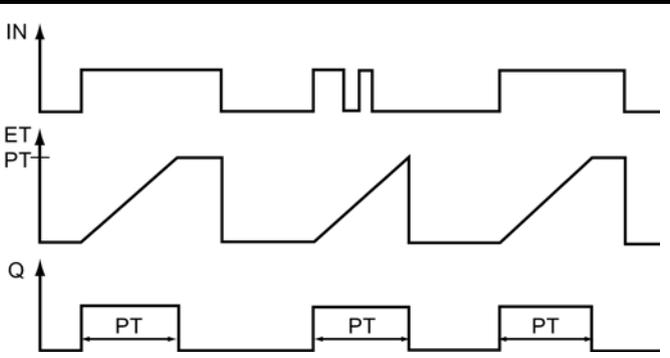
KOP/FUP	SCL	Zeitdiagramm
	<pre>"timer_db".TP(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	

Tabelle 6- 10 TON (Einschaltverzögerung)

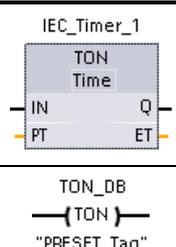
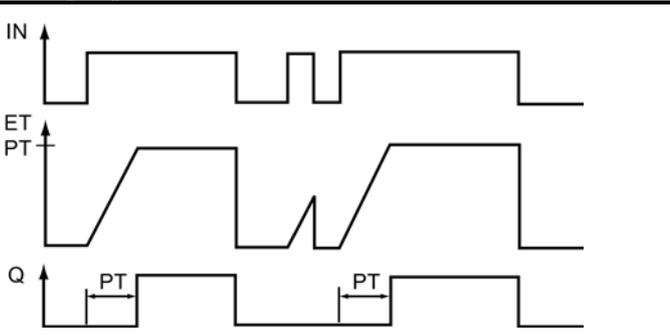
KOP/FUP	SCL	Timingdiagramm
	<pre>"timer_db".TON(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	

Tabelle 6- 11 TOF (Ausschaltverzögerung)

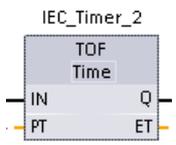
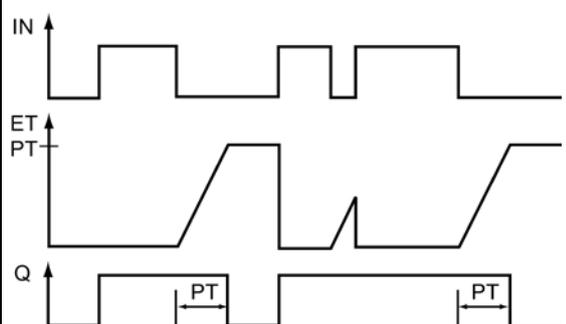
KOP/FUP	SCL	Timingdiagramm
 <p>IEC_Timer_2 TOF Time IN Q PT ET</p>	<pre>"timer_db".TOF(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	 <p>The timing diagram shows three signals: IN, ET, and Q. IN is a square wave. ET is a ramp signal that starts when IN goes high and reaches a plateau when IN goes low. PT is a horizontal line. Q is a square wave that is high when IN is high and low when IN is low, with a delay during the falling edge of IN.</p>
 <p>TONR_DB —(TOF)— "PRESET_Tag"</p>		

Tabelle 6- 12 TONR (Zeit akkumulieren)

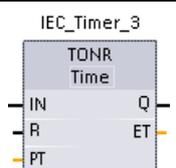
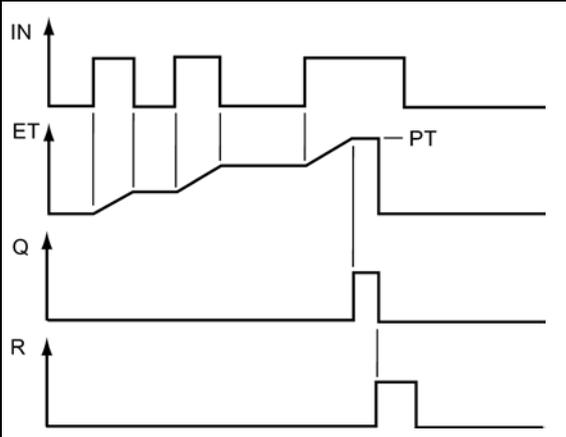
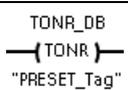
KOP/FUP	SCL	Timingdiagramm
 <p>IEC_Timer_3 TONR Time IN Q R ET PT</p>	<pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	 <p>The timing diagram shows four signals: IN, ET, Q, and R. IN is a square wave. ET is a ramp signal that starts when IN goes high and reaches a plateau when IN goes low. Q is a square wave that is high when IN is high and low when IN is low, with a delay during the falling edge of IN. R is a square wave that is high when IN is high and low when IN is low.</p>
 <p>TONR_DB —(TONR)— "PRESET_Tag"</p>		

Tabelle 6- 13 Spulenanweisungen Zeit voreinstellen -(PT)- und Zeit rücksetzen -(RT)-

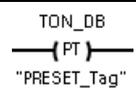
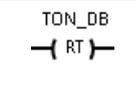
KOP	Beschreibung
 <p>TONR_DB —(PT)— "PRESET_Tag"</p>	<p>Sie können die Spulenanweisungen Zeit voreinstellen -(PT)- und Zeit rücksetzen -(RT)- mit Box-Anweisungen oder Spulen verwenden. Diese Spulenanweisungen können in der Mitte eines Strompfads angeordnet werden. Der Signalzustand des Spulenausgangs ist immer der gleiche wie am Spuleneingang.</p>
 <p>TONR_DB —(RT)—</p>	<ul style="list-style-type: none"> • Wenn die Spule -(PT)- aktiviert wird, wird das Zeitelement PRESET in den Daten des angegebenen DBs IEC_Timer auf die Zeitdauer "PRESET_Tag" gesetzt. • Wenn die Spule -(RT)- aktiviert wird, wird das Zeitelement ELAPSED in den Daten des angegebenen DBs IEC_Timer auf 0 zurückgesetzt.

Tabelle 6- 14 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 Zeitdaten des DBs Zeitspule: Auf das Q-Bit kann nur innerhalb der Zeitdaten des DBs zugegriffen werden.
Box: ET Spule: DBdata.ET	Time	Zeitbox: ET-Boxausgang (abgelaufene Zeit) oder ET-Zeitwert in den Zeitdaten des DBs Zeitspule: Auf das Q-Bit kann nur innerhalb der Zeitdaten des DBs zugegriffen werden.

Tabelle 6- 15 Auswirkung von Wertänderungen in den Parametern PT und IN

Zeit	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.
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 für 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 6- 16 Größe und Bereich des Datentyps TIME

Datentyp	Größe	Gültige Zahlenbereiche ¹
TIME	32 Bit, gespeichert alsDInt -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 Zeitanweisungen nicht verwendet werden. Negative Werte für PT (voreingestellte Zeit) werden bei Ausführung der Zeitanweisung auf Null gesetzt. ET (abgelaufene Zeit) ist immer ein positiver Wert.

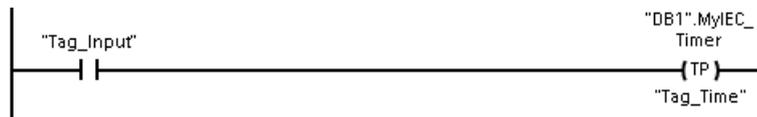
Programmierung von Zeiten

Die folgenden Konsequenzen von Zeiten sind bei der Planung und Erstellung Ihres Anwenderprogramms zu berücksichtigen:

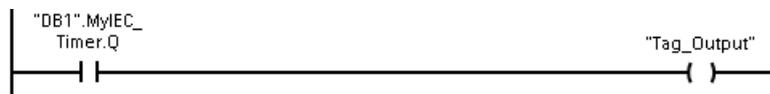
- Sie können mehrere Aktualisierungen einer Zeit im gleichen Zyklus haben. Die Zeit wird bei jeder Ausführung der Zeitanweisung (TP, TON, TOF, TONR) aktualisiert sowie jedes Mal, wenn das Element ELAPSED oder Q der Zeitstruktur als Parameter einer anderen ausgeführten Anweisung verwendet wird. Dies ist 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 ein, 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, sodass 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 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.

Die Zeitspulen -(TP)-, -(TON)-, -(TOF)- und -(TONR)- müssen immer die letzte Anweisung in einem Netzwerk sein. 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 ausgeführt wird, ist der Zustand von DB1.MyIEC_Timer.Q = 1 und der Wert von Tag_Output = 1. Nach Ablauf des Wertes Tag_Time ist DB1.MyIEC_Timer.Q = 0 und der Wert von Tag_Output = 0.

6.3.6 Zähler

Mit den Zähleranweisungen können Sie programminterne Ereignisse und externe Prozessereignisse zählen.

- Der Zähler "Vorwärts zählen" (CTU) zählt um 1 vorwärts, wenn der Wert des Eingangsparameters CU von 0 nach 1 wechselt.
- Der Zähler "Rückwärts zählen" (CTD) zählt um 1 rückwärts, wenn der Wert des Eingangsparameters CD von 0 nach 1 wechselt.
- Der Zähler "Vorwärts und rückwärts zählen" (CTUD) zählt um 1 vorwärts oder rückwärts, wenn der Vorwärtzähleingang (CU) oder der Rückwärtzähleingang (CD) von 0 nach 1 wechselt.

Die S7-1200 bietet außerdem schnelle Zähler (Seite 133) (HSC), um Ereignisse zu zählen, die schneller als die OB-Ausführungsrate auftreten.

Die Anweisungen CU, CD und CTUD nutzen Softwarezähler, deren maximale Zählgeschwindigkeit durch die Ausführungsrate des OBs, in den sie eingefügt wurden, begrenzt ist.

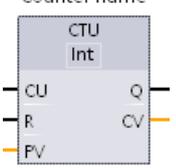
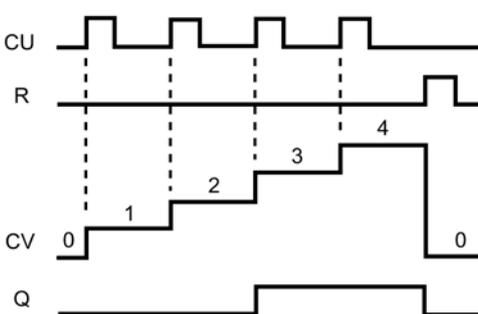
Hinweis

Wenn die zu zählenden Ereignisse innerhalb der Ausführungsrate des OBs auftreten, können Sie die Zähleranweisungen CTU, CTD oder CTUD verwenden. Wenn die Ereignisse schneller als die Ausführungsrate des OBs auftreten, verwenden Sie den HSC.

Jeder Zähler nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten des Zählers zu speichern. In SCL müssen Sie für die einzelne Zähleranweisung zunächst einen DB erstellen, damit Sie ihn referenzieren können. In KOP und FUP erstellt STEP 7 automatisch den DB, wenn Sie die Anweisung einfügen.

Die Anzahl der Zähler, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Einzelne Zähler verwenden 3 Byte (für SInt oder USInt), 6 Byte (für Int oder UInt) bzw. 12 Byte (für DInt oder UDInt).

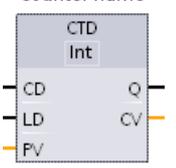
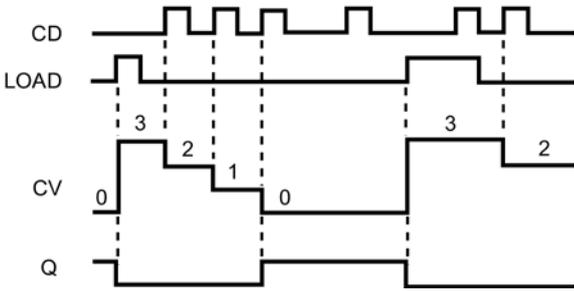
Tabelle 6- 17 CTU Vorwärtszähler

KOP/FUP	SCL	Bedienung
<p>"Counter name"</p> 	<pre>"ctu_db".CTU(CU:=_bool_in, R:=_bool_in, PV:=_undef_in, Q=>_bool_out, CV=>_undef_out);</pre>	

Das Zeitdiagramm zeigt die Funktionsweise eines CTU-Zählers mit ganzzahligem Zählwert ohne Vorzeichen (und PV = 3).

- 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, dann wird CV auf 0 zurückgesetzt.

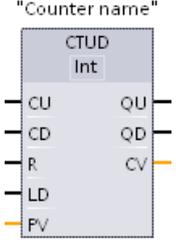
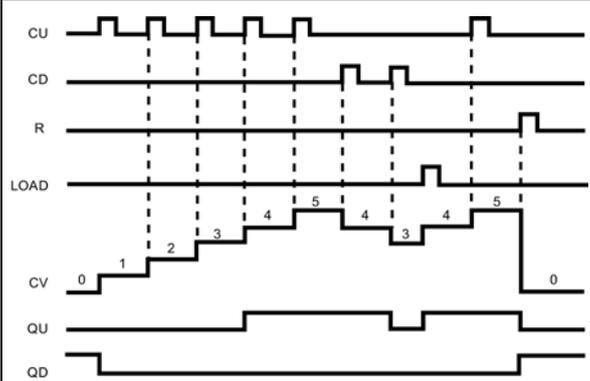
Tabelle 6- 18 CTD Rückwärtszähler

KOP/FUP	SCL	Funktionsweise
<p>"Counter name"</p> 	<pre>"ctd_db".CTU(CD:=_bool_in, LOAD:=_bool_in, PV:=_undef_in, Q=>_bool_out, CV=>_undef_out);</pre>	

Das Zeitdiagramm zeigt die Funktionsweise eines CTD-Zählers mit ganzzahligem Zählwert ohne Vorzeichen (und PV = 3).

- 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 in den Zähler geladen.

Tabelle 6- 19 CTUD Vorwärts- und Rückwärtszähler

KOP/FUP	SCL	Funktionsweise
<p>"Counter name"</p> 	<pre>"ctud_db".CTUD (CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LOAD:=_bool_in, PV:=_undef_in, QU=>_bool_out, QD=>_bool_out, CV=>_undef_out);</pre>	

Das Zeitdiagramm zeigt die Funktionsweise eines CTUD-Zählers mit ganzzahligem Zählwert ohne Vorzeichen (und PV = 4).

- Ist der Wert des Parameters CV (aktueller Zählwert) größer oder gleich dem Wert des Parameters PV (voreingestellter Wert), 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 CV auf 0 zurückgesetzt.

6.3.7 Impulsdauermodulation (PWM)

Tabelle 6- 20 Anweisung CTRL_PWM

KOP/FUP	SCL	Beschreibung
	<pre>"ctrl_pwm_db" (PWM:=_hw_pwm_in_, enable:=_bool_in_, busy=>_bool_out_, status=>_word_out_);</pre>	<p>Die Anweisung CTRL_PWM bietet Ihnen 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>

Die Anweisung CTRL_PWM speichert die Parameterinformationen im DB. In SCL müssen Sie für die Anweisung zunächst einen DB erstellen, damit Sie ihn referenzieren können. In KOP und FUP erstellt STEP 7 automatisch den DB, wenn Sie die Anweisung einfügen. Die Parameter des Datenbausteins werden von der Anweisung CTRL_PWM bestimmt.

Die Impulsdauer wird beim ersten Wechsel der CPU in RUN auf den in der Gerätekonfiguration eingegebenen Anfangswert gesetzt. Um die Impulsdauer zu ändern, schreiben Sie die gewünschten Werte in das in der Gerätekonfiguration ("Ausgangsadressen/Anfangsadresse") angegebene Datenwort für den Ausgang (Q). Um die gewünschte Impulsdauer in das entsprechende Datenwort für den Ausgang (Q) zu schreiben, nutzen Sie eine Anweisung (wie Move, Convert, eine mathematische Anweisung oder PID). Sie müssen dabei den gültigen Bereich des Ausgangswertes beachten (d. h. Prozent, Tausendstel, Zehntausendstel oder S7-Analogformat).



- ① Zykluszeit
- ② Impulsdauer

Die relative Einschaltdauer kann z. B. als Prozentsatz der Zykluszeit oder als relative Menge (z. B. 0 bis 1000 oder 0 bis 10000) ausgedrückt werden. Die Impulsdauer kann zwischen 0 (kein Impuls, immer aus) und Vollausschlag (kein Impuls, immer ein) liegen.

Der PWM-Ausgang kann zwischen 0 und Vollausschlag liegen und bietet 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.

6.4 Einfache Erstellung von Datenprotokollen

Ihr Steuerungsprogramm kann mit den Anweisungen Data log Laufzeitdatenwerte in beständigen Protokolldateien speichern. Die Datenprotokolldateien werden im Flash-Speicher (CPU oder Memory Card) gespeichert. Die Daten der Protokolldateien werden im herkömmlichen CSV-Format (durch Komma getrennte Werte) gespeichert. Die Datensätze sind in einer zirkulären Protokolldatei vordefinierter Größe organisiert.

Die Anweisungen Data log dienen in Ihrem Programm dazu, einen Datensatz anzulegen, zu öffnen, zu schreiben und die Protokolldateien zu schließen. Sie entscheiden, welche Programmwerte protokolliert werden, indem Sie einen Datenpuffer anlegen, der einen einzigen Protokolldatensatz definiert. Ihr Datenpuffer wird als temporärer Speicher für einen neuen Protokolldatensatz verwendet. Neue aktuelle Werte müssen während der Laufzeit programmatisch in den Puffer übertragen werden. Wenn alle aktuellen Datenwerte aktualisiert sind, können Sie die Anweisung DataLogWrite ausführen, um Daten aus dem Puffer in einen Protokolldatensatz zu übertragen.

Datenprotokolldateien können Sie auf der Dateibrowser-Seite des Webserver 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.

Mit den Anweisungen DataLog speichern Sie Laufzeit-Prozessdaten programmatisch im Flash-Speicher der CPU. Die Datensätze sind in einer kreisförmigen Protokolldatei vordefinierter Größe organisiert. Neue Datensätze werden an die Datenprotokolldatei angehängt. Wenn die maximale Anzahl Datensätze in der Datenprotokolldatei gespeichert ist, überschreibt der nächste geschriebene Datensatz den ältesten Datensatz. Um das Überschreiben von Datensätzen zu verhindern, können Sie die Anweisung DataLogNewFile verwenden. Neue Datensätze werden dann in einer neuen Datenprotokolldatei gespeichert, und die alte Datenprotokolldatei verbleibt in der CPU.

Tabelle 6- 21 Anweisung DataLogWrite

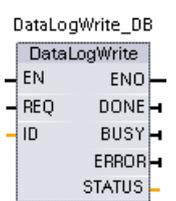
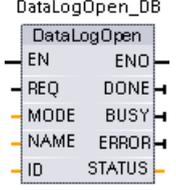
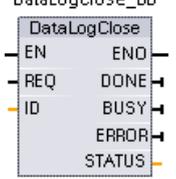
KOP/FUP	SCL	Beschreibung
	<pre>"DataLogWrite_DB" (req:=_bool_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, ID:=_dword_inout_);</pre>	<p>DataLogWrite schreibt einen Datensatz in das angegebene Datenprotokoll. Das bereits bestehende Zieldatenprotokoll muss geöffnet sein.</p> <p>Sie müssen den Datensatzpuffer programmatisch mit aktuellen Laufzeitdatenwerten laden und dann die Anweisung DataLogWrite ausführen, um neue Datensatzdaten aus dem Puffer in ein Datenprotokoll zu übertragen.</p> <p>Tritt während einer noch nicht beendeten Anweisung DataLogWrite ein Spannungsausfall auf, geht der ins Datenprotokoll übertragene Datensatz möglicherweise verloren.</p>

Tabelle 6- 22 Anweisungen DataLogCreate und DataLogNewFile

KOP/FUP	SCL	Beschreibung
<p style="text-align: center;">DataLogCreate_ DB</p>	<pre>"DataLogCreate_DB" (req:=_bool_in_, records:=_udint_in_, format:=_uint_in_, timestamp:=_uint_in_, done=>_bool_out_, busy=>_bool_out_, error=>_bool_out_, status=>_word_out_, name:=_string_inout_, ID:=_dword_inout_, header:=_variant_inout_, data:=_variant_inout_);</pre>	<p>DataLogCreate¹ erstellt und initialisiert eine Datenprotokolldatei im Verzeichnis \DataLogs der CPU. Die Datenprotokolldatei wird in einer vordefinierten festen Größe angelegt.</p>
<p style="text-align: center;">DataLogNewFile_ DB</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>Mit DataLogNewFile¹ kann Ihr Programm eine neue Datenprotokolldatei basierend auf einer bestehenden Datenprotokolldatei anlegen. Ein neues Datenprotokoll wird angelegt und implizit geöffnet, es trägt den angegebenen NAMEN. Der Header-Datensatz aus dem ursprünglichen Datenprotokoll wird zusammen mit den Eigenschaften des ursprünglichen Datenprotokolls dupliziert. Die ursprüngliche Datenprotokolldatei wird implizit geschlossen.</p>

¹ Die Operationen DataLogCreate und DataLogNewFile 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 überwachen, damit das neue Datenprotokoll für andere Datenprotokolloperationen verwendet werden kann.

Tabelle 6- 23 Anweisungen DataLogOpen und DataLogClose

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>Die Anweisung DataLogOpen öffnet eine bereits vorhandene Datenprotokolldatei. Ein Datenprotokoll muss geöffnet sein, damit Sie neue Datensätze in das Protokoll schreiben können. Datenprotokolle lassen sich einzeln öffnen und schließen. Acht Datenprotokolle können gleichzeitig geöffnet sein.</p>
	<pre>"DataLogClose_DB" (req:= _bool_in_, done=> _bool_out_, busy=> _bool_out_, error=> _bool_out_, status=> _word_out_, ID:= _dword_inout_);</pre>	<p>Die Anweisung DataLogClose 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>

6.5 Einfaches Überwachen und Testen Ihres Anwenderprogramms

6.5.1 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 290) 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 Peripherieeingang forcen. Um einen Peripherieeingang 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.

6.5.2 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 6- 24 Querverweiselemente

Spalte	Beschreibung
Objekt	Name des Objekts, das die angegebenen unterlagerten Objekte verwendet oder das von den unterlagerten Objekten verwendet wird
Anzahl	Anzahl Verwendungen
Adresse	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

6.5.3 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 Programmierer 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.

6.5.4 Diagnoseanweisungen zur Überwachung der Hardware

6.5.4.1 Zustände der LEDs der CPU lesen

Mit der Anweisung LED kann Ihr Anwenderprogramm den Zustand der LEDs auf der CPU ermitteln. Mit Hilfe dieser Informationen können Sie eine Variable für Ihr HMI-Gerät programmieren.

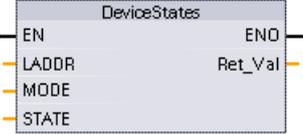
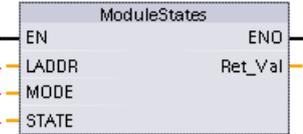
Tabelle 6- 25 Anweisung LED

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := #LED(laddr:=_hw_io_in_, LED:=_uint_in_);</pre>	<p>RET_VAL gibt die folgenden LED-Zustände für die CPU zurück</p> <ul style="list-style-type: none"> • RUN/STOP: grün oder gelb • Fehler: rot • MAINT (Wartung): gelb • Link: grün • Tx/Rx (Senden/Empfangen): gelb

6.5.4.2 Anweisungen zum Lesen des Diagnosezustands der Geräte

STEP 7 verfügt auch über Anweisungen zum Lesen der Statusinformationen, die von den Hardwaregeräten in Ihrem Netzwerk geliefert werden.

Tabelle 6- 26 Diagnoseanweisungen

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := #GET_DIAG(mode:=_uint_in_, laddr:=_hw_any_in_, channel:=_uint_in_, cnt_diag=>_uint_out_, diag:=_uint_inout_, detail:=_variant_inout_);</pre>	<p>Mit der Anweisung GET_DIAG lesen Sie die Diagnoseinformationen aus einem angegebenen Hardwaregerät.</p>
	<pre>ret_val := DeviceStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>Mit der Anweisung DeviceStates lesen Sie den Zustand von PROFINET- oder PROFIBUS-Geräten.</p>
	<pre>ret_val := ModuleStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	<p>Mit der Anweisung ModuleStates lesen Sie den Zustand von PROFINET- oder PROFIBUS-Modulen.</p>

6.6 Schneller Zähler (HSC)

Die schnellen Zähler (HSC) zählen Ereignisse, die schneller als die Ausführungsrate des OBs auftreten. Die Anweisung CTRL_HSC steuert die Funktionsweise des HSC.

Hinweis

Wenn die zu zählenden Ereignisse innerhalb der Ausführungsrate des OBs auftreten, können Sie die Zähleranweisungen CTU, CTD oder CTUD verwenden. Wenn die Ereignisse schneller als die Ausführungsrate des OBs auftreten, verwenden Sie den HSC.

Sie konfigurieren die Parameter für jeden HSC in der Gerätekonfiguration für die CPU: Zählermodus, E/A-Anschaltungen, Alarmzuweisung und Betrieb als schneller Zähler oder als Gerät für die Impulsfrequenzmessung.

Tabelle 6- 27 Anweisung CTRL_HSC

KOP/FUP	SCL	Beschreibung
	<pre>"counter_name" (hsc:=_hw_hsc_in_, HSC:= , DIR:=_bool_in_, CV:=_bool_in_, RV:=_bool_in_, Period:=_bool_in_, New_DIR:=_int_in_, New_CV:=_int_in_, New_RV:=_dint_in_, New_Period:=_int_in_, Busy:=_bool_out_, Status:=_word_out_);</pre>	<p>Jede Anweisung CTRL_HSC nutzt eine in einem Datenbaustein abgelegte Struktur, um Zählerdaten zu speichern.</p> <p>In SCL müssen Sie für die einzelne Zähleranweisung zunächst einen DB erstellen, damit Sie ihn referenzieren können. In KOP und FUP erstellt STEP 7 automatisch den DB, wenn Sie die Anweisung einfügen.</p>

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 Zähleralarm 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.

Hinweis

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.

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. Werden die Werte der folgenden Booleschen Merker auf 0 gesetzt, führt dies zu keiner Änderung.

- Wird DIR = 1 gesetzt, wird ein Wert NEW_DIR geladen.
- Wird CV = 1 gesetzt, wird ein Wert NEW_CV geladen.
- Wird RV = 1 gesetzt, wird ein Wert NEW_RV geladen.
- Wird PERIOD = 1 gesetzt, wird ein Wert NEW_PERIOD geladen.

6.6.1 Funktionsweise von schnellen Zählern

Schnelle Zähler (HSC) zählen Ereignisse, die schneller als die Ausführungsrate des Zyklus-OBs auftreten. Wenn die zu zählenden Ereignisse langsamer als die Ausführungsrate des OBs auftreten, können Sie die Standard-Zähleranweisungen CTU, CTD oder CTUD verwenden. Wenn die Ereignisse schneller als die Ausführungsrate des OBs auftreten, verwenden Sie den schnelleren HSC. Über die Anweisung CTRL_HSC kann Ihr Programm programmatisch einige der HSC-Parameter ändern.

Beispiel: Sie können den HSC als Eingang für einen Winkelschrittgeber nutzen. Der Winkelschrittgeber sorgt für eine bestimmte Anzahl von Zählwerten pro Umdrehung sowie für einen Rücksetzimpuls einmal pro Umdrehung. Der bzw. die Taktgeber und der Rücksetzimpuls des Winkelschrittgebers liefern die Eingänge für den schnellen Zähler.

Der erste von mehreren voreingestellten Werten wird in den schnellen Zähler geladen. Die Ausgänge werden für die Zeitspanne aktiviert, während der der aktuelle Wert des Zählers kleiner als der voreingestellte Wert ist. Der HSC löst einen Alarm aus, wenn der aktuelle Wert des Zählers gleich dem voreingestellten Wert ist, wenn der Zähler zurückgesetzt wird oder wenn ein Richtungswechsel auftritt.

Wenn der aktuelle Wert gleich dem voreingestellten Wert ist und es zu einem Alarmereignis kommt, dann wird ein neuer voreingestellter Wert geladen und der nächste Signalzustand für die Ausgänge gesetzt. Tritt ein Alarmereignis auf, weil der Zähler zurückgesetzt wird, dann werden der erste voreingestellte Wert und die ersten Signalzustände der Ausgänge gesetzt und der Zyklus wiederholt.

Da die Alarmergebnisse in einer sehr viel geringeren Geschwindigkeit auftreten als der schnelle Zähler zählt, kann eine präzise Steuerung der schnellen Anweisungen mit relativ geringem Einfluss auf den Zyklus der CPU implementiert werden. Da Sie Alarmergebnisse bestimmten Interruptprogrammen zuordnen können, kann jede neue Voreinstellung in einem getrennten Interruptprogramm geladen werden, damit so der Zustand einfach gesteuert werden kann. Sie können alternativ auch alle Alarmereignisse in einem einzigen Interruptprogramm bearbeiten.

Auswahl des HSC-Eingangskanals

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.

Hinweis

Eingangskanäle von CPU und SB (ab Firmware V4) haben konfigurierbare Eingangfilterzeiten

Bei früheren Firmware-Versionen 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 des Eingangsfilters von 6,4 ms ist für Ihre Prozesssignale möglicherweise zu langsam. Die Filterzeiten für die HSC-Eingänge müssen für Ihre HSC-Anwendung optimiert werden. Weitere Informationen erhalten Sie unter "Konfigurieren der Filterzeiten von Digitaleingängen".

Tabelle 6- 28 CPU-Eingang: maximale Frequenz

CPU	CPU-Eingangskanal	1- oder 2-Phasenbetrieb	A/B-Zähler
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
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 6- 29 SB-Eingang: maximale Frequenz (optionales Board)

Signalboard (SB)	SB-Eingangskanal	1- oder 2-Phasenbetrieb	A/B-Zähler
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

Funktionalität für den HSC auswählen

Alle HSCs arbeiten in der gleichen Zählerart auf die gleiche Weise. Zählerart, Richtungssteuerung und anfängliche Richtung für die HSC-Funktionseigenschaften werden in der CPU-Gerätekonfiguration zugewiesen.

Es gibt vier grundlegende Arten von schnellen Zählern:

- Einphasenzähler mit interner Richtungssteuerung
- Einphasenzähler mit externer Richtungssteuerung
- Zweiphasenzähler mit 2 Takteingängen
- A/B-Zähler

Sie können jeden HSC-Typ mit oder ohne Rücksetzeingang verwenden. Wenn Sie den Rücksetzeingang aktivieren (mit einigen Einschränkungen, siehe folgende Tabelle), wird der aktuelle Wert zurückgesetzt. Er bleibt so lange zurückgesetzt, bis Sie den Rücksetzeingang deaktivieren.

- Frequenzfunktion: Der HSC kann in bestimmten Betriebsarten so konfiguriert werden (Zählart), dass er eine Frequenz statt der aktuellen Impulszahl ausgibt. Es gibt drei verschiedene Frequenzmesszeiten: 0,01, 0,1 oder 1,0 Sekunden.

Die Frequenzmesszeit legt fest, wie oft der HSC einen neuen Frequenzwert berechnet und ausgibt. Die ausgegebene Frequenz ist ein Mittelwert, der anhand der Gesamtzahl der Zählwerte im letzten Messzeitraum berechnet wird. Ändert sich die Frequenz schnell, ist der ausgegebene Wert ein Zwischenergebnis zwischen der höchsten und der niedrigsten während des Messzeitraums aufgetretenen Frequenz. Die Frequenz wird unabhängig von der eingestellten Frequenzmesszeit immer in Hertz (Impulse pro Sekunde) angegeben.

- Zählerarten und Zählereingänge: Die folgende Tabelle zeigt die Eingänge, die für Funktionen wie Taktgeber, Richtungssteuerung und Zurücksetzen des HSC verwendet werden.

Ein Eingang kann nicht für zwei verschiedene Funktionen verwendet werden. Wird ein Eingang jedoch nicht von der aktuellen Zählerart des definierten schnellen Zählers benötigt, kann er für andere Zwecke genutzt werden. Befindet sich beispielsweise HSC1 in einer Betriebsart, in der die zwei integrierten Eingänge, aber nicht der dritte externe Rücksetzeingang (Standardzuweisung an E0.3) verwendet wird, so kann E0.3 für Flankenalarme oder für HSC2 belegt werden.

Tabelle 6- 30 Zählarten für den HSC

Typ	Eingang 1	Eingang 2	Eingang 3	Funktion
Einphasenzähler mit interner Richtungssteuerung	Takt	-	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
Einphasenzähler mit externer Richtungssteuerung	Takt	Richtung	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
Zweiphasenzähler mit 2 Takteingängen	Takt vorwärts	Takt rückwärts	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
A/B-Zähler	Phase A	Phase B	-	Zählwert oder Frequenz
			Zurücksetze n ¹	Zählwert

¹ Bei einem Winkelschrittgeber: Phase Z, Referenzpunkt

Eingangsadressen für den HSC

Beim Konfigurieren der CPU können Sie die "Hardware-Eingänge" für jeden HSC aktivieren und konfigurieren.

Alle HSC-Eingänge müssen an Klemmen an der CPU oder am optionalen Signalboard (das in die Front der CPU gesteckt wird) angeschlossen sein.

Hinweis

Wie Sie in den folgenden Tabellen sehen, überschneiden sich die Standardzuweisungen der optionalen Signale der verschiedenen HSCs. Zum Beispiel wird für das optionale externe Rücksetzen von HSC 1 der gleiche Eingang verwendet wie für einen der Eingänge von HSC 2.

Bei CPUs ab V4 können Sie die HSC-Eingänge während der CPU-Konfiguration neu zuweisen. Sie brauchen nicht die Standard-Eingangszuweisungen zu verwenden.

Stellen Sie stets sicher, dass Sie Ihre HSCs so konfigurieren, dass **kein** Eingang von zwei HSCs verwendet wird.

Die folgenden Tabellen zeigen die standardmäßigen HSC-Eingangszuweisungen für die integrierten E/A der CPUs und eines optionalen SBs. (Wenn die ausgewählte SB-Ausführung nur zwei Eingänge hat, sind nur die Eingänge 4.0 und 4.1 verfügbar.)

Definitionen für die Tabelle der HSC-Eingänge

- **Einphasenzähler:** **C** ist der Zähleingang, **[d]** ist der Richtungseingang (optional) und **[R]** ist der Eingang für **externes Rücksetzen** (optional) (Rücksetzen ist nur im Zählmodus verfügbar).
- **Zweiphasenzähler:** **CU** ist der Vorwärtszähleingang, **CD** ist der Rückwärtszähleingang und **[R]** ist der Eingang für **externes Rücksetzen** (optional) (Rücksetzen ist nur im Zählmodus verfügbar).
- **A/B-Zähler:** **A** ist der Zähleingang **A**, **B** ist der Zähleingang **B** und **[R]** ist der Eingang für **externes Rücksetzen** (optional). (Rücksetzen ist nur im Zählmodus verfügbar.)

Tabelle 6- 31 CPU 1211C: Zuweisungen der HSC-Standardadressen

HSC-Zählermodus		Integrierter CPU-Eingang (Standard 0.x)						Eingang eines optionalen SBs (Standard 4.x) ¹			
		0	1	2	3	4	5	0	1	2	3
HSC 1	Einphasig	C	[d]		[R]			C	[d]		[R]
	Zweiphasig	CU	CD		[R]			CU	CD		[R]
	A/B-Zähler	A	B		[R]			A	B		[R]
HSC 2	Einphasig		[R]	C	[d]				[R]	C	[d]
	Zweiphasig		[R]	CU	CD				[R]	CU	CD
	A/B-Zähler		[R]	A	B				[R]	A	B
HSC 3	Einphasig					C	[d]	C	[d]		[R]
	Zweiphasig										
	A/B-Zähler										
HSC 4	Einphasig					C	[d]	C	[d]		[R]
	Zweiphasig					CU	CD				
	A/B-Zähler					A	B				
HSC 5	Einphasig							C	[d]		[R]
	Zweiphasig							CU	CD		[R]
	A/B-Zähler							A	B		[R]
HSC 6	Einphasig								[R]	C	[d]
	Zweiphasig								[R]	CU	CD
	A/B-Zähler								[R]	A	B

¹ Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Tabelle 6- 32 CPU 1212C: Zuweisungen der HSC-Standardadressen

HSC-Zählermodus		Integrierter CPU-Eingang (Standard 0.x)								Eingang eines optionalen SBs (Standard 4.x) ¹			
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 1	Einphasig	C	[d]		[R]					C	[d]		[R]
	Zweiphasig	CU	CD		[R]					CU	CD		[R]
	A/B-Zähler	A	B		[R]					A	B		[R]
HSC 2	Einphasig		[R]	C	[d]						[R]	C	[d]
	Zweiphasig		[R]	CU	CD						[R]	CU	CD
	A/B-Zähler		[R]	A	B						[R]	A	B
HSC 3	Einphasig					C	[d]		[R]	C	[d]		[R]
	Zweiphasig					CU	CD		[R]				
	A/B-Zähler					A	B		[R]				
HSC 4	Einphasig						[R]	C	[d]	C	[d]		[R]
	Zweiphasig						[R]	CU	CD				
	A/B-Zähler						[R]	A	B				

HSC-Zählermodus		Integrierter CPU-Eingang (Standard 0.x)							Eingang eines optionalen SBs (Standard 4.x) ¹				
		0	1	2	3	4	5	6	7	0	1	2	3
HSC 5	Einphasig									C	[d]		[R]
	Zweiphasig									CU	CD		[R]
	A/B-Zähler									A	B		[R]
HSC 6	Einphasig										[R]	C	[d]
	Zweiphasig										[R]	CU	CD
	A/B-Zähler										[R]	A	B

¹ Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Tabelle 6- 33 CPU 1214C, CPU 1215C und CPU1217C:

Zuweisungen der HSC-Standardadressen

(nur integrierte Eingänge, Adressen des optionalen SBs siehe nächste Tabelle)

HSC-Zählermodus		Digitaleingangsbyte 0 (Standard: 0.x)							Digitaleingangsbyte 1 (Standard: 1.x)						
		0	1	2	3	4	5	6	7	0	1	2	3	4	5
HSC 1	Einphasig	C	[d]		[R]										
	Zweiphasig	CU	CD		[R]										
	A/B-Zähler	A	B		[R]										
HSC 2	Einphasig		[R]	C	[d]										
	Zweiphasig		[R]	CU	CD										
	A/B-Zähler		[R]	A	B										
HSC 3	Einphasig					C	[d]		[R]						
	Zweiphasig					CU	CD		[R]						
	A/B-Zähler					A	B		[R]						
HSC 4	Einphasig						[R]	C	[d]						
	Zweiphasig						[R]	CU	CD						
	A/B-Zähler						[R]	A	B						
HSC 5	Einphasig									C	[d]	[R]			
	Zweiphasig									CU	CD	[R]			
	A/B-Zähler									A	B	[R]			
HSC 6	Einphasig												C	[d]	[R]
	Zweiphasig												CU	CD	[R]
	A/B-Zähler												A	B	[R]

Tabelle 6- 34 Optionales SB in CPUs der obigen Tabelle: Zuweisungen der HSC-Standardadressen

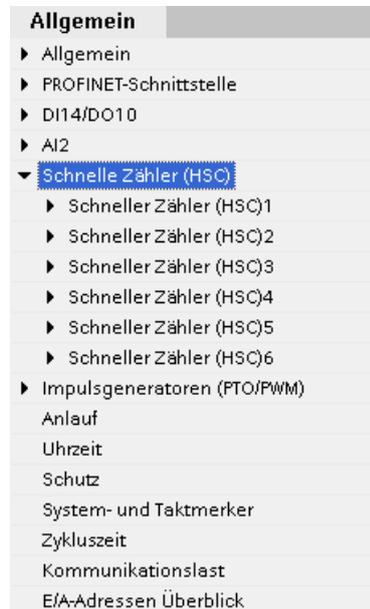
HSC		Eingänge des optionalen SBs (Standard: 4.x) ¹			
		0	1	2	3
HSC 1	Einphasig	C	[d]		[R]
	Zweiphasig	CU	CD		[R]
	A/B-Zähler	A	B		[R]
HSC 2	Einphasig		[R]	C	[d]
	Zweiphasig		[R]	CU	CD
	A/B-Zähler		[R]	A	B
HSC 5	Einphasig	C	[d]		[R]
	Zweiphasig	CU	CD		[R]
	A/B-Zähler	A	B		[R]
HSC 6	Einphasig		[R]	C	[d]
	Zweiphasig		[R]	CU	CD
	A/B-Zähler		[R]	A	B

¹ Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Hinweis

Die von schnellen Zählern verwendeten digitalen E/A werden während der CPU-Gerätekonfiguration zugewiesen. Wenn HSC-Geräten Adressen digitaler E/A zugewiesen werden, können die Adresswerte der zugewiesenen E/A nicht durch die Forcefunktion einer Beobachtungstabelle geändert werden.

6.6.2 Konfiguration eines schnellen Zählers



Es können bis zu 6 schnelle Zähler konfiguriert werden. Öffnen Sie die CPU-Gerätekonfiguration und weisen Sie die Eigenschaften der einzelnen HSC zu. Aktivieren Sie einen HSC durch Auswahl der Option "Aktivieren" für den HSC.

Mit der Anweisung CTRL_HSC in Ihrem Anwenderprogramm können Sie die Funktionsweise des HSC steuern.



WARNUNG

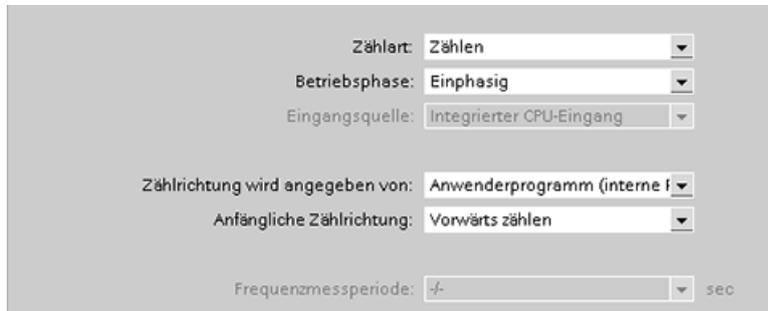
Gefahren beim Ändern der Filterzeiteinstellung für digitale Eingangskanäle

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 tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

Damit die neue Filterzeit sofort wirksam wird, ist die CPU aus- und wieder einzuschalten.

Nach der Aktivierung des HSC konfigurieren Sie die anderen Parameter wie Zählerfunktion, Anfangswerte, Rücksetzoptionen und Interruptereignisse.

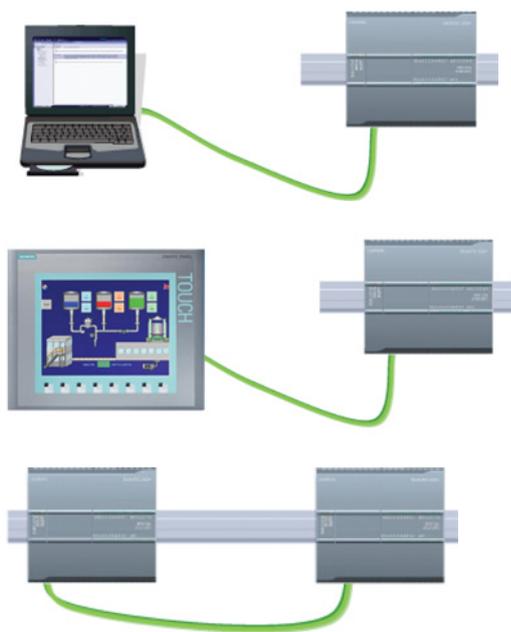


The image shows a configuration window for an HSC (High-Speed Counter) with the following settings:

- Zählart: Zählen
- Betriebsphase: Einphasig
- Eingangsquelle: Integrierter CPU-Eingang
- Zählrichtung wird angegeben von: Anwenderprogramm (interne I
- Anfängliche Zählrichtung: Vorwärts zählen
- Frequenzmessperiode: +/- sec

Weitere Informationen zum Konfigurieren des HSC finden Sie im Abschnitt zum Konfigurieren der CPU (Seite 84).

Einfache Kommunikation zwischen Geräten



Bei direkter Verbindung zwischen dem Programmiergerät und einer CPU:

- Das Projekt muss die CPU enthalten.
- Das Programmiergerät ist nicht Teil des Projekts, auf dem Gerät muss jedoch STEP 7 ausgeführt werden.

Um eine Direktverbindung zwischen einem HMI-Bediengerät und einer CPU herzustellen, muss das Projekt sowohl die CPU als auch die HMI enthalten.

Bei direkter Verbindung zwischen zwei CPUs:

- Das Projekt muss beide CPUs enthalten.
- Sie müssen zwischen den beiden CPUs eine Netzwerkverbindung konfigurieren.

Die S7-1200 CPU ist eine PROFINET IO-Steuerung und kommuniziert mit STEP 7 auf einem Programmiergerät, mit HMI-Geräten und mit anderen CPUs oder Geräten nicht von Siemens. Für die direkte Kommunikation zwischen einem Programmiergerät oder einem HMI-Gerät und einer CPU ist kein Ethernet-Switch erforderlich. Erst wenn mehr als zwei CPUs oder HMI-Geräte in einem Netzwerk vorhanden sind, wird ein Ethernet-Switch benötigt.

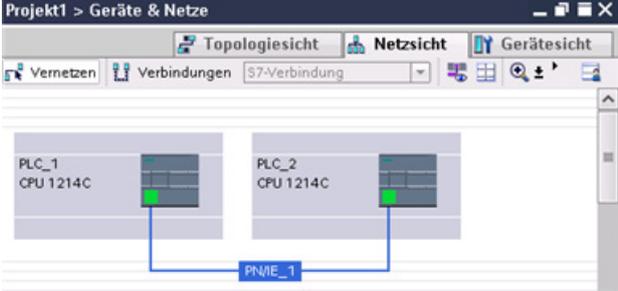
Durch Hinzufügen eines PROFIBUS CMs kann Ihre CPU auch als Master oder Slave in einem PROFIBUS-Netzwerk fungieren.

Andere Kommunikationsschnittstellen (CM, CP oder CB) unterstützen eine Vielzahl Protokolle, z. B. Punkt-zu-Punkt (PTP), Modbus, USS und GPRS (Modem).

7.1 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 7- 1 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>	

7.2 Kommunikationsoptionen

Die S7-1200 bietet mehrere Arten der Kommunikation zwischen CPUs und Programmiergeräten, HMI-Geräten und anderen CPUs.

WARNUNG

Wenn sich ein Angreifer physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.

Das TIA Portal, die CPU und HMI-Geräte (außer HMI-Geräten über GET/PUT) nutzen die sichere Kommunikation, die Schutz vor Replay- und Man-in-the-Middle-Angriffen bietet. Nach der Aktivierung der Kommunikation erfolgt der Austausch signierter Meldungen in Klartext, wodurch ein Angreifer Daten lesen kann, die Daten jedoch vor unberechtigtem Schreiben geschützt sind. Das TIA Portal, nicht der Kommunikationsprozess, verschlüsselt die Daten von Knowhow-geschützten Bausteinen.

Alle anderen 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.

Informationen und Empfehlungen bezüglich der Sicherheit finden Sie in unseren Operational Guidelines für Industrial Security auf der Website "Service & Support" von Siemens.

PROFINET

PROFINET wird für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern über Ethernet verwendet:

- PROFINET unterstützt 16 IO-Geräte mit maximal 256 Submodulen. PROFIBUS erlaubt 3 unabhängige PROFIBUS DP-Master und unterstützt 32 IO-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 RT IO-Steuerung

Als IO-Steuerung über PROFINET RT 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 erfolgt 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.

Siehe auch

Siemens-Website Service & Support
(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

7.3 Anzahl der asynchronen Kommunikationsverbindungen

Die CPU unterstützt die folgende maximale Anzahl von gleichzeitigen, asynchronen Kommunikationsverbindungen für PROFINET und PROFIBUS:

- 8 Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV.
- 3 CPU-zu-CPU-S7-Verbindungen für Server-GET/PUT-Daten
- 8 CPU-zu-CPU-S7-Verbindungen für Client-GET/PUT-Daten

Hinweis

S7-1200, S7-300 und S7-400 CPUs nutzen die Anweisungen GET und PUT für die S7-Kommunikation von CPU zu CPU. Eine S7-200 CPU verwendet ETHx_XFER-Anweisungen für die S7-Kommunikation von CPU zu CPU.

-
- HMI-Verbindungen: Die Anzahl der für Ihre Anwendung erforderlichen HMI-Verbindungen können Sie über das TIA Portal konfigurieren.
 - PG-Verbindungen: Die CPU bietet Verbindungen, um 1 Programmiergerät (PG) zu unterstützen.
 - Webserver (HTTP)-Verbindungen: Die CPU bietet Verbindungen für den Webserver.

Hinweis

Die Kommunikationsverbindungen von offener Benutzerkommunikation, S7-Verbindung, HMI, Programmiergerät und Webserver (HTTP) können basierend auf den verwendeten Funktionen mehrere Verbindungsressourcen nutzen.

7.4 PROFINET- und PROFIBUS-Anweisungen

PROFINET-Anweisungen

Die Anweisungen TSEND_C und TRCV_C vereinfachen die PROFINET-Kommunikation, indem sie die Funktionalität der Anweisungen TCON und TDISCON mit der Anweisung TSEND oder TRCV verbinden.

- 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 sie automatisch von der CPU gehalten und überwacht. TSEND_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TSEND in einer Anweisung.
- 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. Die Anweisung TRCV_C verbindet die Funktionen der Anweisungen TCON, TDISCON und TRCV in einer Anweisung.

Die Anweisungen TCON, TDISCON, TSEND und TRCV werden ebenfalls unterstützt.

Mit den Anweisungen TUSEND und TURCV können Sie Daten über UDP senden oder empfangen. TUSEND und TURCV (sowie TSEND, TRCV, TCON, TDISCON) funktionieren asynchron, das bedeutet, dass sich die Verarbeitung des Auftrags über mehrere Anweisungsaufrufe erstreckt.

Die Anweisung IP_CONF ändert die Parameter der IP-Konfiguration über das Anwenderprogramm. IP_CONF arbeitet asynchron. Die Ausführung erstreckt sich über mehrere Aufrufe.

PROFIBUS-Anweisungen

Mit der Anweisung DPNRM_DG (Diagnose auslesen) können Sie die aktuellen Diagnosedaten eines DP-Slaves in dem Format, das in EN 50 170 Band 2, PROFIBUS angegeben ist, lesen.

Anweisungen für die dezentrale Peripherie für PROFINET, PROFIBUS und AS-i

Die folgenden Anweisungen können Sie für PROFINET, PROFIBUS und GPRS einsetzen.

- Mit den Anweisungen RDREC (Datensatz lesen) und WRREC (Datensatz schreiben) übertragen Sie einen angegebenen Datensatz von oder zu einer Komponente, z. B. zu einem Modul in einem zentralen Baugruppenträger oder zu einer dezentralen Komponente (PROFIBUS DP oder PROFINET IO).
- Mit der Anweisung RALRM (Alarm lesen) lesen Sie einen Alarm und dessen Informationen aus einem DP-Slave oder einer PROFINET IO-Gerätekomponente. Die Informationen in den Ausgangsparametern enthalten die Startinformationen des aufgerufenen OBs sowie die Informationen der Alarmquelle.
- Mit den Anweisungen DPRD_DAT (konsistente Daten lesen) und DPWR_DAT (konsistente Daten schreiben) übertragen Sie konsistente Datenbereiche größer als 64 Bytes von oder zu einem DP-Standardslave.
- Nur bei PROFIBUS lesen Sie mit der Anweisung DPNRM_DG die aktuellen Diagnosedaten eines DP-Slaves in dem Format, das in EN 50 170 Band 2, PROFIBUS, angegeben ist.

7.5 PROFINET

7.5.1 Offene Benutzerkommunikation

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 7-2 Protokolle und Kommunikationsanweisungen

Protokoll	Verwendungsbeispiele	Eintragen der Daten in den Empfangsbereich	Kommunikationsanweisungen	Adressierungsart
TCP	CPU-zu-CPU-Kommunikation	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) Portnummern zu
	Transport von Telegrammen	Datenempfang mit angegebener Länge	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
ISO on TCP	CPU-zu-CPU-Kommunikation	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
	Fragmentierung und Zusammensetzung von Meldungen	Protokollgesteuert	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
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 RT	Kommunikation zwischen CPU und PROFINET IO-Gerät	Datenübertragung und -empfang mit angegebener Länge	Integriert	Integriert

7.5.1.1 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.

7.5.1.2 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" der Anweisung. Unter den Parametern ist auch die Verbindungs-ID der Verbindung.

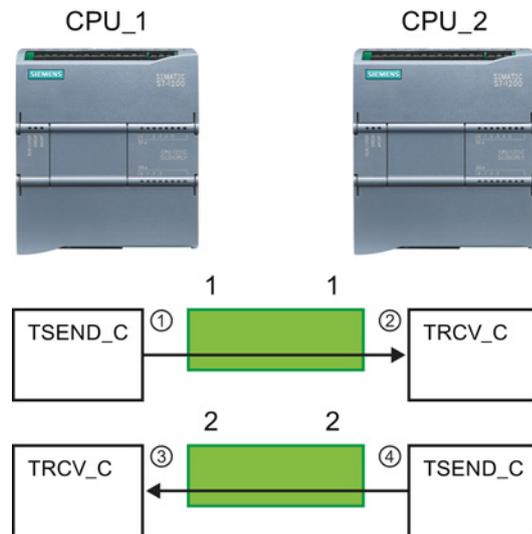
- Die Verbindungs-ID muss für die CPU eindeutig sein. Jede Verbindung, die Sie erstellen, benötigt einen anderen DB und eine andere Verbindungs-ID.
- Die lokale CPU und die Partner-CPU können dieselbe Verbindungs-ID für dieselbe Verbindung verwenden, doch die Verbindungs-IDs müssen nicht übereinstimmen. 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.

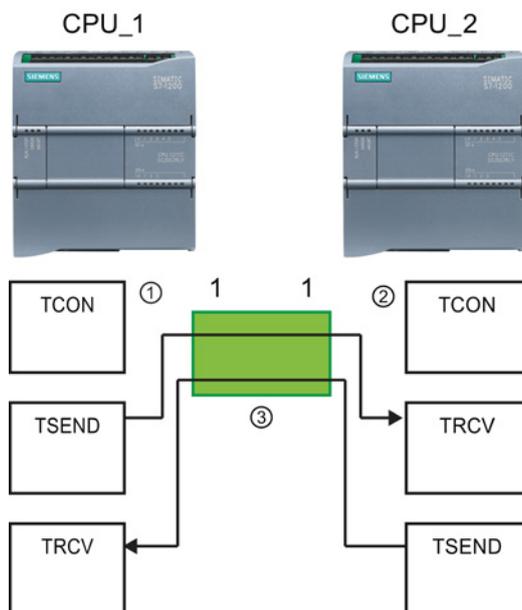
- Die Anweisung TSEND_C in CPU_1 bezieht sich auf die Anweisung TRCV_C in CPU_2 über die erste Verbindung ("Verbindungs-ID 1" bei beiden CPUs, CPU_1 und CPU_2).
- Die Anweisung TRCV_C in CPU_1 bezieht sich auf die Anweisung TSEND_C in CPU_2 über die zweite Verbindung ("Verbindungs-ID 2" bei beiden CPUs, 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).

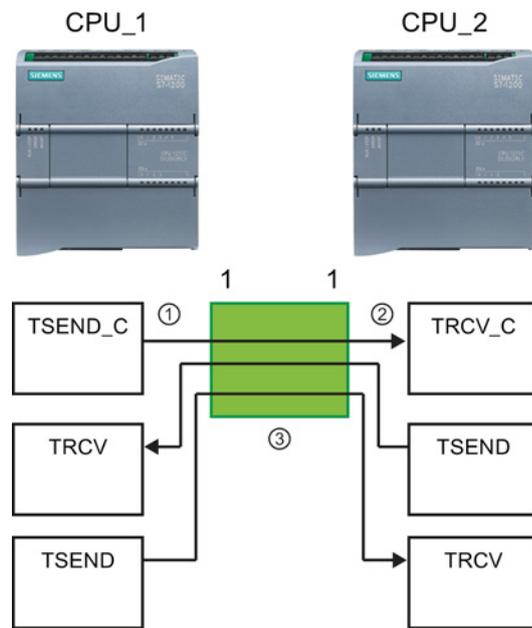
Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die nur eine Verbindung zum Senden und Empfangen von Daten nutzen.

- 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).

7.5.1.3 Parameter für die PROFINET-Verbindung

Bei den Anweisungen TSEND_C, TRCV_C und TCON müssen verbindungsbezogene Parameter angegeben werden, 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" an. Kann auf das Register "Konfiguration" nicht zugegriffen werden, müssen Sie die Struktur TCON_Param programmatisch angeben.

TCON_Param

Tabelle 7-3 Struktur der Verbindungsbeschreibung (TCON_Param)

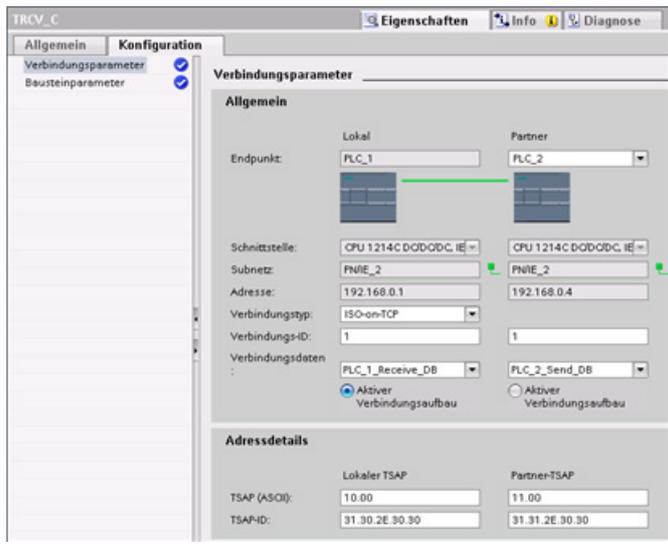
Byte	Parameter und Datentyp		Beschreibung
0 ... 1	block_length	UInt	Länge: 64 Bytes (fest)
2 ... 3	id	CONN_OUC (Word)	Referenz auf diese Verbindung: Wertebereich: 1 (Standard) bis 4095. Geben Sie den Wert dieses Parameters für die Anweisung TSEND_C, TRCV_C oder TCON unter ID an.
4	connection_type	USInt	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	USInt	ID der lokalen PROFINET- oder Industrial Ethernet-Schnittstelle: 1 (Standard)
7	local_tsap_id_len	USInt	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	USInt	Dieser Parameter wird nicht verwendet.
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.

Byte	Parameter und Datentyp		Beschreibung
12 ... 27	local_tsap_id	Array [1..16] of Byte	<p>Komponente der lokalen Adresse der Verbindung:</p> <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 Hexadezimalnotierung; – local_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotierung; – 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. <p>Hinweis: Stellen Sie sicher, dass jeder Wert von local_tsap_id innerhalb der CPU eindeutig ist.</p>
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Dieser Parameter wird nicht verwendet.
34 ... 39	rem_staddr	Array [1..6] of USInt	<p>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:</p> <pre>rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = irrelevant</pre>
40 ... 55	rem_tsap_id	Array [1..16] of Byte	<p>Komponente der Partneradresse der Verbindung:</p> <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 ... 61	next_staddr	Array [1..6] of Byte	Dieser Parameter wird nicht verwendet.
62 ... 63	spare	Word	Reserviert: W#16#0000

7.5.2 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

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 7-4 Verbindungspfad konfigurieren (über die Eigenschaften der Anweisung)

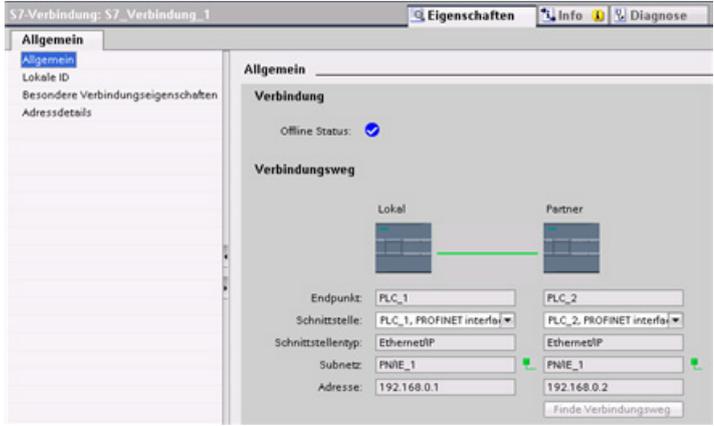
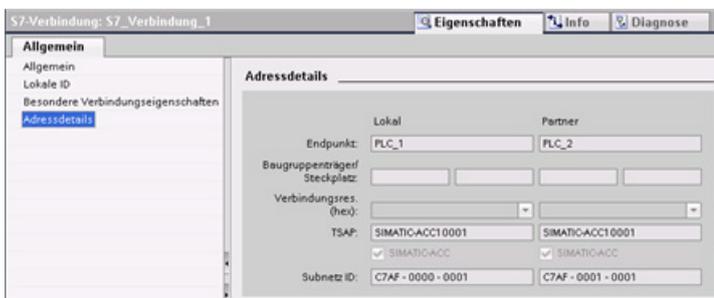
TCP, ISO-on-TCP und UDP	Verbindungseigenschaften
<p>Bei den TCP-, ISO-on-TCP- und UDP-Ethernet-Protokollen konfigurieren Sie die Verbindungen der lokalen und der Partner-CPU in den "Eigenschaften" der Anweisung (TSEND_C, TRCV_C oder TCON).</p> <p>Die Abbildung zeigt die "Verbindungseigenschaften" im 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 7- 5 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.</p> <p>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 149) und im Abschnitt "S7-Kommunikation" unter "S7-Verbindung erstellen" (Seite 170) finden Sie weitere Informationen sowie eine Liste der verfügbaren Kommunikationsanweisungen.

Tabelle 7- 6 Parameter für die CPU-Verbindung

Parameter	Definition																
Adresse	Zugewiesene IP-Adressen																
Allgemein	<table border="1"> <tr> <td>Endpunkt</td> <td>Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td>Schnittstelle</td> <td>Name der Schnittstellen</td> </tr> <tr> <td>Subnetz</td> <td>Name der Subnetze</td> </tr> <tr> <td>Schnittstellentyp</td> <td><i>Nur S7-Kommunikation:</i> Typ der Schnittstelle</td> </tr> <tr> <td>Verbindungstyp</td> <td>Typ des Ethernet-Protokolls</td> </tr> <tr> <td>Verbindungs-ID</td> <td>ID-Nummer</td> </tr> <tr> <td>Verbindungsdaten</td> <td>Datenspeicher für die lokale CPU und die Partner-CPU</td> </tr> <tr> <td>Aktive Verbindung herstellen</td> <td>Optionsfeld zum Auswählen der lokalen CPU oder der Partner-CPU als aktive Verbindung</td> </tr> </table>	Endpunkt	Name der Partner-CPU (Empfänger)	Schnittstelle	Name der Schnittstellen	Subnetz	Name der Subnetze	Schnittstellentyp	<i>Nur S7-Kommunikation:</i> 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
Endpunkt	Name der Partner-CPU (Empfänger)																
Schnittstelle	Name der Schnittstellen																
Subnetz	Name der Subnetze																
Schnittstellentyp	<i>Nur S7-Kommunikation:</i> 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																
Adressdetails	<table border="1"> <tr> <td>Endpunkt</td> <td><i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td>Baugruppenträger/Steckplatz</td> <td><i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse</td> </tr> <tr> <td>Verbindungsressource</td> <td><i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU</td> </tr> </table>	Endpunkt	<i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)	Baugruppenträger/Steckplatz	<i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse	Verbindungsressource	<i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU										
Endpunkt	<i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)																
Baugruppenträger/Steckplatz	<i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse																
Verbindungsressource	<i>Nur S7-Kommunikation:</i> Komponente des TSAP für die Konfiguration einer S7-Verbindung mit einer S7-300 oder S7-400 CPU																

Parameter		Definition
	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 (bis zu 64-K-Verbindungen). 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.

7.6 PROFIBUS

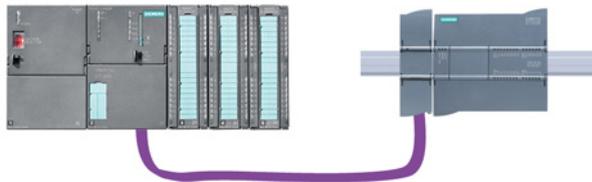
Ein PROFIBUS-System nutzt einen Bus-Master, um Slavegeräte abzufragen, die an mehreren Stellen auf einem seriellen RS485-Bus verteilt sind. Ein PROFIBUS-Slave ist ein beliebiges Peripheriegerät (E/A-Wandler, Ventil, Motorantrieb oder Messgerät), das Daten verarbeitet und die Ausgabe an den Master sendet. 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

(<http://support.automation.siemens.com/WW/llisapi.dll?func=cslib.csinfo&lang=en&objid=6GK72425DX300XE0&caller=view>) 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 ET200S 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 V4.0 können Sie maximal drei PROFIBUS-CMs je Station konfigurieren, wobei es sich um eine beliebige Kombination aus DP-Master- 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.

7.6.1 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.

7.6.2 Verweis auf die Benutzerhandbücher für PROFIBUS-CMs

Weitere Informationen

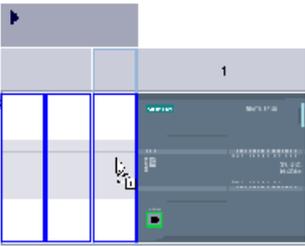
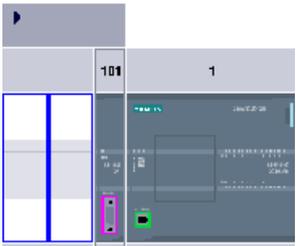
Ausführliche Informationen über die PROFIBUS-CMs finden Sie in den Handbüchern zu den jeweiligen Geräten. Diese stehen im Internet auf den Seiten des Kundensupports von Siemens Industrial Automation unter den folgenden IDs zur Verfügung:

- CM 1242-5 (<http://support.automation.siemens.com/WW/view/en/49852105>)
- CM 1243-5 (<http://support.automation.siemens.com/WW/view/en/49851842>)

7.6.3 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 7- 7 PROFIBUS CM 1243-5 (DP-Master) zur Gerätekonfiguration hinzufügen

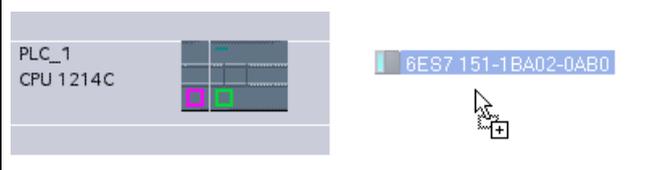
Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-5 (DP-Master)			

Im Hardwarekatalog können Sie auch die DP-Slaves hinzufügen. Um beispielsweise einen ET200 S DP-Slave hinzuzufügen, erweitern Sie im Hardwarekatalog die folgenden Behälter:

- Dezentrale E/A
- ET200 S
- Schnittstellenmodule
- PROFIBUS

Wählen Sie dann "6ES7 151-1BA02-0AB0" (IM151-1 HF) in der Liste der Bestellnummern aus und fügen Sie den ET200 S DP-Slave wie in der folgenden Abbildung gezeigt ein.

Tabelle 7- 8 ET200 S DP-Slave zur Gerätekonfiguration hinzufügen

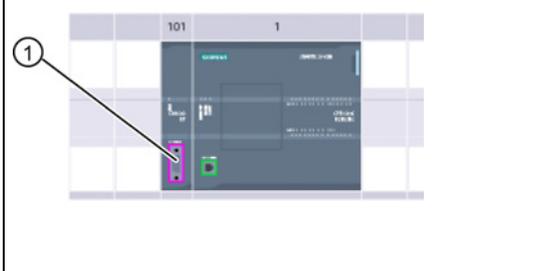
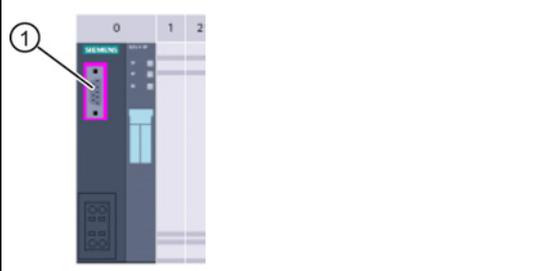
DP-Slave einfügen	Ergebnis
	

7.6.4 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 7- 9 PROFIBUS-Schnittstellen von CM 1243-5 (DP-Master) und ET200 S DP-Slave konfigurieren

CM 1243-5 (DP-Master)	ET200 S DP-Slave
	

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 7- 10 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.

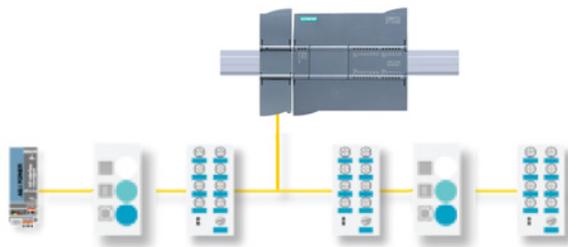
Parameter	Beschreibung
Übertragungsgeschwindigkeit	<p>Übertragungsgeschwindigkeit im konfigurierten PROFIBUS-Netzwerk: Die PROFIBUS-Übertragungsgeschwindigkeiten liegen zwischen 9,6 kBit/s und 12 MBit/s. Die Übertragungsgeschwindigkeit ist von den Eigenschaften der verwendeten PROFIBUS-Teilnehmer abhängig. Die Übertragungsgeschwindigkeit darf nicht größer sein als die vom langsamsten Teilnehmer unterstützte Geschwindigkeit.</p> <p>Die Übertragungsgeschwindigkeit wird normalerweise für den Master im PROFIBUS-Netzwerk festgelegt, wobei alle DP-Slaves automatisch die gleiche Übertragungsgeschwindigkeit nutzen (Auto-Baud).</p>

7.7 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. Mithilfe eines einzigen AS-i-Kabels können Sensoren und Aktoren (AS-i-Slavegeräte) über das CM 1243-2 an die CPU angeschlossen werden. Das CM 1243-2 wickelt die gesamte AS-i-Netzwerkkoordination ab und überträgt mit Hilfe der dem CM 1243-2 zugewiesenen E/A-Adressen Daten und Statusinformationen von den Aktoren und Sensoren an die CPU. Je nach Slavetyp 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.

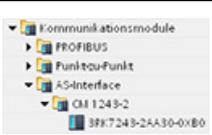
In der folgenden Abbildung ist die S7-1200 ein AS-i-Master, der das AS-i-Bedienpanel und digitale bzw. analoge Slavegeräte von E/A-Modulen steuert.



7.7.1 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 7- 11 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.

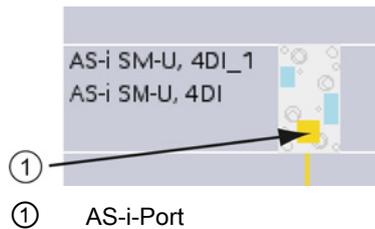
Tabelle 7- 12 AS-i-Slave zur Gerätekonfiguration hinzufügen

AS-i-Slave stecken	Ergebnis
	

7.7.2 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.



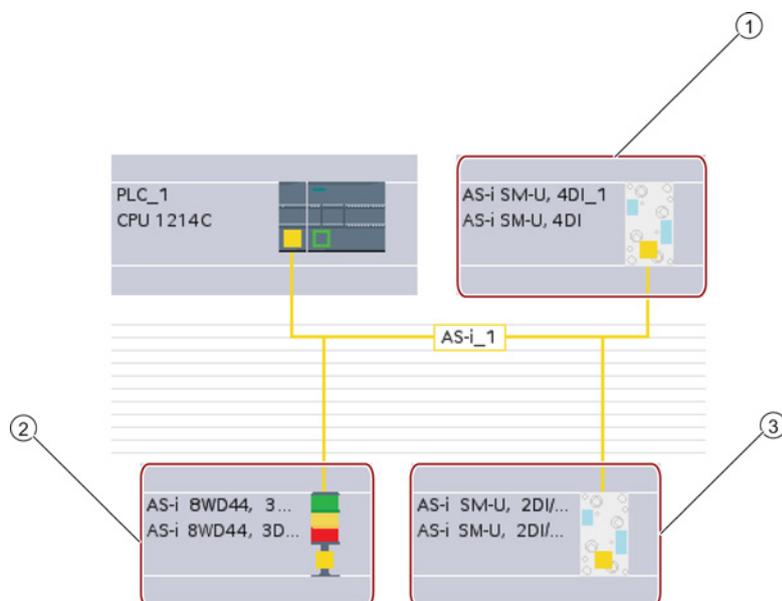
AS-i-Slaveadresse zuweisen

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.

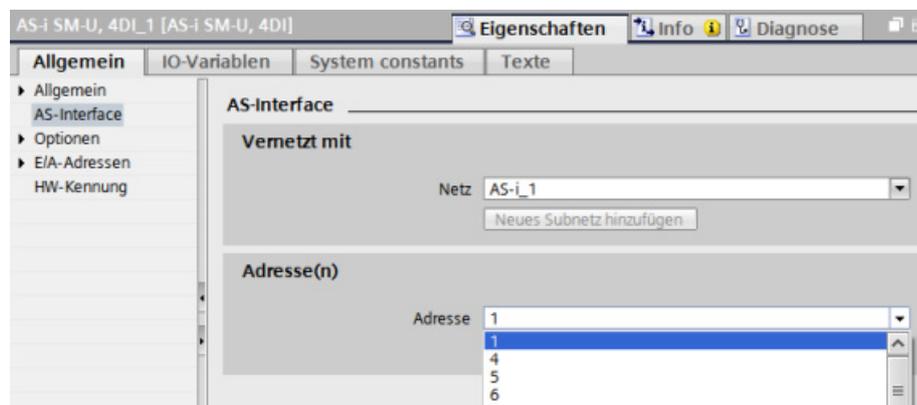
Jede Adresse im Bereich von 1 bis 31 kann einem AS-i-Slavegerät zugewiesen werden. D. h. mit anderen Worten: Es ist nicht wichtig, ob die Slaves mit Adresse 21 beginnen oder ob der erste Slave tatsächlich die Adresse 1 erhält.

Im folgenden Beispiel wurden drei AS-i-Geräten die Adressen "1" (Standardgerät), "2A" (Gerät vom Typ A/B-Knoten) und "3" (Standardgerät) zugewiesen:



- ① AS-i-Slaveadresse 1; Gerät: AS-i SM-U, 4DI; Bestellnummer: 3RG9 001-0AA00
- ② AS-i-Slaveadresse 2A; Gerät: AS-i 8WD44, 3DO, A/B; Bestellnummer: 8WD4 428-0BD
- ③ AS-i-Slaveadresse 3; Gerät: AS-i SM-U, 2DI/2DO; Bestellnummer: 3RG9 001-0AC00

Geben Sie hier die AS-i-Slaveadresse ein:



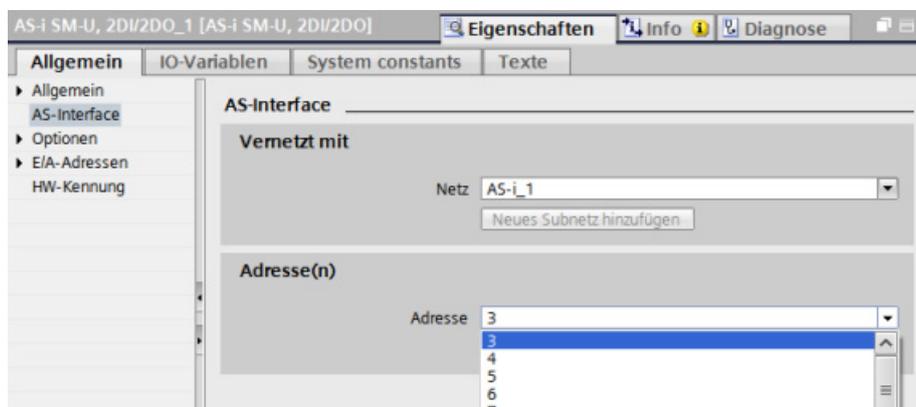
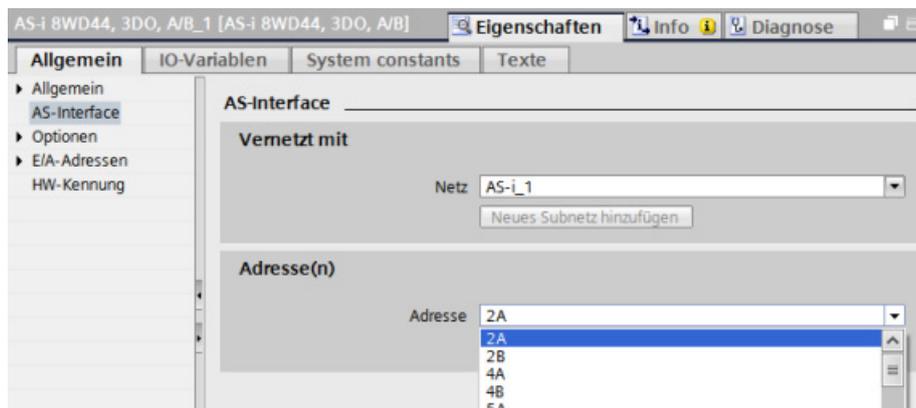


Tabelle 7- 13 Parameter für die AS-i-Schnittstelle

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

7.8 S7-Kommunikation

7.8.1 Anweisungen GET und PUT

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.

STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

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.

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 91) in der Gerätekonfiguration der CPU im Register "Eigenschaften" des Inspektorfensters unter der Eigenschaft "Schutz" aktivieren.

7.8.2 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

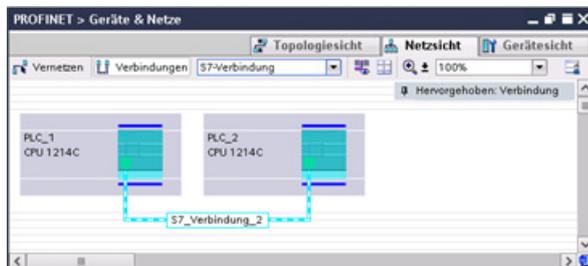
Soll auch der clientseitige Zugriff auf CPU-Daten möglich sein, d. h., sollen die Kommunikationsdienste der CPU nicht beschränkt werden, beachten Sie die Anweisungen unter "Zugriffsschutz für die S7-1200 CPU".

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".



Klicken Sie auf die Schaltfläche "Hervorgehoben: Verbindung", um den Konfigurationsdialog "Eigenschaften" der Kommunikationsanweisung aufzurufen.

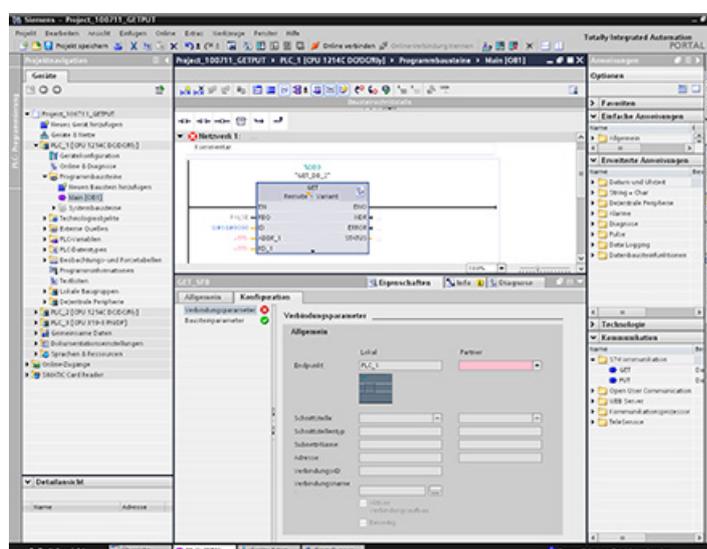
Siehe auch

Netzwerkverbindung erstellen (Seite 144)

7.8.3 Parametrieren der GET/PUT-Verbindung

Die Zuweisung der Verbindungsparameter über die Anweisungen GET und PUT ist eine Hilfe für den Anwender beim Konfigurieren der S7-Verbindungen für die CPU-zu-CPU-Kommunikation.

Die Zuweisung der Verbindungsparameter mithilfe der Anweisungen GET oder PUT wird gestartet, nachdem ein GET- oder PUT-Baustein eingefügt wurde.



Nach Auswahl eines Teils der Anweisung werden im Inspektorfenster die Eigenschaften der Verbindung angezeigt. Definieren Sie die Kommunikationsparameter über das Register "Konfiguration" im Dialog "Eigenschaften" der Kommunikationsanweisung.

Nach Einfügen eines Bausteins GET oder PUT wird automatisch das Register "Konfiguration" angezeigt und die Seite "Verbindungsparameter" aufgerufen. Auf dieser Seite kann der Anwender die benötigte S7-Verbindung konfigurieren und den Parameter "Verbindungs-ID" festlegen, der vom Bausteinparameter "ID" referenziert wird. Auf der Seite "Bausteinparameter" kann der Anwender weitere Bausteinparameter konfigurieren.

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 91) in der Gerätekonfiguration der CPU im Register "Eigenschaften" des Inspektorfensters unter der Eigenschaft "Schutz" aktivieren.

7.9 GPRS

7.9.1 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 Querkommunikation zwischen Stationen.

Die Querkommunikation zwischen Stationen ist nur über das GSM-Netz möglich. Bei der 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.

- 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 Kommunikation über das GSM-Netz nutzen soll, sind neben der Stationsausrüstung folgende Voraussetzung erforderlich:

- Ein Vertrag mit einem geeigneten GSM-Netzbetreiber

Wenn GPRS genutzt werden soll, dann muss der Vertrag die Nutzung des Dienstes GPRS 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

Im Folgenden finden Sie einige Konfigurationsbeispiele für Stationen mit CP 1242-7.

SMS-Versand

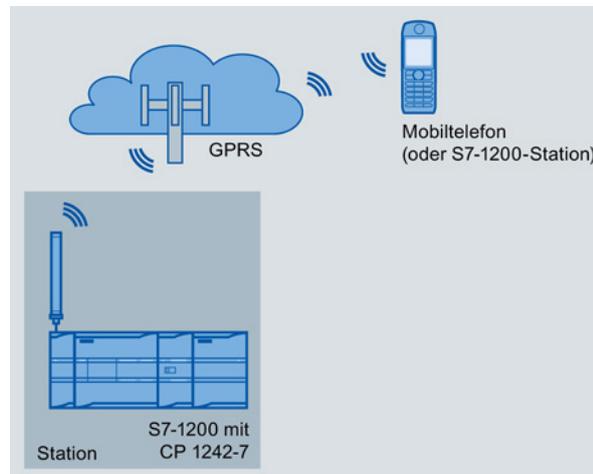


Bild 7-1 SMS-Versand einer S7-1200-Station

Eine SIMATIC S7-1200 mit CP 1242-7 kann Meldungen per SMS an ein projektiertes Mobiltelefon oder eine projektierte S7-1200-Station versenden.

Telecontrol durch eine Zentrale

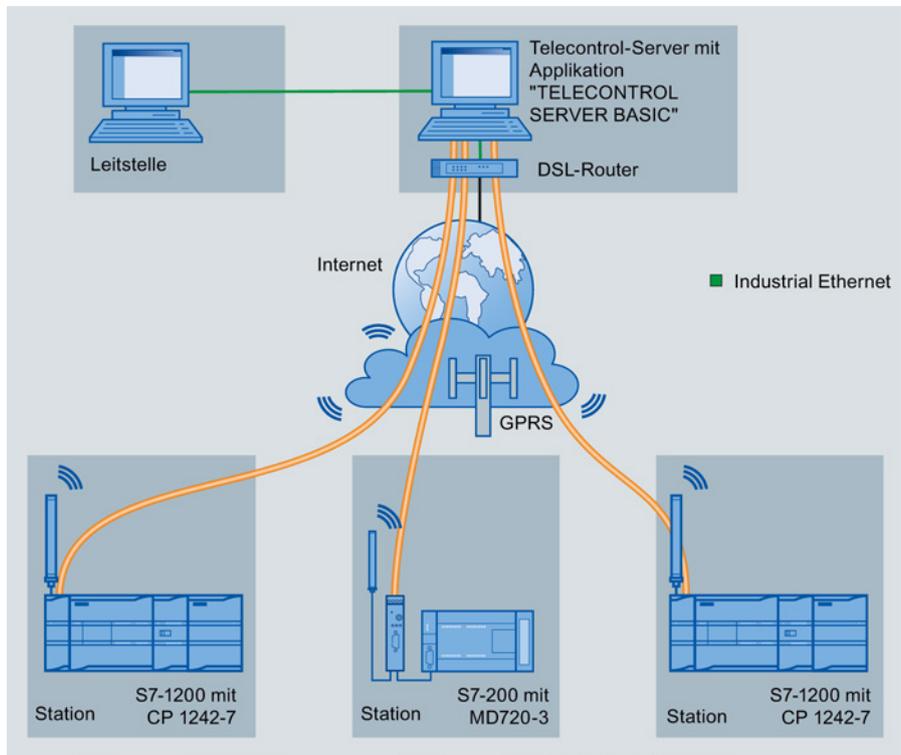


Bild 7-2 Kommunikation von S7-1200-Stationen mit einer Zentrale

Bei den Telecontrol-Anwendungen kommunizieren SIMATIC S7-1200-Stationen mit CP 1242-7 über das GSM-Netz und Internet mit einer Zentrale. Auf dem Telecontrol-Server in der Zentrale ist die Applikation "TELECONTROL SERVER BASIC" 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 Internet an den Telecontrol-Server in der Zentrale gesendet. Der Telecontrol-Server dient der Steuerung und Überwachung der entfernten Stationen.
- Kommunikation zwischen Station und einem Zentrale-PC mit OPC-Client
Wie im ersten Fall kommunizieren die Stationen mit dem Telecontrol-Server. Mithilfe des OPC-Servers von TELECONTROL SERVER BASIC tauscht der Telecontrol-Server die Daten mit einem Zentrale-PC aus. Auf dem Zentrale-PC kann beispielsweise WinCC mit integriertem OPC-Client installiert sein.
- Querkommunikation zwischen Stationen über eine Zentrale
Für die Querkommunikation zwischen Stationen leitet der Telecontrol-Server die Telegramme der Sender-Station an die Empfänger-Station weiter.

Direkte Querkommunikation zwischen Stationen

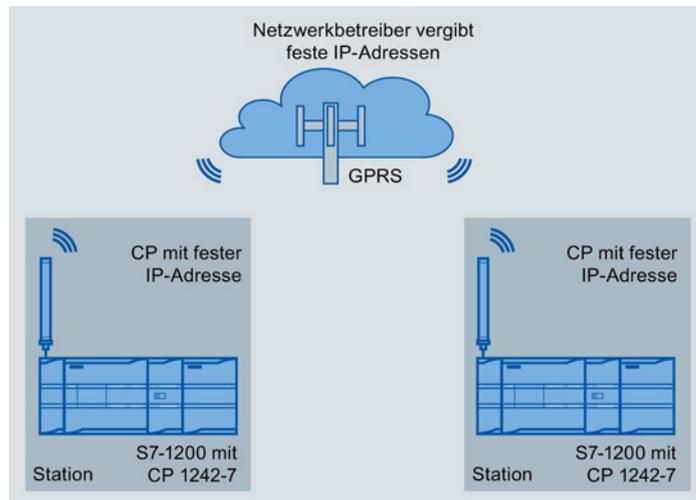


Bild 7-3 Direkte Querkommunikation von zwei S7-1200-Stationen

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

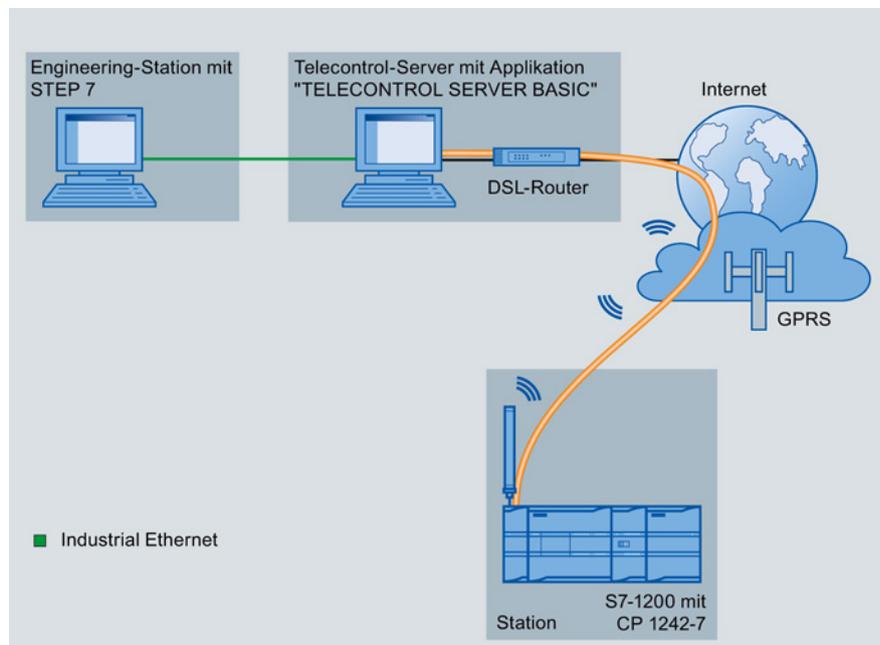


Bild 7-4 TeleService über GPRS

Bei TeleService über GPRS kommuniziert eine Engineering-Station, auf der STEP 7 installiert ist, über das GSM-Netz und das Internet mit einer SIMATIC S7-1200-Station mit CP 1242-7. Die Verbindung wird über einen Telecontrol-Server geführt, der als Vermittler dient und mit dem Internet verbunden ist.

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 Leitzentrale

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 Applikation "TELECONTROL SERVER BASIC" installiert. Über die integrierte OPC-Server-Funktion kommuniziert der Telecontrol-Server mit einem übergeordneten zentralen Leitsystem.

- Querkommunikation zwischen S7-1200-Stationen über ein GSM-Netz

Die Querkommunikation zwischen entfernten Stationen mit CP 1242-7 kann auf zwei unterschiedliche Arten abgewickelt werden:

- Indirekte Kommunikation 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 immer über den Telecontrol-Server. Der CP 1242-7 arbeitet in der 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 zu adressieren. Die möglichen Kommunikationsdienste und Sicherheitsfunktionen (z. B. VPN) hängen dabei vom Angebot des Netzwerkbetreibers ab. Der CP 1242-7 arbeitet in der 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 Applikation "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

Weitere Dienste und Funktionen des CP 1242-7

- Uhrzeitsynchronisation des CP über Internet
Die Uhrzeit des CP können Sie folgendermaßen stellen:
 - In der Betriebsart "Telecontrol" wird die Uhrzeit vom Telecontrol-Server übertragen. Der CP stellt damit seine Uhrzeit.
 - In der Betriebsart "GPRS direkt" kann der CP die Uhrzeit über SNTP anfordern.
Zur Synchronisation der CPU-Uhrzeit können Sie die aktuelle Uhrzeit mithilfe eines Bausteins aus dem CP auslesen.
- Zwischenspeicherung der zu sendenden Telegramme bei Verbindungsproblemen
- Protokollierung des Datenvolumens
Die übertragenen Datenvolumina werden protokolliert und können zu weiteren Zwecken ausgewertet werden.

Projektierung und Baugruppentausch

Für die Projektierung der Baugruppe ist folgendes Projektierungswerkzeug erforderlich:

STEP 7 Version V11.0 SP1 oder höher

Für STEP 7 V11.0 SP1 benötigen Sie zusätzlich das Support Package "CP 1242-7" (HSP0003001).

Für die Prozessdatenübertragung per GPRS verwenden Sie im Anwenderprogramm der Station die Telecontrol-Kommunikationsanweisungen.

Die Projektierungsdaten des CP 1242-7 werden auf der jeweils lokalen CPU gespeichert. Dies ermöglicht im Ersatzteillfall den einfachen Austausch des CP.

Sie können bis zu drei Baugruppen des Typs CP 1242-7 pro S7-1200 stecken. Damit lassen sich beispielsweise redundante Kommunikationspfade aufbauen.

Elektrische Anschlüsse

- Spannungsversorgung des CP 1242-7
Der CP hat einen eigenen Anschluss für die externe Spannungsversorgung DC 24 V.
- Funk-Schnittstelle für das GSM-Netz
Für die GSM-Kommunikation ist eine externe Antenne erforderlich. Diese wird über die SMA-Buchse des CP angeschlossen.

Die 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



Bild 7-5 GSM/GPRS-Antenne ANT794-4MR

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



Bild 7-6 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.

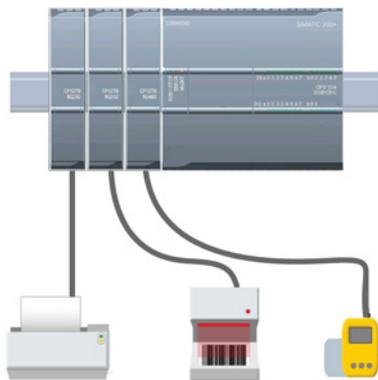
7.10 Kommunikationsprotokolle PtP, USS und Modbus

7.10.1 Punkt-zu-Punkt-Kommunikation

Die CPU unterstützt die folgende Punkt-zu-Punkt-Kommunikation (PtP) für zeichenbasierte serielle Protokolle.

- PtP (Seite 180)
- USS (Seite 181)
- Modbus (Seite 183)

Das Punkt-zu-Punkt-Protokoll 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.

7.10.2 Mit den seriellen Kommunikationsschnittstellen arbeiten

Zwei Kommunikationsmodule (CMs) und ein Kommunikationsboard (CB) bieten die Schnittstelle für die PtP-Kommunikation:

- CM 1241 RS232
- CM 1241 RS422/485
- CB 1241 RS485

Sie können bis zu drei CMs (jeden Typs) plus ein CB, insgesamt vier Kommunikationsschnittstellen anschließen. Bauen Sie das CM links von der CPU oder eines anderen CMs ein. Bauen Sie das CB auf der Vorderseite der CPU ein. Ausführliche Anweisungen zum Ein- und Ausbau des Moduls finden Sie im Kapitel "Einbau" im S7-1200 Systemhandbuch.

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.

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.

7.10.3 PtP-Operationen

Mit den Anweisungen PORT_CFG, SEND_CFG und RCV_CFG können Sie die Konfiguration aus Ihrem Anwenderprogramm heraus ändern.

- PORT_CFG ändert die Portparameter wie z. B. die Baudrate.
- SEND_CFG ändert die Konfiguration der Parameter für die serielle Übertragung.
- RCV_CFG ändert die Konfiguration der Parameter eines seriellen Empfängers in einem Kommunikations-Port. Diese Anweisung konfiguriert die Bedingungen, die den Beginn und das Ende einer empfangenen Meldung kennzeichnen. Meldungen, die diese Bedingungen erfüllen, werden von der Anweisung RCV_PTP empfangen.

Die dynamischen Konfigurationsänderungen werden in der CPU nicht dauerhaft gespeichert. Nach dem Einschalten wird die statische Anfangskonfiguration aus der Gerätekonfiguration verwendet.

Die Anweisungen SEND_PTP, RCV_PTP und RCV_RST steuern die PtP-Kommunikation:

- SEND_PTP überträgt den angegebenen Puffer an das CM oder CB. Die CPU führt das Anwenderprogramm weiterhin aus, während das Modul die Daten mit der angegebenen Baudrate sendet.
- RCV_PTP prüft die Meldungen, die im CM oder CB empfangen wurden. Wenn eine Meldung verfügbar ist, wird sie zur CPU übertragen.
- Die Anweisung RCV_RST setzt den Empfangspuffer zurück.

Jedes CM oder CB kann bis zu 1 KB puffern. Dieser Puffer kann mehreren empfangenen Meldungen zugeordnet werden.

Die Anweisungen SGN_SET und SGN_GET gelten nur für das RS232-CM. Verwenden Sie diese Anweisungen, um die RS232-Kommunikationssignale zu lesen oder zu setzen.

7.10.4 USS-Anweisungen

Die S7-1200 unterstützt das USS-Protokoll und stellt die Anweisungen bereit, die speziell für die Kommunikation mit Antrieben über den RS485-Port eines CMs oder CBs konzipiert sind. Mit den USS-Anweisungen können Sie den physikalischen Antrieb und die Parameter zum Lesen und Schreiben des Antriebs steuern. Jedes RS485-CM oder -CB unterstützt bis zu 16 Antriebe.

- Über die Anweisung USS_PORT wird die eigentliche Kommunikation zwischen der CPU und allen Antrieben, die an ein CM oder CB angeschlossen sind, abgewickelt. Fügen Sie in Ihrer Anwendung für jedes CM oder CB eine andere Anweisung USS_PORT ein. Stellen Sie sicher, dass das Anwenderprogramm die Anweisung USS_PORT schnell genug ausführt, um eine Kommunikationszeitüberschreitung durch den Antrieb zu verhindern. Sie können die Anweisung USS_PORT im Programmzyklus-OB oder in einem beliebigen Alarm-OB verwenden.
- Mit der Anweisung USS_DRV wird ein angegebener Antrieb im USS-Netzwerk angesteuert. Die Ein- und Ausgangsparameter der Anweisung USS_DRV entsprechen den Zuständen und Bedienfunktionen des Antriebs. Sind 16 Antriebe im Netzwerk vorhanden, so muss das Programm die Anweisung USS_DRV mindestens 16 Mal enthalten, d. h. einmal für jeden Antrieb.

Stellen Sie sicher, dass die CPU die Anweisung USS_DRV mit der Geschwindigkeit ausführt, die für die Steuerung der Antriebsfunktionen erforderlich ist. Sie können die Anweisung USS_DRV nur in einem Programmzyklus-OB verwenden.

- Mit den Anweisungen 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 so viele dieser Anweisungen enthalten, wie benötigt werden. Es kann jedoch zu einem gegebenen Zeitpunkt immer nur eine Lese- oder Schreibanforderung für einen Antrieb aktiv sein. Sie können die Anweisungen USS_RPM und USS_WPM nur in einem Programmzyklus-OB verwenden.

Ein Instanz-DB enthält temporären Speicherbereich und Pufferspeicher für alle mit den einzelnen CMs oder CBs verbundenen Antriebe im USS-Netzwerk. Der Instanz-DB wird von den USS-Anweisungen für einen Antrieb verwendet, um gemeinsam auf die gespeicherten Daten zuzugreifen.

Zeit für die Kommunikation mit dem Antrieb berechnen

Die Kommunikation mit dem Antrieb läuft asynchron zum Zyklus der CPU ab. Die CPU 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 7- 14 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

7.10.5 Modbus-Anweisungen

Die CPU unterstützt Modbus-Kommunikation über verschiedene Netzwerke:

- 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 (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 Client/Server-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 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.

WARNUNG

Wenn sich ein Angreifer physischen Zugriff auf Ihre Netzwerke verschaffen kann, kann er möglicherweise Daten lesen und schreiben.

Das TIA Portal, die CPU und HMI-Geräte (außer HMI-Geräten über GET/PUT) nutzen die sichere Kommunikation, die Schutz vor Replay- und Man-in-the-Middle-Angriffen bietet. Nach Aktivierung der Kommunikation erfolgt der Austausch signierter Meldungen in Klartext, wodurch ein Angreifer Daten lesen kann, die Daten jedoch vor unberechtigten Schreibzugriffen geschützt sind. Das TIA Portal, nicht der Kommunikationsprozess, verschlüsselt die Daten von Knowhow-geschützten Bausteinen.

Alle anderen 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.

Informationen und Empfehlungen zur Sicherheit finden Sie in unseren "Operational Guidelines für Industrial Security" auf der Website "Service & Support":

www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_de.pdf
(http://www.industry.siemens.com/topics/global/en/industrial-security/Documents/operational_guidelines_industrial_security_en.pdf)

Hinweis

Modbus TCP funktioniert erst ab CPU Firmware Release V1.02 einwandfrei. Der Versuch, die Modbus-Anweisungen mit einer früheren Firmware-Version auszuführen, führt zu einem Fehler.

Tabelle 7- 15 Modbus-Anweisungen

Art der Kommunikation	Anweisung
Modbus RTU (RS232 oder RS485)	MB_COMM_LOAD: Eine Ausführung von MB_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 den Anweisungen MB_MASTER oder or MB_SLAVE verwendet werden.
	MB_MASTER: Mit der Modbus-Master-Anweisung kann die CPU als Modbus-RTU-Mastergerät für die Kommunikation mit einem oder mehreren Modbus-Slavegeräten eingesetzt werden.
	MB_SLAVE: Mit der Modbus-Slave-Anweisung kann die CPU als Modbus-RTU-Slavegerät für die Kommunikation mit einem Modbus-Mastergerät eingesetzt werden.
Modbus TCP (PROFINET)	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.

Die Modbus-Anweisungen nutzen keine Kommunikations-Alarmereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss die Anweisungen MB_MASTER/MB_SLAVE oder MB_Client/MB_Server auf abgeschlossene Sende- und Empfangsvorgänge abfragen.

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 von Modbus-Meldungen auslösen und Antworten vom Server empfangen
3. Gegebenenfalls die Verbindungsunterbrechung zwischen Client und Server auslösen, um die Verbindung mit einem anderen Server zu ermöglichen

Einfache PID-Regelung

STEP 7 bietet die folgenden PID-Anweisungen 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.

Hinweis

Änderungen, die Sie an der PID-Konfiguration vornehmen und im Betriebszustand RUN ins Zielsystem laden, werden erst dann wirksam, wenn die CPU von STOP in RUN geht. Im Dialog "PID-Parameter" über "Startwertsteuerung" vorgenommene Änderungen werden sofort wirksam.

Beide PID-Anweisungen (PID_3Step und PID_Compact) können den P-, I- und D-Anteil während des Anlaufs berechnen (sofern die "Erstoptimierung" konfiguriert ist). Sie können die Anweisung auch für die "Feineinstellung" konfigurieren, um die Parameter zu optimieren. Sie brauchen die Parameter nicht manuell festzulegen.

Hinweis

Führen Sie die PID-Anweisung in einem wiederholten konstanten Zeitintervall der Abtastzeit aus (nach Möglichkeit in einem Zyklus-OB).

Weil der PID-Regler eine bestimmte Zeit benötigt, um auf Änderungen des Stellwerts zu reagieren, berechnen Sie den Ausgangswert nicht in jedem Zyklus. Führen Sie die PID-Anweisung nicht im Zyklus-OB des Hauptprogramms aus (z. B. OB 1).

Die Abtastzeit des PID-Algorithmus stellt die Zeit zwischen zwei Berechnungen des Ausgangswerts (Stellwerts) dar. Der Ausgangswert wird während der Selbsteinstellung berechnet und auf ein Vielfaches der Zykluszeit gerundet. Alle anderen Funktionen der PID-Anweisung werden bei jedem Aufruf ausgeführt.

PID-Algorithmus

Der PID-Regler (Proportional/Integral/Differential) misst das Zeitintervall zwischen zwei Aufrufen und wertet dann die Ergebnisse aus, um die Abtastzeit zu überwachen. Bei jedem Wechsel des Betriebszustands sowie beim ersten Anlauf wird ein Mittelwert der Abtastzeit errechnet. Dieser Wert dient als Referenzwert für die Überwachungsfunktion und zur Berechnung. Bei der Überwachung wird die aktuelle Messzeit zwischen zwei Aufrufen und der Mittelwert der für den Regler festgelegten Abtastzeit überwacht.

Der Ausgangswert des PID-Reglers besteht aus drei Anteilen:

- P (Proportional): Bei Berechnung mit P-Anteil ist der Ausgangswert proportional zur Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert).
- I (Integral): Bei Berechnung mit I-Anteil steigt der Ausgangswert proportional zur Dauer der Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert), um letztlich die Differenz zu korrigieren.
- D (Differential): Bei Berechnung mit D-Anteil steigt der Ausgangswert als Funktion der zunehmenden Änderungsgeschwindigkeit der Differenz zwischen dem Sollwert und dem Prozesswert (Eingangswert). Der Ausgangswert wird so schnell wie möglich an den Sollwert angeglichen.

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID_Compact anhand folgender Formel.

$$y = K_p \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K _p	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenzierverszug (D-Anteil)
T _i	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T _D	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

Der PID-Regler berechnet den Ausgangswert für die Anweisung PID_3Step anhand folgender Formel.

$$\Delta y = K_p \cdot s \cdot \left[(b \cdot w - x) + \frac{1}{T_i \cdot s} (w - x) + \frac{T_d \cdot s}{a \cdot T_d \cdot s + 1} (c \cdot w - x) \right]$$

y	Ausgangswert	x	Prozesswert
w	Sollwert	s	Laplace-Operator
K _p	Proportionale Verstärkung (P-Anteil)	a	Koeffizient für den Differenzierverszug (D-Anteil)
T _i	Integralzeit (I-Anteil)	b	Gewichtung des Proportionalanteils (P-Anteil)
T _D	Differentialzeit (D-Anteil)	c	Gewichtung des Differentialanteils (D-Anteil)

8.1 Anweisung PID und Technologieobjekt einfügen

STEP 7 bietet zwei Anweisungen für den PID-Regler:

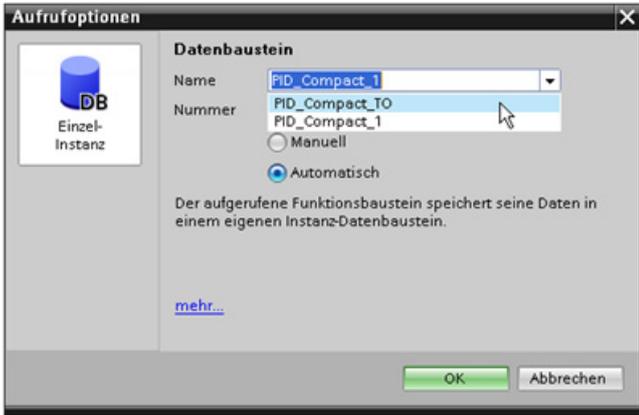
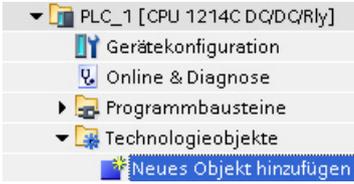
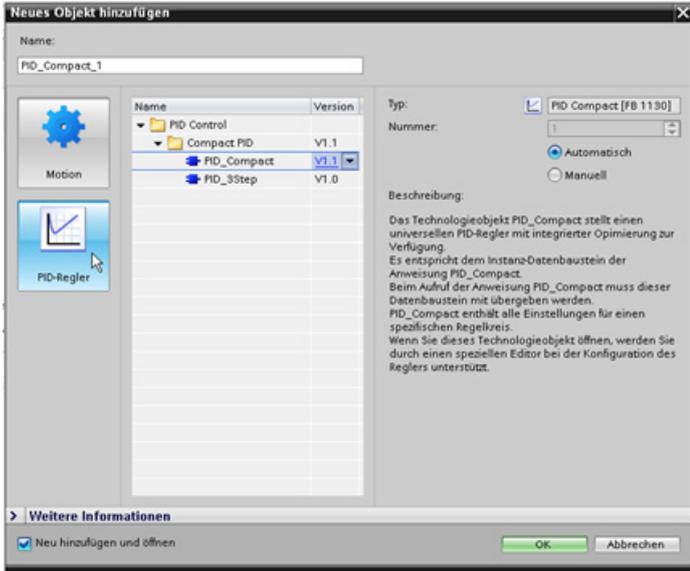
- Die Anweisung PID_Compact und das zugehörige Technologieobjekt bieten einen universellen PID-Regler mit Einstellung. Das Technologieobjekt enthält alle Einstellungen für den Regelkreis.
- Die Anweisung PID_3Step und das zugehörige Technologieobjekt bieten einen PID-Regler mit bestimmten Einstellungen für motorbetriebene Ventile. Das Technologieobjekt enthält alle Einstellungen für den Regelkreis. Der Regler PID_3Step bietet zwei zusätzliche Boolesche Ausgänge.

Nach dem Anlegen des Technologieobjekts müssen Sie die Parameter konfigurieren (Seite 204). Sie passen auch die Parameter für die Selbsteinstellung an ("Erstoptimierung" während des Anlaufs oder manuelle "Feineinstellung"), um den PID-Regler in Betrieb zu nehmen (Seite 207).

Tabelle 8- 1 Anweisung PID und Technologieobjekt einfügen

<p>Wenn Sie eine PID-Anweisung in Ihr Anwenderprogramm einfügen, legt STEP 7 automatisch ein Technologieobjekt und einen Instanz-DB für die Anweisung an. Der Instanz-DB enthält alle Parameter für die PID-Anweisung. Jede PID-Anweisung benötigt einen eigenen, eindeutigen Instanz-DB, um ordnungsgemäß zu funktionieren.</p> <p>Nach dem Einfügen der PID-Anweisung und dem Erstellen des Technologieobjekts sowie des Instanz-DBs konfigurieren Sie die Parameter für das Technologieobjekt (Seite 204).</p>	
---	---

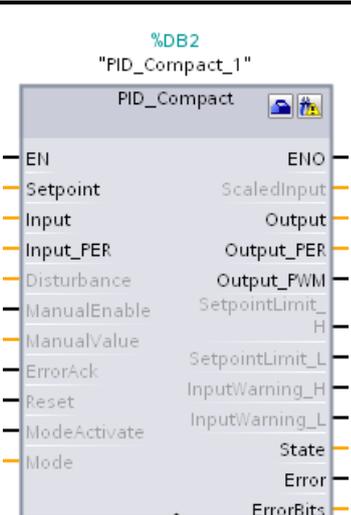
Tabelle 8-2 (Optional) Technologieobjekt in der Projektnavigation anlegen

<p>Sie können Technologieobjekte für Ihr Projekt auch vor dem Einfügen der PID-Anweisung anlegen. Wenn Sie das Technologieobjekt vor dem Einfügen der PID-Anweisung in Ihr Anwenderprogramm anlegen, können Sie das Technologieobjekt später beim Einfügen der PID-Anweisung auswählen.</p>	
<p>Um ein Technologieobjekt auszuwählen, doppelklicken Sie in der Projektnavigation auf das Symbol "Neues Objekt hinzufügen".</p>	
<p>Klicken Sie auf das Symbol "Regeln" und wählen Sie das Technologieobjekt für die Art des PID-Reglers aus (PID_Compact oder PID_3Step). Sie können wahlweise auch einen Namen für das Technologieobjekt eingeben.</p> <p>Klicken Sie auf "OK", um das Technologieobjekt anzulegen.</p>	

8.2 Operation PID_Compact

Die Anweisung PID_Compact bietet einen universellen PID-Regler mit integrierter Selbsteinstellung für den Automatik- und Handbetrieb.

Tabelle 8-3 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>

- STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.
- Im SCL-Beispiel ist "PID_Compact_1" der Name des Instanz-DBs.

Tabelle 8-4 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Setpoint	IN	Real	Sollwert des PID-Reglers im Automatikbetrieb. (Standardwert: 0.0)
Input	IN	Real	Eine Variable des Anwenderprogramms wird als Quelle des Prozesswerts verwendet. (Standardwert: 0.0) Wenn Sie den Parameter Input verwenden, müssen Sie Config.InputPerOn = FALSE einstellen.
Input_PER	IN	Word	Ein Analogeingang wird als Quelle des Prozesswerts verwendet. (Standardwert: W#16#0) Wenn Sie den Parameter Input_PER verwenden, müssen Sie Config.InputPerOn = TRUE einstellen.
Disturbance	IN	Real	Störgröße oder Vorsteuerungswert
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. (Standardwert: FALSE): <ul style="list-style-type: none"> Eine Flanke von FALSE nach TRUE aktiviert den "Handbetrieb", während sich bei Zustand = 4 die Betriebsart nicht verändert. Solange ManualEnable = TRUE ist, können Sie die Betriebsart nicht mit einer steigenden Flanke an ModeActivate wechseln und auch nicht das Dialogfeld für die Inbetriebnahme verwenden. Eine Flanke von TRUE nach FALSE aktiviert die von Mode zugewiesene Betriebsart. Hinweis: Es wird empfohlen, die Betriebsart lediglich über ModeActivate zu ändern.
ManualValue	IN	Real	Ausgangswert für Handbetrieb. (Standardwert: 0.0) Verwendbar sind Werte von Config.OutputLowerLimit bis Config.OutputUpperLimit.
ErrorAck	IN	Bool	Setzt die ErrorBits und Warnausgänge zurück. Flanke von FALSE nach TRUE
Reset	IN	Bool	Startet den Regler neu. (Standardwert: FALSE): <ul style="list-style-type: none"> Flanke von FALSE nach TRUE: <ul style="list-style-type: none"> Wechselt in den "inaktiven" Modus Setzt die ErrorBits und Warnausgänge zurück Löscht den Integralanteil Speichert die PID-Parameter Solange Reset = TRUE ist, bleibt PID_Compact im "inaktiven" Modus (Zustand = 0). Flanke von TRUE nach FALSE: <ul style="list-style-type: none"> PID_Compact wechselt in die Betriebsart, die im Parameter Mode gespeichert ist.
ModeActivate	IN	Bool	PID_Compact schaltet in den Betriebszustand, der im Parameter Mode gespeichert ist. Flanke von FALSE nach TRUE:
Mode	IN	Int	Die gewünschte PID-Betriebsart; wird bei einer steigenden Flanke am Eingang Mode Activate aktiviert.
ScaledInput	OUT	Real	Skalierter Prozesswert. (Standardwert: 0.0)
Output ¹	OUT	Real	Ausgangswert im Format REAL. (Standardwert: 0.0)
Output_PER ¹	OUT	Word	Analoger Ausgangswert. (Standardwert: W#16#0)

Parameter und Datentyp		Datentyp	Beschreibung
Output_PWM ¹	OUT	Bool	Ausgangswert für die Impulsdauermodulation. (Standardwert: FALSE) Der Ausgangswert wird aus Ein- und Aus-Zeiten gebildet.
SetpointLimit_H	OUT	Bool	Sollwert oberer Grenzwert. (Standardwert: FALSE) Wenn SetpointLimit_H = TRUE ist, dann ist der absolute obere Grenzwert des Sollwerts erreicht (Setpoint ≥ Config.SetpointUpperLimit). Der Sollwert ist auf Config.SetpointUpperLimit begrenzt.
SetpointLimit_L	OUT	Bool	Sollwert unterer Grenzwert. (Standardwert: FALSE) Wenn SetpointLimit_L = TRUE ist, dann ist der absolute untere Grenzwert des Sollwerts erreicht (Setpoint ≤ Config.SetpointLowerLimit). Der Sollwert ist auf Config.SetpointLowerLimit begrenzt.
InputWarning_H	OUT	Bool	Wenn InputWarning_H = TRUE, hat der Prozesswert die obere Warngrenze erreicht oder überschritten. (Standardwert: FALSE)
InputWarning_L	OUT	Bool	Wenn InputWarning_L = TRUE, hat der Prozesswert die untere Warngrenze erreicht oder unterschritten. (Standardwert: FALSE)
State	OUT	Int	Aktuelle Betriebsart des PID-Reglers. (Standardwert: 0) Die Betriebsart können Sie mit dem Eingangsparameter Mode und einer steigenden Flanke an ModeActivate ändern: <ul style="list-style-type: none"> • State = 0: Inaktiv • State = 1: Erstopptimierung • State = 2: Manuelle Feineinstellung • State = 3: Automatikbetrieb • State = 4: Handbetrieb • State = 5: Ersatzausgangswert mit Fehlerüberwachung
Error	OUT	Bool	Wenn Error = TRUE, steht mindestens eine Fehlermeldung in diesem Zyklus an. (Standardwert: FALSE) Hinweis: Der Parameter Error in V1.x PID war das Feld der ErrorBits mit den Fehlercodes. Jetzt zeigt ein Boolescher Merker an, dass ein Fehler aufgetreten ist.
ErrorBits	OUT	DWord	In der Parametertabelle (Seite 193) der ErrorBits für die Anweisung PID_Compact werden die anstehenden Fehlermeldungen beschrieben. (Standardwert: DW#16#0000 (kein Fehler)). ErrorBits ist remanent und wird bei einer steigenden Flanke an Reset oder ErrorAck zurückgesetzt. Hinweis: In V1.x war der Parameter ErrorBits als Parameter Error definiert und nicht vorhanden.

¹ Sie können die Ausgänge der Parameter Output, Output_PER und Output_PWM parallel verwenden.

Funktionsweise des Reglers PID_Compact

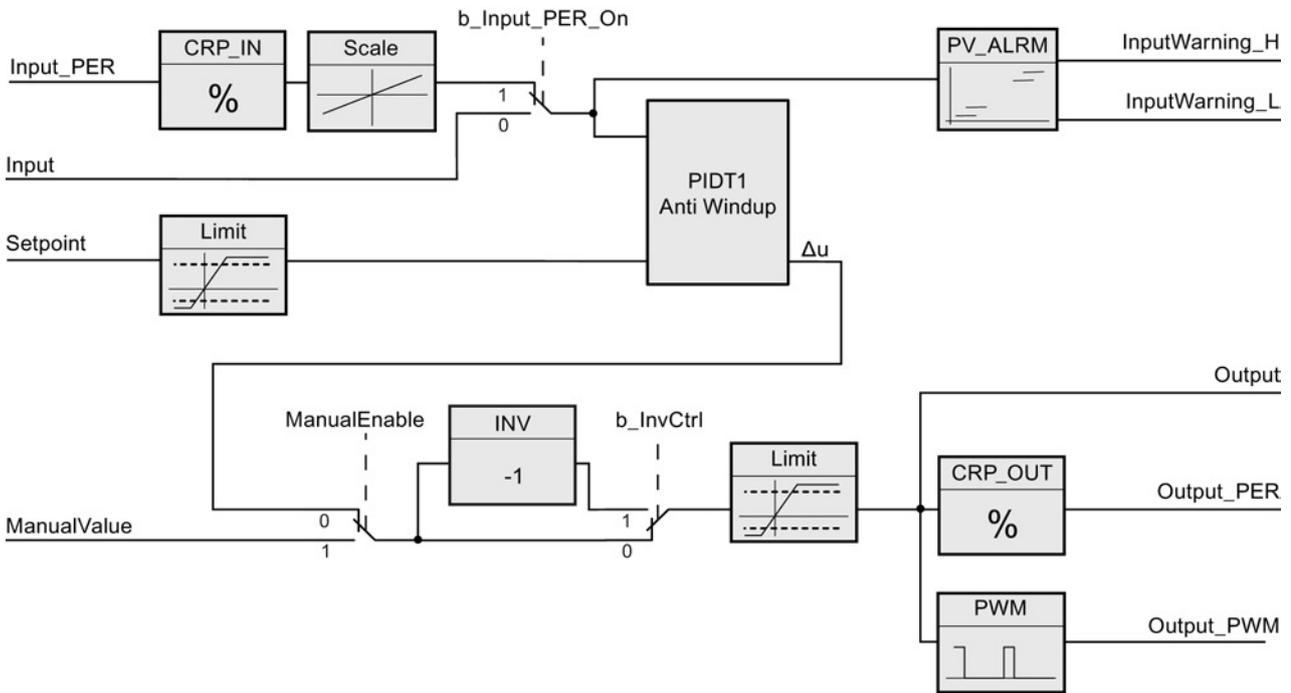


Bild 8-1 Funktionsweise des Reglers PID_Compact

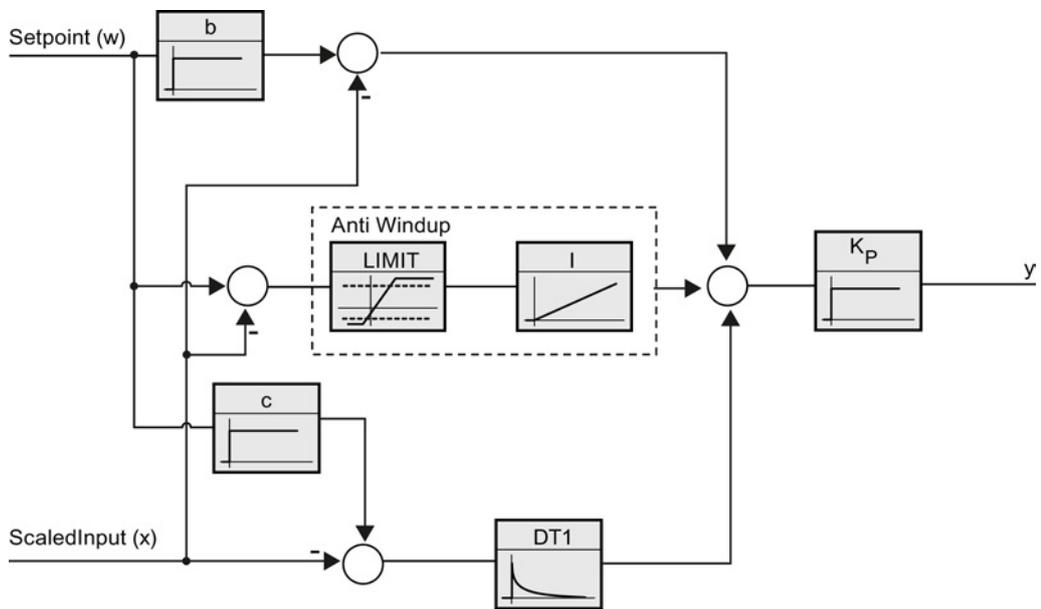


Bild 8-2 Funktionsweise des Reglers PID_Compact als PWD1-Regler mit Anti-Windup

8.3 ErrorBit-Parameter der Anweisung PID_Compact

Stehen mehrere Fehler an, werden die Werte der Fehlercodes mittels binärer Addition angezeigt. Die Anzeige von Fehlercode 0003 beispielsweise weist darauf hin, dass auch die Fehler 0001 und 0002 anstehen.

Tabelle 8- 5 ErrorBit-Parameter der Anweisung PID_Compact

ErrorBit (DW#16#...)	Beschreibung
0000	Kein Fehler
0001 ^{1,2}	Der Parameter Input liegt außerhalb der Prozesswertgrenzen. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2,3}	Ungültiger Wert am Parameter Input_PER. Prüfen Sie, ob am Analogeingang ein Fehler anliegt.
0004 ⁴	Fehler während der Feineinstellung. Schwingung des Prozesswerts konnte nicht erhalten werden.
0008 ⁴	Fehler beim Start der Erstopptimierung. Der Prozesswert ist zu nah am Sollwert. Starten Sie die Feineinstellung.
0010 ⁴	Der Sollwert wurde während der Einstellung geändert. Hinweis: Sie können die zulässige Schwankung am Sollwert in der Variablen CancelTuningLevel festlegen.
0020	Während der Feineinstellung ist keine Erstopptimierung zulässig. Hinweis: Wenn vor dem Auftreten des Fehlers ActivateRecoverMode = TRUE war, bleibt PID_Compact im Feineinstellungsmodus.
0080 ⁴	Fehler während der Erstopptimierung. Inkorrekte Konfiguration der Grenzen des Ausgangswerts. Prüfen Sie, ob die Grenzen des Ausgangswerts richtig konfiguriert sind, und passen Sie die Steuerungslogik an.
0100 ⁴	Fehler während der Feineinstellung führte zu ungültigen Parametern.
0200 ^{2,3}	Ungültiger Wert am Parameter Input: Wert hat ungültiges Zahlenformat.
0400 ^{2,3}	Berechnung des Ausgangswerts fehlgeschlagen. Prüfen Sie die PID-Parameter.
0800 ^{1,2}	Abtastzeitfehler: PID_Compact wird nicht während der Abtastzeit des Weckalarm-OBS aufgerufen.
1000 ^{2,3}	Ungültiger Wert am Parameter Setpoint: Wert hat ungültiges Zahlenformat.
10000	Ungültiger Wert am Parameter ManualValue: Wert hat ungültiges Zahlenformat. Hinweis: Wenn vor dem Auftreten des Fehlers ActivateRecoverMode = TRUE war, verwendet PID_Compact SubstituteOutput als Ausgangswert. Sobald Sie im Parameter ManualValue einen gültigen Wert zuweisen, verwendet PID_Compact diesen als Ausgangswert.

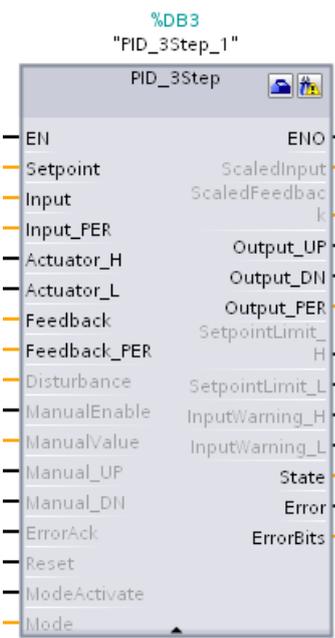
ErrorBit (DW#16#...)	Beschreibung
20000	<p>Ungültiger Wert der Variablen SubstituteValue: Wert hat ungültiges Zahlenformat.</p> <p>PID_Compact verwendet die untere Grenze des Ausgangswerts als Ausgangswert.</p> <p>Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war, ActivateRecoverMode = TRUE ist und der Fehler nicht mehr ansteht, gehtPID_Compact wieder in den Automatikbetrieb.</p>
40000	<p>Ungültiger Wert am Parameter Disturbance: Wert hat ungültiges Zahlenformat.</p> <p>Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv und ActivateRecoverMode = FALSE war, wird Disturbance auf null gesetzt. PID_Compact bleibt im Automatikbetrieb.</p> <p>Hinweis: Wenn vor dem Auftreten des Fehlers die Ersoptimierung oder die Feineinstellung aktiv und ActivateRecoverMode = TRUE war, wechselt PID_Compact in die im Parameter Mode gespeicherte Betriebsart. Wenn Störungen in der aktuellen Phase keine Auswirkungen auf den Ausgangswert haben, wird die Einstellung nicht abgebrochen.</p>

- 1 Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv und ActivateRecoverMode = TRUE war, bleibt PID_Compact im Automatikbetrieb.
- 2 Hinweis: Wenn vor dem Auftreten des Fehlers die Ersoptimierung oder die Feineinstellung aktiv war und ActivateRecoverMode = TRUE ist, wechselt PID_Compact in die im Parameter Mode gespeicherte Betriebsart.
- 3 Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv war und ActivateRecoverMode = TRUE ist, gibt PID_Compact den konfigurierten Ersatzausgangswert aus. Sobald der Fehler nicht mehr ansteht, wechselt PID_Compact wieder in den Automatikbetrieb.
- 4 Hinweis: Wenn vor dem Auftreten des Fehlers ActivateRecoverMode = TRUE war, bricht PID_Compact die Einstellung ab und wechselt in die im Parameter Mode gespeicherte Betriebsart.

8.4 Anweisung PID_3Step

Die Anweisung PID_3Step konfiguriert einen PID-Regler mit Selbsteinstellungsfunktionen, der für motorbetätigte Ventile und Stellglieder optimiert wurde.

Tabelle 8- 6 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>

- STEP 7 erstellt das Technologieobjekt und den Instanz-DB automatisch, wenn Sie die Anweisung einfügen. Der Instanz-DB enthält die Parameter des Technologieobjekts.
- Im SCL-Beispiel ist "PID_3Step_1" der Name des Instanz-DBs.

Tabelle 8-7 Datentypen für die Parameter

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Setpoint	IN	Real	Sollwert des PID-Reglers im Automatikbetrieb. (Standardwert: 0.0)
Input	IN	Real	Eine Variable des Anwenderprogramms wird als Quelle des Prozesswerts verwendet. (Standardwert: 0.0) Wenn Sie den Parameter Input verwenden, müssen Sie Config.InputPerOn = FALSE einstellen.
Input_PER	IN	Word	Ein Analogeingang wird als Quelle des Prozesswerts verwendet. (Standardwert: W#16#0) Wenn Sie den Parameter Input_PER verwenden, müssen Sie Config.InputPerOn = TRUE einstellen.
Actuator_H	IN	Bool	Digitale Positionsrückmeldung des Ventils für den oberen Endpunkt Wenn Actuator_H = TRUE ist, befindet sich das Ventil am oberen Endpunkt und wird nicht weiter in diese Richtung bewegt. (Standardwert: FALSE)
Actuator_L	IN	Bool	Digitale Positionsrückmeldung des Ventils für den unteren Endpunkt Wenn Actuator_L = TRUE ist, befindet sich das Ventil am unteren Endpunkt und wird nicht weiter in diese Richtung bewegt. (Standardwert: FALSE)
Feedback	IN	Real	Zurückgemeldete Position des Ventils. (Standardwert: 0.0) Wenn Sie den Parameter Feedback verwenden, müssen Sie Config.FeedbackPerOn = FALSE einstellen.
Feedback_PER	IN	Int	Zurückgemeldeter Analogwert der Ventilposition. (Standardwert: W#16#0) Wenn Sie den Parameter Feedback_PER verwenden, müssen Sie Config.FeedbackPerOn = TRUE. einstellen. Feedback_PER wird basierend auf den folgenden Variablen skaliert: <ul style="list-style-type: none"> • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Disturbance	IN	Real	Störgröße oder Vorsteuerungswert
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. (Standardwert: FALSE): <ul style="list-style-type: none"> • Eine Flanke von FALSE nach TRUE aktiviert den "Handbetrieb", während sich bei Zustand = 4 die Betriebsart nicht verändert. Solange ManualEnable = TRUE ist, können Sie die Betriebsart nicht mit einer steigenden Flanke an ModeActivate wechseln und auch nicht das Dialogfeld für die Inbetriebnahme verwenden. <ul style="list-style-type: none"> • Eine Flanke von TRUE nach FALSE aktiviert die von Mode zugewiesene Betriebsart. Hinweis: Es wird empfohlen, die Betriebsart lediglich über ModeActivate zu ändern.
ManualValue	IN	Real	Prozesswert für Handbetrieb. (Standardwert: 0.0) Im Handbetrieb geben Sie die absolute Position des Ventils an. ManualValue wird nur ausgewertet, wenn Sie OutputPer verwenden oder wenn eine zurückgemeldete Position verfügbar ist.

Parameter und Datentyp		Datentyp	Beschreibung
ManualUP	IN	Bool	<ul style="list-style-type: none"> • Manual_UP = TRUE: <ul style="list-style-type: none"> – Das Ventil wird geöffnet, auch wenn Sie Output_PER oder eine Positionsrückmeldung verwenden. Das Ventil wird nicht weiter bewegt, wenn der obere Endpunkt erreicht ist. – Siehe auch Config.VirtualActuatorLimit • Manual_UP = FALSE: <ul style="list-style-type: none"> – Wenn Sie Output_PER oder eine Positionsrückmeldung verwenden, wird das Ventil auf ManualValue bewegt. Andernfalls wird das Ventil nicht weiter bewegt. <p>Hinweis: Wenn Manual_UP und Manual_DN gleichzeitig auf TRUE gesetzt werden, wird das Ventil nicht bewegt.</p>
ManualDN	IN	Bool	<ul style="list-style-type: none"> • Manual_DN = TRUE: <ul style="list-style-type: none"> – Das Ventil wird geöffnet, auch wenn Sie Output_PER oder eine Positionsrückmeldung verwenden. Das Ventil wird nicht weiter bewegt, wenn der obere Endpunkt erreicht ist. – Siehe auch Config.VirtualActuatorLimit • Manual_DN = FALSE: <ul style="list-style-type: none"> – Wenn Sie Output_PER oder eine Positionsrückmeldung verwenden, wird das Ventil auf ManualValue bewegt. Andernfalls wird das Ventil nicht weiter bewegt.
ErrorAck	IN	Bool	Setzt die ErrorBits und Warnausgänge zurück. Flanke von FALSE nach TRUE
Reset	IN	Bool	<p>Startet den Regler neu. (Standardwert: FALSE):</p> <ul style="list-style-type: none"> • Flanke von FALSE nach TRUE: <ul style="list-style-type: none"> – Wechselt in den "inaktiven" Modus – Setzt die ErrorBits und Warnausgänge zurück – Löscht den Integralanteil – Speichert die PID-Parameter • Solange Reset = TRUE ist, bleibt PID_3Step im "inaktiven" Modus (Zustand = 0). • Flanke von TRUE nach FALSE: <ul style="list-style-type: none"> – PID_3Step wechselt in die Betriebsart, die im Parameter Mode gespeichert ist.
ModeActivate	IN	Bool	PID_3Step schaltet in den Betriebszustand, der im Parameter Mode gespeichert ist. Flanke von FALSE nach TRUE:
Mode	IN	Int	Die gewünschte PID-Betriebsart; wird bei einer steigenden Flanke am Eingang Mode Activate aktiviert.
ScaledInput	OUT	Real	Skalierter Prozesswert
ScaledFeedback	OUT	Real	<p>Skalierte Positionsrückmeldung des Ventils</p> <p>Hinweis: Für einen Aktor ohne Positionsrückmeldung ist die mit ScaledFeedback angegebene Stellung des Aktors sehr ungenau. ScaledFeedback ist in diesem Fall nur für eine grobe Schätzung der aktuellen Position verwendbar.</p>

Parameter und Datentyp		Datentyp	Beschreibung
Output_UP	OUT	Bool	Digitaler Ausgangswert zum Öffnen des Ventils. (Standardwert: FALSE) Wenn Config.OutputPerOn = FALSE, dann wird der Parameter Output_UP verwendet.
Output_DN	OUT	Bool	Digitaler Ausgangswert zum Schließen des Ventils. (Standardwert: FALSE) Wenn Config.OutputPerOn = FALSE, dann wird der Parameter Output_DN verwendet.
Output_PER	OUT	Word	Analoger Ausgangswert. Wenn Config.OutputPerOn = TRUE, dann wird der Parameter Output_PER verwendet.
SetpointLimitH	OUT	Bool	Sollwert oberer Grenzwert. (Standardwert: FALSE) Wenn SetpointLimitH = TRUE, dann ist der absolute obere Grenzwert des Sollwerts erreicht (Setpoint \geq Config.SetpointUpperLimit). Hinweis: Der Sollwert ist auf (Setpoint \geq Config.SetpointUpperLimit) begrenzt.
SetpointLimitL	OUT	Bool	Sollwert unterer Grenzwert. (Standardwert: FALSE) Wenn SetpointLimitL = TRUE, dann ist der absolute untere Grenzwert des Sollwerts erreicht (Setpoint \geq Config.SetpointLowerLimit). Hinweis: Der Sollwert ist auf (Setpoint \geq Config.SetpointLowerLimit) begrenzt.
InputWarningH	OUT	Bool	Wenn InputWarningH = TRUE, hat der Eingangswert die obere Warngrenze erreicht oder überschritten. (Standardwert: FALSE)
InputWarningL	OUT	Bool	Wenn InputWarningL = TRUE, hat der Eingangswert die untere Warngrenze erreicht oder überschritten. (Standardwert: FALSE)
State	OUT	Int	Aktuelle Betriebsart des PID-Reglers. (Standardwert: 0) Die Betriebsart können Sie mit dem Eingangsparameter Mode und einer steigenden Flanke an ModeActivate ändern: <ul style="list-style-type: none"> • State = 0: Inaktiv • State = 1: Erstopptimierung • State = 2: Manuelle Feineinstellung • State = 3: Automatikbetrieb • State = 4: Handbetrieb • State = 5: Ersatzausgangswert • State = 6: Übergangszeitmessung • State = 7: Fehlerüberwachung • State = 8: Ersatzausgangswert mit Fehlerüberwachung • State = 10: Handbetrieb ohne Endpunktsignale

Parameter und Datentyp	Datentyp	Beschreibung
Error	OUT	Bool
ErrorBits	OUT	DWord

Wenn Error = TRUE, steht mindestens eine Fehlermeldung an. (Standardwert: FALSE)
Hinweis: Der Parameter Error in V1.x PID war das Feld der ErrorBits mit den Fehlercodes. Jetzt zeigt ein Boolescher Merker an, dass ein Fehler aufgetreten ist.

In der Parametertabelle (Seite 202) der ErrorBits für die Anweisung PID_3Step werden die anstehenden Fehlermeldungen beschrieben. (Standardwert: DW#16#0000 (kein Fehler)). ErrorBits ist remanent und wird bei einer steigenden Flanke an Reset oder ErrorAck zurückgesetzt.
Hinweis: In V1.x war der Parameter ErrorBits als Parameter Error definiert und nicht vorhanden.

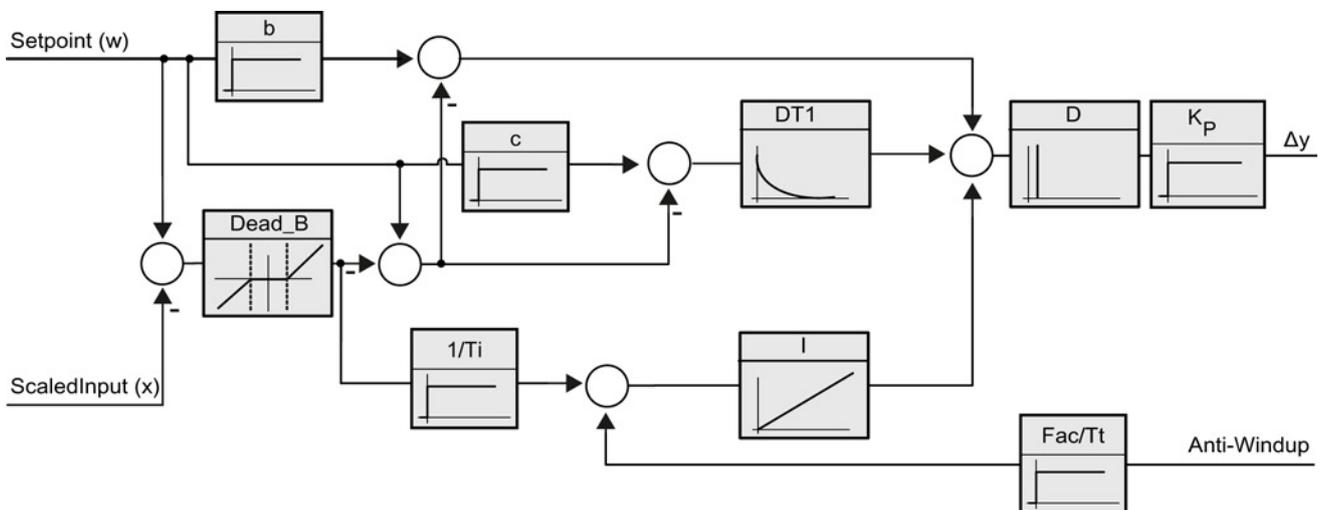


Bild 8-3 Funktionsweise des Reglers PID_3Step als PID T1-Regler mit Anti-Windup

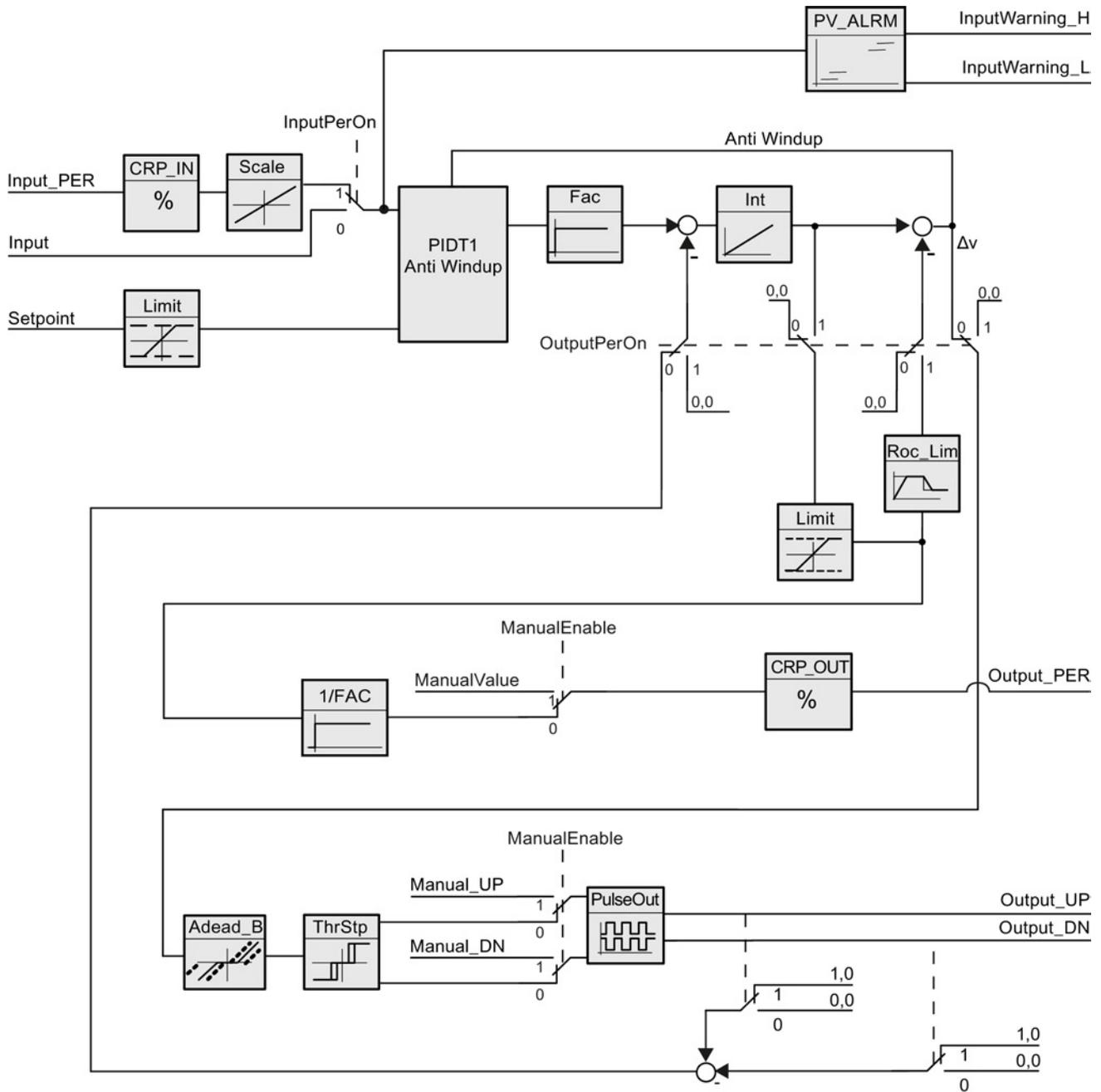


Bild 8-4 Funktionsweise des Reglers PID_3Step ohne Positionsrückmeldung

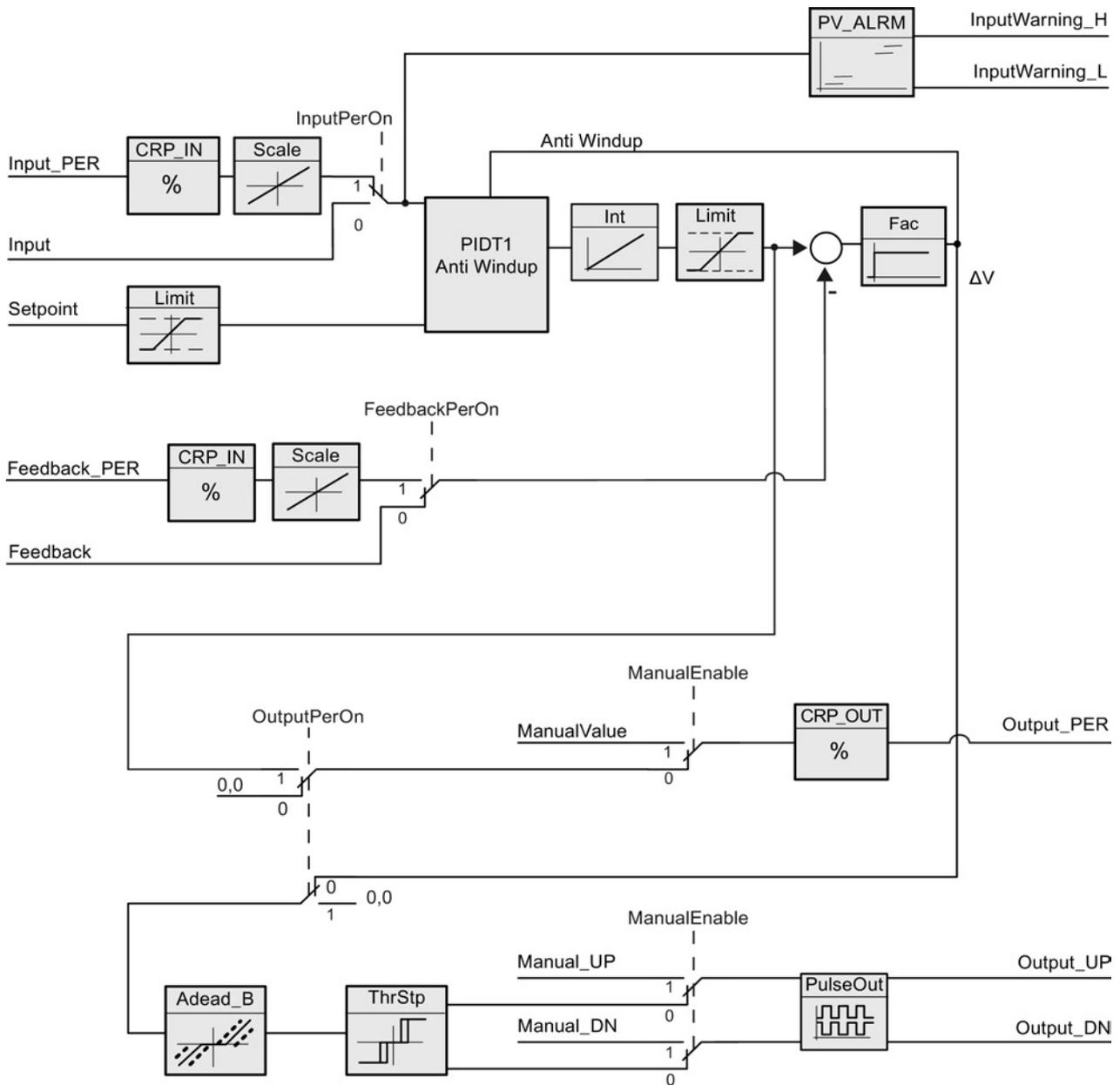


Bild 8-5 Funktionsweise des Reglers PID_3Step mit aktivierter Positionsrückmeldung

8.5 ErrorBit-Parameter der Anweisung PID_3Step

Stehen mehrere Fehler an, werden die Werte der Fehlercodes mittels binärer Addition angezeigt. Die Anzeige von Fehlercode 0003 beispielsweise weist darauf hin, dass auch die Fehler 0001 und 0002 anstehen.

Tabelle 8- 8 ErrorBit-Parameter der Anweisung PID_3STEP

ErrorBit (DW#16#...)	Beschreibung
0000	Kein Fehler
0001 ^{1, 2}	Der Parameter Input liegt außerhalb der Prozesswertgrenzen. Input > Config.InputUpperLimit Input < Config.InputLowerLimit
0002 ^{2, 3}	Ungültiger Wert am Parameter Input_PER. Prüfen Sie, ob am Analogeingang ein Fehler anliegt.
0004 ⁴	Fehler während der Feineinstellung. Schwingung des Prozesswerts konnte nicht erhalten werden.
0010 ⁴	Der Sollwert wurde während der Einstellung geändert. Hinweis: Sie können die zulässige Schwankung am Sollwert in der Variablen CancelTuningLevel festlegen.
0020	Während der Feineinstellung ist keine Erstopptimierung zulässig. Hinweis: Wenn vor dem Auftreten des Fehlers ActivateRecoverMode = TRUE war, bleibt PID_3Step im Feineinstellungsmodus.
0080 ⁴	Fehler während der Erstopptimierung. Inkorrekte Konfiguration der Grenzen des Ausgangswerts. Prüfen Sie, ob die Grenzen des Ausgangswerts richtig konfiguriert sind, und passen Sie die Steuerungslogik an.
0100 ⁴	Fehler während der Feineinstellung führte zu ungültigen Parametern.
0200 ^{2, 3}	Ungültiger Wert am Parameter Input: Wert hat ungültiges Zahlenformat.
0400 ^{2, 3}	Berechnung des Ausgangswerts schlug fehl. Prüfen Sie die PID-Parameter.
0800 ^{1, 2}	Abtastzeitfehler: PID_3Step wird nicht während der Abtastzeit des Weckalarm-OBs aufgerufen.
1000 ^{2, 3}	Ungültiger Wert am Parameter Setpoint: Wert hat ungültiges Zahlenformat.
2000 ^{1, 2, 5}	Ungültiger Wert am Parameter Feedback_PER. Prüfen Sie, ob am Analogeingang ein Fehler ansteht.
4000 ^{1, 2, 5}	Ungültiger Wert am Parameter Feedback: Wert hat ungültiges Zahlenformat.
8000 ^{1, 2}	Fehler bei der digitalen Positionsrückmeldung. Actuator_H = TRUE und Actuator_L = TRUE. Das Stellglied kann nicht auf den Ersatzausgangswert wechseln und bleibt in seiner aktuellen Position. In diesem Zustand ist kein Handbetrieb möglich. Um das Stellglied aus diesem Zustand zu bewegen, müssen Sie den "Endpunkt des Stellglieds" deaktivieren (Config.ActuatorEndStopOn = FALSE) oder in den Handbetrieb ohne Endpunktsignale wechseln (Mode = 10).

ErrorBit (DW#16#...)	Beschreibung
10000	<p>Ungültiger Wert am Parameter ManualValue: Wert hat ungültiges Zahlenformat.</p> <p>Das Stellglied kann nicht auf den manuellen Wert wechseln und bleibt in seiner aktuellen Position.</p> <p>Weisen Sie in ManualValue einen gültigen Wert zu oder bewegen Sie das Stellglied im Handbetrieb mit Manual_UP und Manual_DN.</p>
20000	<p>Ungültiger Wert der Variablen SavePosition: Wert hat ungültiges Zahlenformat.</p> <p>Das Stellglied kann nicht auf den Ersatzausgangswert wechseln und bleibt in seiner aktuellen Position.</p>
40000	<p>Ungültiger Wert am Parameter Disturbance: Wert hat ungültiges Zahlenformat.</p> <p>Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv und ActivateRecoverMode = FALSE war, wird Disturbance auf null gesetzt. PID_3Step bleibt im Automatikbetrieb.</p> <p>Hinweis: Wenn vor dem Auftreten des Fehlers die Erstopptimierung oder die Feineinstellung aktiv und ActivateRecoverMode = TRUE war, wechselt PID_3Step in die im Parameter Mode gespeicherte Betriebsart. Wenn Störungen in der aktuellen Phase keine Auswirkungen auf den Ausgangswert haben, wird die Einstellung nicht abgebrochen.</p> <p>Der Fehler wirkt sich nicht während der Übergangszeitmessung aus.</p>

- ¹ Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv und ActivateRecoverMode = TRUE war, bleibt PID_3Step im Automatikbetrieb.
- ² Hinweis: Wenn vor dem Auftreten des Fehlers die Erstopptimierung, die Feineinstellung oder die Übergangszeitmessung aktiv und ActivateRecoverMode = TRUE war, wechselt PID_3Step in die im Parameter Mode gespeicherte Betriebsart.
- ³ Hinweis: Wenn vor dem Auftreten des Fehlers der Automatikbetrieb aktiv und ActivateRecoverMode = TRUE war, wechselt PID_3Step in die Betriebsart "Ersatzausgangswert mit Fehlerüberwachung" oder "Fehlerüberwachung". Sobald der Fehler nicht mehr ansteht, wechselt PID_3Step wieder in den Automatikbetrieb.
- ⁴ Hinweis: Wenn vor dem Auftreten des Fehlers ActivateRecoverMode = TRUE war, bricht PID_3Step die Einstellung ab und wechselt in die im Parameter Mode gespeicherte Betriebsart.
- ⁵ Das Stellglied kann nicht auf den Ersatzausgangswert wechseln und bleibt in seiner aktuellen Position. Im Handbetrieb können Sie die Position des Stellglieds nur mit Manual_UP und Manual_DN und nicht mit ManualValue ändern.

8.6 PID-Regler konfigurieren

Die Parameter des Technologieobjekts legen die Funktionsweise des PID-Reglers fest. Öffnen Sie den Konfigurationseditor über das Symbol.

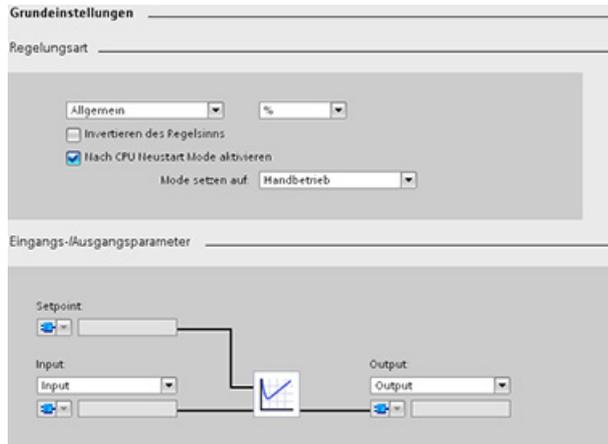


Tabelle 8-9 Beispielhafte Konfigurationseinstellungen für die Anweisung PID_Compact

Einstellungen		Beschreibung
Grundlagen	Reglertyp	Wählt die physikalischen Einheiten aus.
	Reglerlogik invertieren	Ermöglicht die Auswahl eines invers funktionierenden PID-Reglers. <ul style="list-style-type: none"> Ist diese Option nicht ausgewählt, verhält sich der PID-Regler entsprechend der direkten Funktionsweise und der Ausgang des PID-Regelkreises steigt, wenn der Eingangswert < Sollwert ist. Ist diese Option ausgewählt, steigt der Ausgang des PID-Regelkreises, wenn der Eingangswert > Sollwert ist.
	Nach CPU-Neustart letzte Betriebsart aktivieren	Startet den PID-Regler neu, nachdem er zurückgesetzt wurde oder wenn eine Eingangsgrenze überschritten und in den gültigen Bereich zurückgeführt wurde.
	Eingang	Wählt entweder den Parameter Input oder den Parameter Input_PER (bei analog) für den Prozesswert aus. Input_PER kann direkt von einem analogen Eingangsmodul kommen.
	Ausgang	Wählt entweder den Parameter Output oder den Parameter Output_PER (bei analog) für den Ausgangswert aus. Output_PER kann direkt an ein analoges Ausgangsmodul gehen.
Prozesswert	Skaliert sowohl den Bereich als auch die Grenzen für den Prozesswert. Wenn der Prozesswert unter die untere Grenze fällt oder über die obere Grenze steigt, wechselt der PID-Regelkreis in die inaktive Betriebsart und setzt den Ausgangswert auf 0. Um Input_PER zu verwenden, müssen Sie den analogen Prozesswert (Eingangswert) skalieren.	

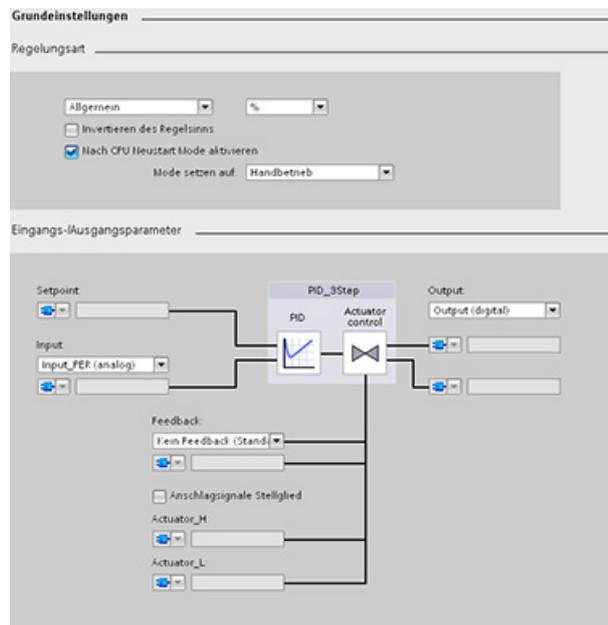


Tabelle 8- 10 Beispielhafte Konfigurationseinstellungen für die Anweisung PID_3Step

Einstellungen		Beschreibung
Grundlagen	Reglertyp	Wählt die physikalischen Einheiten aus.
	Reglerlogik invertieren	Ermöglicht die Auswahl eines invers funktionierenden PID-Reglers. <ul style="list-style-type: none"> Ist diese Option nicht ausgewählt, verhält sich der PID-Regler entsprechend der direkten Funktionsweise und der Ausgang des PID-Regelkreises steigt, wenn der Eingangswert < Sollwert ist. Ist diese Option ausgewählt, steigt der Ausgang des PID-Regelkreises, wenn der Eingangswert > Sollwert ist.
	Betriebsart nach CPU-Neustart aktivieren	Startet den PID-Regler neu, nachdem er zurückgesetzt wurde oder wenn eine Eingangsgrenze überschritten und in den gültigen Bereich zurückgeführt wurde. Betriebsart einstellen: Definiert die Betriebsart, die der PID-Regler nach dem Neustart einnehmen soll.
	Eingang	Wählt entweder den Parameter Input oder den Parameter Input_PER (bei analog) für den Prozesswert aus. Input_PER kann direkt von einem analogen Eingangsmodul kommen.
	Ausgang	Gibt an, ob die digitalen Ausgänge (Output_UP und Output_DN) oder der analoge Ausgang (Output_PER) für den Ausgangswert verwendet werden soll.
	Rückmeldung	Gibt die Art des an den PID-Regler zurückgemeldeten Gerätezustands an: <ul style="list-style-type: none"> Keine Rückmeldung (Standard) Rückmeldung Feedback_PER
Prozesswert	Skaliert sowohl den Bereich als auch die Grenzen für den Prozesswert. Wenn der Prozesswert unter die untere Grenze fällt oder über die obere Grenze steigt, wechselt der PID-Regelkreis in die inaktive Betriebsart und setzt den Ausgangswert auf 0. Um Input_PER zu verwenden, müssen Sie den analogen Prozesswert (Eingangswert) skalieren.	

Einstellungen		Beschreibung
Stellglied	Motorübergangszeit	Legt die Zeit vom offenen zum geschlossenen Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Kleinste EIN-Zeit	Legt die minimale Betätigungszeit für das Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Kleinste AUS-Zeit	Legt die minimale Pausenzeit für das Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Reaktion auf den Fehler	Definiert das Verhalten des Ventils, wenn ein Fehler erkannt oder der PID-Regler zurückgesetzt wird. Wenn Sie festlegen, dass eine Ersatzposition verwendet werden soll, geben Sie die "Sicherheitsposition" ein. Bei analoger Rückmeldung oder analogem Ausgang wählen Sie einen Wert zwischen der unteren und oberen Grenze für den Ausgang. Bei Digitalausgängen können Sie nur 0 % (aus) oder 100 % (ein) wählen.
	Positionsrückmeldung skalieren ¹	<ul style="list-style-type: none"> • "Oberer Endpunkt" und "Unterer Endpunkt" definieren die maximale positive Position (vollständig geöffnet) und die maximale negative Position (vollständig geschlossen). "Oberer Endpunkt" muss größer als "Unterer Endpunkt" sein. • "Prozesswert obere Grenze" und "Prozesswert untere Grenze" definieren die untere und obere Position des Ventils während der Einstellung und im Automatikbetrieb. • "FeedbackPER" ("Low" und "High") definieren die analoge Rückmeldung der Ventilposition. "FeedbackPER High" muss höher als "FeedbackPER Low" sein.
Erweitert	Prozesswertüberwachung	Legt den oberen und unteren Grenzwert für den Prozesswert fest.
	PID-Parameter	In diesem Fenster kann der Anwender bei Bedarf seine eigenen PID-Einstellungsparameter eingeben. Hierfür muss das Kontrollkästchen "Manuelle Eingabe aktivieren" aktiviert sein.

¹ "Positionsrückmeldung skalieren" kann nur geändert werden, wenn Sie in den grundlegenden Einstellungen "Rückmeldung" aktiviert haben.

8.7 Inbetriebnahme des PID-Reglers

Sie konfigurieren den PID-Regler im Inbetriebnahme-Editor für die Selbsteinstellung beim Anlauf und für die Selbsteinstellung während des Betriebs. Um den Inbetriebnahme-Editor aufzurufen, klicken Sie im Anweisungsverzeichnis oder in der Projektnavigation auf das entsprechende Symbol.



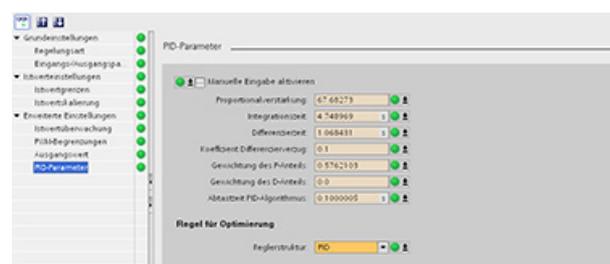
Tabelle 8- 11 Beispiel für einen Inbetriebnahmebildschirm (PID_3Step)

	<ul style="list-style-type: none"> • Messung: Um den Sollwert, den Prozesswert (Eingangswert) und den Ausgangswert in einer Echtzeitkurve anzuzeigen, geben Sie die Abtastzeit ein und klicken auf die Schaltfläche "Start". • Optimierungsart: Um den PID-Regler einzustellen, wählen Sie entweder "Erstoptimierung" oder "Feineinstellung" (manuell) aus und klicken auf die Schaltfläche "Start". Der PID-Regler durchläuft mehrere Phasen, um die Zeiten für Systemantwort und Aktualisierung zu berechnen. Anhand dieser Werte werden die entsprechenden Optimierungsparameter berechnet. <p>Nach Abschluss des Einstellungsprozesses können Sie die neuen Parameter speichern, indem Sie im Inbetriebnahme-Editor im Bereich "PID-Parameter" auf die Schaltfläche "PID-Parameter laden" klicken.</p> <p>Tritt während der Einstellung ein Fehler auf, geht der Ausgangswert des PID-Reglers auf 0. Der PID-Regler wird in die "inaktive" Betriebsart versetzt. Der Status zeigt den Fehler an.</p>
--	--

PID-Startwertsteuerung

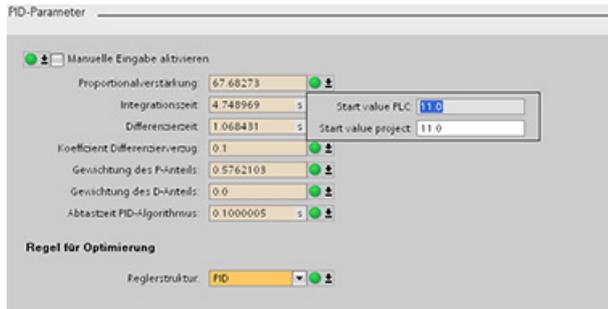
Sie können die Istwerte der PID-Konfigurationsparameter ändern, so dass das Verhalten des PID-Reglers im Online-Modus optimiert werden kann.

Öffnen Sie die Technologieobjekte für Ihren PID-Regler und dessen Konfigurationsobjekt. Um die Startwertsteuerung aufzurufen, klicken Sie auf das Brillensymbol in der linken oberen Ecke des Dialogfelds:



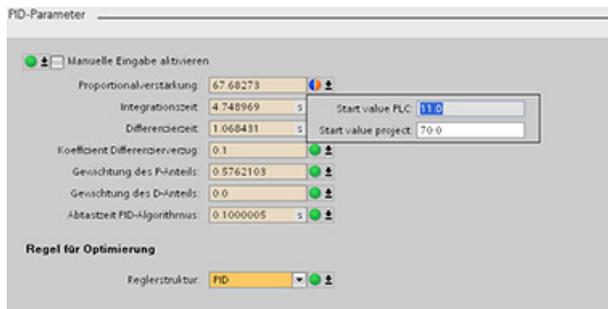
Nun können Sie die Werte beliebiger Konfigurationsparameter Ihres PID-Reglers wie in nachstehender Abbildung gezeigt ändern.

Sie können für jeden Parameter den Istwert mit dem Startwert des Projekts (offline) und dem Startwert des PLCs (online) vergleichen. Dies ist erforderlich, um die Online-/Offline-Unterschiede des Technologieobjekt-Datenbausteins (TO-DB) zu vergleichen und die Werte zu kennen, die beim nächsten Wechsel von STOP nach START des PLCs als aktuelle Werte verwendet werden. Außerdem erhalten Sie durch ein Vergleichssymbol einen visuellen Hinweis, um Online-/Offline-Unterschiede schnell zu erkennen:



Die obige Abbildung zeigt den Bildschirm der PID-Parameter mit Vergleichssymbolen. Die Symbole weisen darauf hin, welche Werte sich zwischen Online- und Offline-Projekt unterscheiden. Ein grünes Symbol kennzeichnet die Werte, die gleich sind. Ein blau-orangefarbenes Symbol zeigt an, dass die Werte voneinander abweichen.

Außerdem können Sie auf die Parameterschaltfläche mit dem Abwärtspfeil klicken, um ein kleines Fenster anzuzeigen, in dem für jeden Parameter der Startwert des Projekts (offline) und der Startwert des PLCs (online) angezeigt werden:



Webserver für einfachen Internetanschluss

Der Webserver bietet Webseitenzugriff auf Daten über Ihre CPU und auf Prozessdaten in der CPU. Mithilfe der Webseiten können Sie über den Webbrowser Ihres PCs oder Mobilgeräts auf die CPU zugreifen. Die Standard-Webseiten ermöglichen berechtigten Anwendern die Ausführung einer Vielzahl von Funktionen:

- Sie können den Betriebszustand (RUN oder STOP) der CPU ändern.
- Sie können den Status der PLC-Variablen überwachen und ändern.
- Sie können von der CPU erfasste Datenprotokolle anzeigen und herunterladen.
- Sie können den Diagnosepuffer der CPU anzeigen.
- Sie können die Firmware der CPU aktualisieren.

Sie können für den Webserver auch benutzerdefinierte Webseiten erstellen, über die Sie auf CPU-Daten zugreifen können. Sie können diese Seiten mit einer HTML-Authoring-Software Ihrer Wahl anlegen. Sie fügen vordefinierte "AWP"-Befehle (Automation Web Programming) in Ihren HTML-Code ein, um auf die Daten in der CPU zuzugreifen.

Benutzer und Berechtigungsstufen für den Webserver richten Sie in der Gerätekonfiguration der CPU in STEP 7 ein.

Voraussetzungen für den Webbrowser

Der Webserver unterstützt die folgenden PC-Webbrowser:

- Internet Explorer 8.0 oder höher
- Mozilla Firefox 3.0 oder höher
- Opera 11.0 oder höher
- Google Chrome 21.0 oder höher
- Apple Safari 6.0 oder höher

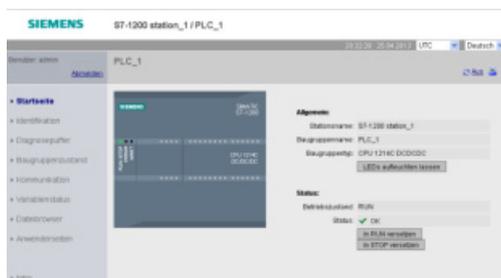
Der Webserver unterstützt die folgenden Webbrowser auf Mobilgeräten:

- Internet Explorer 8.0 oder höher für HMI-Panels
- Mobile Safari, iOS 5.0 oder höher
- Mobile Android Browser 2.3.4 oder höher
- Mobile Google Chrome, Android 4.0 oder höher

Informationen zu browserspezifischen Einschränkungen, die sich auf die Anzeige von Standard- oder benutzerdefinierten Webseiten auswirken können, finden Sie in den Themen zu Einschränkungen (Seite 213).

9.1 Einfache Nutzung der Standard-Webseiten

Die Nutzung der Standard-Webseiten ist äußerst einfach. Sie müssen bei der Konfiguration der CPU lediglich den Webserver aktivieren.

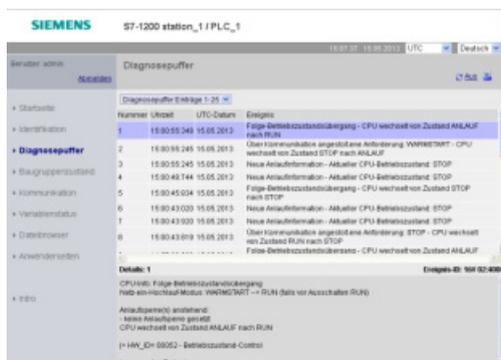


Die Startseite zeigt eine Darstellung der CPU an, mit der Sie verbunden sind, und führt allgemeine Informationen über die CPU auf.

Wenn Sie sich als Benutzer mit den erforderlichen Rechten anmelden, können Sie den Betriebszustand der CPU (STOP und RUN) ändern und die LEDs blinken lassen.



Auf der Variablenzustandsseite können Sie alle E/A oder Speicherdaten in Ihrer CPU überwachen oder ändern. Zum Beobachten von Werten müssen Sie sich mit Datenleserechten, zum Ändern von Werten müssen Sie sich mit Datenschreiberechten anmelden. Sie können eine direkte Adresse (wie E0.0), einen PLC-Variablenamen oder eine Variable aus einem spezifischen Programmabstein eingeben. Die Datenwerte werden automatisch aktualisiert, sofern Sie diese automatische Aktualisierung nicht ausschalten.

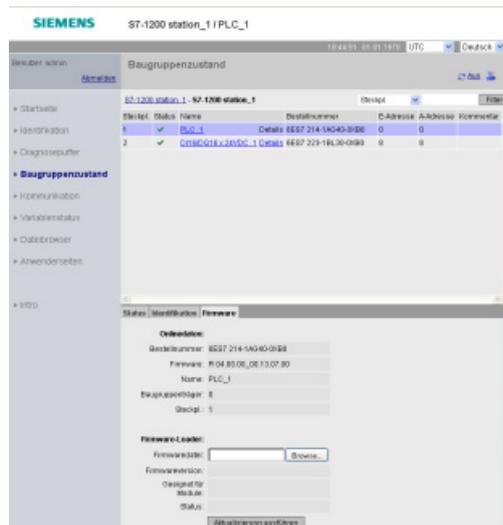


Die Diagnosepufferseite zeigt den Diagnosepuffer an und steht Benutzern mit Rechten für die Diagnoseabfrage zur Verfügung. Sie können den Bereich der anzuzeigenden Diagnoseeinträge auswählen.

Die Diagnoseeinträge führen die aufgetretenen Ereignisse sowie Uhrzeit und Datum der CPU auf, zu denen die Ereignisse aufgetreten sind. Wählen Sie ein einzelnes Ereignis aus, um ausführliche Informationen dazu anzuzeigen.



Auf der Dateibrowser-Seite können Sie Dateien im Ladespeicher der CPU wie z. B. Datenprotokolle (Seite 126) und Rezepte anzeigen, herunterladen oder bearbeiten. Sofern die CPU nicht mit Schutzstufe 4 geschützt ist, können alle Anwender die Dateien auf der Dateibrowser-Seite anzeigen. Anwender mit Rechten zum Ändern von Dateien können Dateien löschen, bearbeiten und umbenennen.



Auf der Modulinformationsseite können Sie nicht nur Informationen zu den Modulen in Ihrer Station anzeigen, sondern die Seite ermöglicht Ihnen auch die Aktualisierung der Firmwareversion in Ihrer CPU oder anderen Modulen, die Firmware-Updates unterstützen. Anwender mit Rechten für den Diagnoseabruf können Modulinformationen anzeigen. Anwender mit Rechten zur Ausführung eines Firmware-Updates können die Firmware aktualisieren.

Weitere Standard-Webseiten zeigen Informationen über die CPU (z. B. Seriennummer, Version und Bestellnummer) und die Kommunikationsparameter (z. B. Netzwerkadressen, physische Eigenschaften der Kommunikationsschnittstellen und Kommunikationsstatistik) an.



WARNUNG

Nicht berechtigter Zugriff auf die CPU über den Webserver

Nicht berechtigter Zugriff auf die CPU oder das Einstellen von ungültigen Werten für PLC-Variablen kann den Prozessbetrieb stören und zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen.

Weil durch die Aktivierung des Webserverns berechnigte Benutzer in der Lage sind, Betriebszustandsänderungen vorzunehmen, PLC-Daten zu schreiben und Firmware-Updates durchzuführen, empfiehlt Siemens, die folgenden Sicherheitsvorkehrungen einzuhalten:

- Aktivieren Sie den Zugriff auf den Webserver nur über das HTTPS-Protokoll.
- Schützen Sie die Benutzerkennungen für den Webserver durch ein starkes Passwort. Starke Passwörter sind mindestens zehn Zeichen lang, bestehen aus Buchstaben, Zahlen und Sonderzeichen, sind keine Wörter, die in einem Wörterbuch gefunden werden können, und sind keine Namen oder Kennungen, die sich aus persönlichen Daten ableiten lassen. Halten Sie das Passwort geheim und ändern Sie es häufig.
- Erweitern Sie keinesfalls die standardmäßigen Mindestrechte des Benutzers "Jeder".
- 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).

9.2 Bedingungen, die sich auf die Nutzung des Webserver auswirken können

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. 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" können Sie das Zertifikat herunterladen und in die Internetoptionen Ihres Webbrowsers importieren. 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 HTTP-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, weitere HTML-Dateien). Einige Verbindungen bleiben bestehen, während der Webserver eine Seite anzeigt, andere werden nach der Erstverbindung freigegeben.

Wenn Sie beispielsweise den Browser Mozilla Firefox 8 nutzen, der maximal sechs beständige Verbindungen unterstützt, 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.

Hinweis

Vor dem Schließen des Webserverns 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.

9.2.1 Funktionseinschränkungen bei Deaktivierung von JavaScript in den Internetoptionen

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 in Ihrem Webbrowser JavaScript nicht aktivieren können, können Funktionen mit JavaScript-Steuerung nicht ausgeführt werden.

Allgemein

Die Seiten werden nicht dynamisch aktualisiert. Sie müssen die Seiten manuell über das Symbol zum Aktualisieren aktualisieren, um die aktuellen Daten anzuzeigen.

Diagnosepufferseite

Die Diagnosepufferseite verwendet JavaScript wie folgt:

- Einzelheiten zu Ereignissen anzeigen: Mit JavaScript wählen Sie eine Zeile im Diagnosepuffer aus, um im unteren Abschnitt Einzelheiten dazu anzuzeigen. Ohne JavaScript müssen Sie auf den Hyperlink des Ereignisfelds eines Diagnosepuffereintrags klicken, um die Ereignisdaten im unteren Bereich anzuzeigen.
- Bereich der anzuzeigenden Diagnosepuffereinträge ändern: Mit JavaScript können Sie über die Klappliste oben auf der Seite den Bereich der anzuzeigenden Diagnosepuffereinträge auswählen, woraufhin die Seite automatisch aktualisiert wird. Ohne JavaScript können Sie über die Klappliste oben auf der Seite den Bereich der anzuzeigenden Diagnosepuffereinträge auswählen, doch Sie müssen dann auf den Link "Go" klicken, um die Diagnosepufferseite entsprechend zu aktualisieren.

Beachten Sie, dass der Hyperlink "Go" und die Hyperlinks der Ereignisfelder nur angezeigt werden, wenn JavaScript nicht aktiviert ist. Diese Links sind nicht erforderlich und deshalb auch nicht vorhanden, wenn JavaScript aktiviert ist.

Modulinformationsseite

Ohne aktiviertes JavaScript gelten die folgenden Einschränkungen:

- Sie können keine Daten filtern.
- Sie können keine Felder sortieren.

Variablenzustandsseite

Ohne aktiviertes JavaScript gelten die folgenden Einschränkungen:

- Nachdem Sie eine Variable eingegeben haben, müssen Sie den Fokus manuell in die Zeile "New variable" setzen, um eine neue Variable einzugeben.
- Wenn Sie ein Anzeigeformat auswählen, wird der Datenwert nicht automatisch im ausgewählten Format angezeigt. Sie müssen auf die Schaltfläche "Monitor value" klicken, um die Anzeige mit dem neuen Format zu aktualisieren.

9.2.2 Eingeschränkte Funktionen, wenn Cookies in den Internetoptionen nicht erlaubt sind

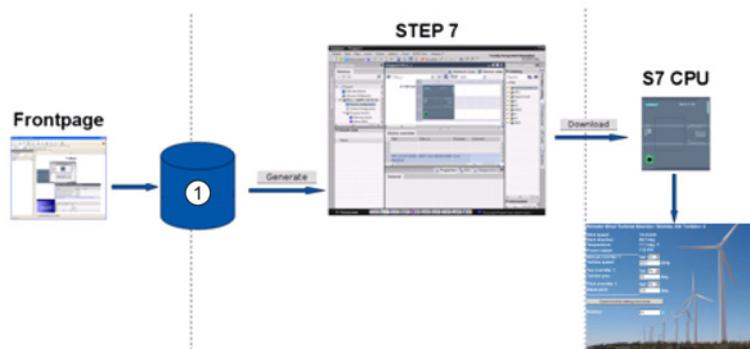
Wenn Cookies in Ihrem Webbrowser deaktiviert sind, gelten die folgenden Einschränkungen:

- 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.

9.3 Benutzerdefinierte Webseiten einfach anlegen

9.3.1 Eigene benutzerdefinierte Webseiten einfach anlegen

Für den S7-1200 Webserver können Sie auch eigene anwendungsspezifische HTML-Seiten mit Daten des Zielsystems anlegen. Sie erstellen diese Seiten mit einem HTML-Editor Ihrer Wahl und laden die Seiten in die CPU, von wo aus sie über die Standard-Webseiten aufrufbar sind.



① HTML-Dateien mit eingebetteten AWP-Befehlen

Hierfür sind mehrere Tätigkeiten durchzuführen:

- Erstellen Sie die HTML-Seiten mit einem HTML-Editor wie Microsoft Frontpage.
- Nehmen Sie AWP-Befehle in HTML-Kommentare im HTML-Code auf: Bei den AWP-Befehlen handelt es sich um einen fest vorgegebenen Satz Befehle für den Zugriff auf CPU-Informationen.
- Konfigurieren Sie STEP 7 zum Lesen und Verarbeiten von HTML-Seiten.
- Generieren Sie die Programmbausteine aus den HTML-Seiten.
- Programmieren Sie STEP 7 für die Steuerung der Verwendung der HTML-Seiten.
- Übersetzen Sie die Programmbausteine und laden Sie sie in die CPU.
- Greifen Sie über Ihren PC oder Ihr Mobilgerät auf die benutzerdefinierten Webseiten zu.

Um Ihre eigenen HTML-Seiten zur Verwendung mit dem Webserver zu erstellen, können Sie ein Softwarepaket Ihrer Wahl verwenden. Achten Sie darauf, dass Ihr HTML-Code mit den HTML-Standards des W3C (World Wide Web Consortium) konform ist. 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-Authentifizierungstools 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 für die DB-Fragmente belegen. Wenn der interne Ladespeicher Ihrer CPU für Ihre benutzerdefinierten Webseiten nicht ausreicht, stellen Sie über eine Memory Card externen Ladespeicher zur Verfügung.

Um Ihren HTML-Code so zu programmieren, dass Daten aus der S7-1200 verwendet werden, können Sie AWP-Befehle als HTML-Kommentare einfügen. Speichern Sie abschließend Ihre HTML-Seiten auf Ihrem PC und notieren Sie sich den Speicherpfad.

Hinweis

Die maximale Dateigröße für HTML-Dateien mit AWP-Befehlen beträgt 64 KB. Ihre Dateien dürfen diesen Grenzwert nicht überschreiten.

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.

9.3.2 Einschränkungen bei benutzerdefinierten Webseiten

Die Einschränkungen bei Standard-Webseiten 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.

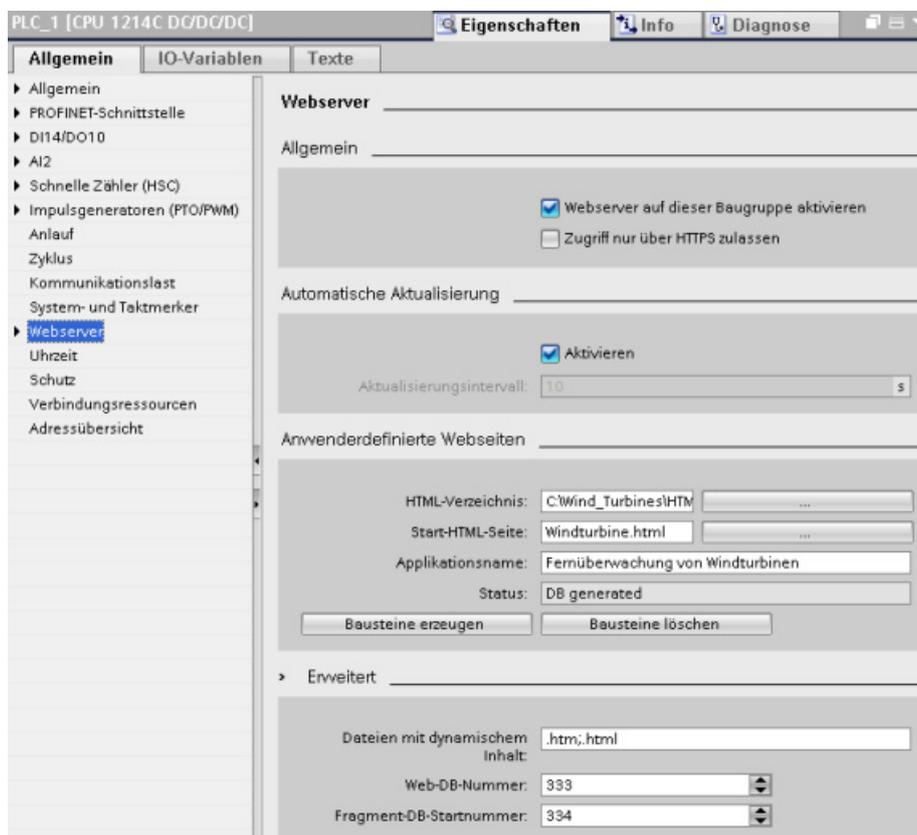
Mit den Online- und Diagnosefunktionen in STEP 7 können Sie den belegten und den freien Platz im Ladespeicher abfragen. Sie können zudem in den Eigenschaften der einzelnen Bausteine, die STEP 7 für Ihre benutzerdefinierten Webseiten generiert, den benötigten Platz im Ladespeicher prüfen.

Hinweis

Wenn Sie den Platz für Ihre benutzerdefinierten Webseiten verringern müssen, entfernen Sie ggf. einige der eingefügten Bilder.

9.3.3 Konfiguration einer benutzerdefinierten Webseite

Um die benutzerdefinierten Webseiten zu konfigurieren, bearbeiten Sie die Eigenschaften "Webserver" der CPU.



Nachdem Sie die Webserver-Funktionalität aktiviert haben, geben Sie die folgenden Informationen ein:

- Name und Speicherort der HTML-Standardstartseite für die Generierung der DBs für die benutzerdefinierten Webseiten.
- Name Ihrer Anwendung (optional). Der Anwendungsname dient zur weiteren Unterteilung bzw. Gruppierung der Webseiten. Wenn ein Anwendungsname vorhanden ist, wird die URL in folgendem Format angezeigt:
 http://ww.xx.yy.zz/awp/<Anwendungsname>/<Seitenname>.html
- Dateierweiterungen, die auf das Vorhandensein von AWP-Befehlen zu prüfen sind. Standardmäßig analysiert STEP 7 Dateien mit den Erweiterungen *.htm, *.html und *.js. Wenn Sie andere Dateierweiterungen nutzen, fügen Sie sie ein.
- Kennnummern des Steuer-DBs und des ersten DB-Fragments.

Nachdem Sie den Webserver konfiguriert haben, klicken Sie auf die Schaltfläche "Bausteine erstellen", um die DBs aus den HTML-Seiten zu generieren. Nachdem Sie die DBs generiert haben, sind Ihre Webseiten ein Teil Ihres Anwenderprogramms. Der Steuerdatenbaustein für den Betrieb Ihrer Webseiten und die DB-Fragmente enthalten alle HTML-Seiten.

9.3.4 WWW-Anweisung verwenden

Die WWW-Anweisung ermöglicht, dass die benutzerdefinierten Webseiten über die Standard-Webseiten aufrufbar sind. Ihr Anwenderprogramm muss die Anweisung WWW nur einmal ausführen, um den Zugriff auf benutzerdefinierte Webseiten zu ermöglichen. Sie können jedoch auch festlegen, dass benutzerdefinierte Webseiten nur unter bestimmten Bedingungen verfügbar sind. Ihr Anwenderprogramm kann die Anweisung WWW dann entsprechend den Anforderungen Ihrer Anwendung aufrufen.

Tabelle 9- 1 Anweisung WWW

KOP/FUP	SCL	Beschreibung
	<pre>ret_val := #WWW(ctrl_db:=_uint_in_);</pre>	<p>Gibt den Steuer-DB für die benutzerdefinierten Webseiten an.</p> <p>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.</p>

Ihr Anwenderprogramm verwendet den Steuer-DB normalerweise so, wie er über den Vorgang "Bausteine erstellen" angelegt wurde, ohne weitere Änderungen. Das Anwenderprogramm kann jedoch globale Befehle in den Steuerdatenbaustein einfügen, die den Webserver deaktivieren oder anschließend erneut aktivieren. Außerdem muss das Anwenderprogramm das Verhalten von benutzerdefinierten Webseiten, die Sie als manuelle DB-Fragmente anlegen, über eine Anforderungstabelle im Steuer-DB steuern.

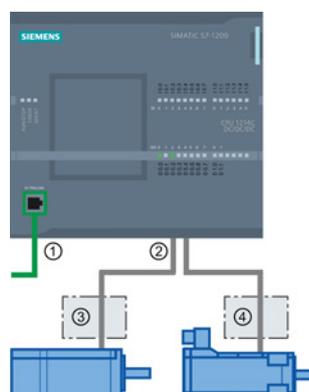
Einfache Bewegungssteuerung

Die CPU bietet Bewegungssteuerungsfunktionen für den Betrieb von Schrittmotoren und Servomotoren mit Impulsschnittstelle. Die Bewegungssteuerungsfunktion übernimmt die Steuerung und Überwachung der Antriebe.

- Das Technologieobjekt "Achse" konfiguriert die Daten des mechanischen Antriebs, die Antriebsschnittstelle, die dynamischen Parameter und andere Eigenschaften des Antriebs.
- Sie konfigurieren die Impuls- und Richtungsausgänge der CPU zum Steuern des Antriebs.
- Ihr Anwenderprogramm nutzt die Bewegungssteuerungsanweisungen zum Steuern der Achse und zum Auslösen von Bewegungsaufgaben.
- Stellen Sie über die PROFINET-Schnittstelle eine Online-Verbindung zwischen der CPU und dem Programmiergerät her. Neben den Online-Funktionen der CPU stehen für die Bewegungssteuerung weitere Inbetriebnahme- und Diagnosefunktionen zur Verfügung.

Hinweis

Änderungen, die Sie an der Bewegungssteuerungskonfiguration vornehmen und im Betriebszustand RUN ins Zielsystem laden, werden erst dann wirksam, wenn die CPU von STOP in RUN geht.



- ① PROFINET
- ② Impuls- und Richtungsausgänge
- ③ Leistungsteil Schrittmotor
- ④ Leistungsteil Servomotor

Die DC/DC/DC-Varianten der S7-1200 CPU haben integrierte Ausgänge für die direkte Steuerung von Antrieben. Die Relaisvarianten der CPU benötigen das Signalboard mit DC-Ausgängen für die Antriebssteuerung.

Ein Signalboard (SB) erweitert die integrierten E/A um einige zusätzliche Ein- und Ausgänge. Ein SB mit zwei Digitalausgängen kann als Impuls- und Richtungsausgang zur Steuerung eines Motors verwendet werden. Ein SB mit vier Digitalausgängen kann für Impuls- und Richtungsausgänge zur Steuerung zweier Motoren verwendet werden. Integrierte Relaisausgänge können nicht als Impulsausgänge zur Motorsteuerung verwendet werden. Unabhängig davon, ob Sie integrierte E/A oder SB-E/A oder eine Kombination aus beiden verwenden, können Sie maximal vier Impulsgeneratoren definieren.

Die vier Impulsgeneratoren verfügen über Standard-E/A-Zuweisungen. Sie können die Impulsgeneratoren jedoch auch für jeden digitalen Ausgang an der CPU oder am SB konfigurieren. Impulsgeneratoren an der CPU können nicht SMs oder dezentralen E/A zugewiesen werden.

Hinweis

Impulsfolgen können von anderen Anweisungen im Anwenderprogramm nicht verwendet werden

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder Bewegungssteuerungsanweisungen) konfigurieren, steuern die entsprechenden Ausgangsadressen nicht mehr die Ausgänge. Wenn Ihr Anwenderprogramm einen Wert in einen Ausgang schreibt, der als Impulsgenerator genutzt wird, schreibt die CPU diesen Wert nicht in den physischen Ausgang.

Tabelle 10- 1 Maximale Anzahl von steuerbaren Antrieben

Typ der CPU		Integrierte E/A; Kein SB installiert		Mit einem SB (2 x DC-Ausgänge)		Mit einem SB (4 x DC-Ausgänge)	
		Mit Richtung	Ohne Richtung	Mit Richtung	Ohne Richtung	Mit Richtung	Ohne Richtung
CPU 1211C	DC/DC/DC	2	4	3	4	4	4
	AC/DC/RLS	0	0	1	2	2	4
	DC/DC/RLS	0	0	1	2	2	4
CPU 1212C	DC/DC/DC	3	4	3	4	4	4
	AC/DC/RLS	0	0	1	2	2	4
	DC/DC/RLS	0	0	1	2	2	4
CPU 1214C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLS	0	0	1	2	2	4
	DC/DC/RLS	0	0	1	2	2	4
CPU 1215C	DC/DC/DC	4	4	4	4	4	4
	AC/DC/RLS	0	0	1	2	2	4
	DC/DC/RLS	0	0	1	2	2	4
CPU 1217C	DC/DC/DC	4	4	4	4	4	4

Hinweis

Maximal sind vier Impulsgeneratoren möglich.

Unabhängig davon, ob Sie integrierte E/A, SB-E/A oder eine Kombination aus beiden verwenden, können Sie maximal vier Impulsgeneratoren definieren.

Tabelle 10- 2 CPU-Ausgang: Höchsthfrequenz

CPU	CPU-Ausgangskanal:	Impuls- und Richtungsausgang	A/B, vorwärts/rückwärts und Impuls/Richtung
1211C	Aa.0 bis Aa.3	100 kHz	100 kHz
1212C	Aa.0 bis Aa.3	100 kHz	100 kHz
	Aa.4, Aa.5	20 kHz	20 kHz
1214C und 1215C	Aa.0 bis Aa.3	100 kHz	100 kHz
	Aa.4 bis Ab.1	20 kHz	20 kHz
1217C	DAa.0 bis DAa.3 (.0+, .0- bis .3+, .3-)	1 MHz	1 MHz
	DAa.4 bis DAb.1	100 kHz	100 kHz

Tabelle 10- 3 Signalboard-Ausgang: Höchsthfrequenz (optionales Board)

Signalboard (SB)	SB-Ausgangskanal	Impuls- und Richtungsausgang	A/B, vorwärts/rückwärts und Impuls/Richtung
SB 1222, 200 kHz	DAe.0 bis DAe.3	200 kHz	200 kHz
SB 1223, 200 kHz	DAe.0, DAe.1	200 kHz	200 kHz
SB 1223	DAe.0, DAe.1	20 kHz	20 kHz

Tabelle 10- 4 Grenzfrequenzen von Impulsausgängen

Impulsausgang	Frequenz
Integriert	4 PTO: $2 \text{ Hz} \leq f \leq 1 \text{ MHz}$, 4 PTO: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ oder jede Kombination dieser Werte für 4 PTOs. ^{1 2}
Standard-SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
Schnelle SBs	$2 \text{ Hz} \leq f \leq 200 \text{ kHz}$

¹ In der nachstehenden Tabelle finden Sie vier mögliche Kombinationen der Ausgangsgeschwindigkeit bei der CPU 1217C.

² In der nachstehenden Tabelle finden Sie vier mögliche Kombinationen der Ausgangsgeschwindigkeit bei der CPU 1211C, CPU 1212C, CPU 1214C oder CPU 1215C.

Konfigurationsbeispiele für die Ausgangsgeschwindigkeit bei der CPU 1217C

Hinweis

Mit den integrierten Differentialausgängen kann die CPU 1217C vier Impulsausgänge bis 1 MHz generieren.

Die folgenden Beispiele zeigen vier mögliche Kombinationen von Ausgangsgeschwindigkeiten:

- Beispiel 1: 4 x 1-MHz-PTOs, kein Richtungsausgang
- Beispiel 2: 1 x 1-MHz-, 2 x 100-kHz- und 1 x 20-kHz-PTOs, alle mit Richtungsausgang
- Beispiel 3: 4 x 200-kHz-PTOs, kein Richtungsausgang
- Beispiel 4: 2 x 100-kHz-PTOs und 2 x 200-kHz-PTOs, alle mit Richtungsausgang

P = Impuls D = Richtung		Integrierte CPU-Ausgänge										Ausgänge schnelles SB				Ausgänge Standard-SB	
		1-MHz-Ausgänge (A)				100-kHz-Ausgänge (A)						200-kHz-Ausgänge (A)				20-kHz-Ausgänge (A)	
		0.0+	0.1+	0.2+	0.3+	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		0.0-	0.1-	0.2-	0.3-												
Beispiel 1: 4 x 1 MHz (kein Richtungsausgang)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Beispiel 2: 1 x 1 MHz, 2 x 100 kHz und 1 x 20 kHz (alle mit Richtungsausgang)	PTO1	P	D														
	PTO2					P	D										
	PTO3							P	D								
	PTO4															P	D
Beispiel 3: 4 x 200 kHz (kein Richtungsausgang)	PTO1											P					
	PTO2												P				
	PTO3													P			
	PTO4														P		
Beispiel 4: 2 x 100	PTO1					P	D										
	PTO2							P	D								
	PTO3											P	D				

P = Impuls D = Richtung		Integrierte CPU-Ausgänge										Ausgänge schnelles SB		Ausgänge Standard-SB				
kHz, 2 x 200 kHz (alle mit Richtung sausgan g)	PTO4														P	D		

Konfigurationsbeispiele für die Ausgangsgeschwindigkeit bei der CPU 1211C, CPU 1212C, CPU 1214C und CPU 1215C

Die folgenden Beispiele zeigen vier mögliche Kombinationen von Ausgangsgeschwindigkeiten:

- Beispiel 1: 4 x 100-kHz-PTOs, kein Richtungsausgang
- Beispiel 2: 2 x 100-kHz-PTOs und 2 x 20-kHz-PTOs, alle mit Richtungsausgang
- Beispiel 3: 4 x 200-kHz-PTOs, kein Richtungsausgang
- Beispiel 4: 2 x 100-kHz-PTOs und 2 x 200-kHz-PTOs, alle mit Richtungsausgang

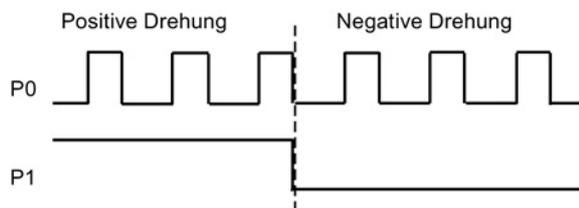
P = Impuls D = Richtung		Integrierte CPU-Ausgänge										Ausgänge schnelles SB				Ausgänge Standard-SB	
		100-kHz-Ausgänge (A)				20-kHz-Ausgänge (A)						200-kHz-Ausgänge (A)				20-kHz-Ausgänge (A)	
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	1.0	1.1	4.0	4.1	4.2	4.3	4.0	4.1
		CPU 1211C															
		CPU 1212C				CPU 1212C											
		CPU 1214C				CPU 1214C		CPU 1214C									
		CPU 1215C				CPU 1215C		CPU 1215C									
Beispiel 1: 4 x 100 kHz (kein Richtungsausgang)	PTO1	P															
	PTO2		P														
	PTO3			P													
	PTO4				P												
Beispiel 2: 2 x 100 kHz, 2 x 20 kHz (alle mit Richtungsausgang)	PTO1	P	D														
	PTO2			P	D												
	PTO3					P	D										
	PTO4							P	D								

P = Impuls D = Richtung	Integrierte CPU-Ausgänge										Ausgänge schnelles SB				Ausgänge Standard-SB	
	PTO1	PTO2	PTO3	PTO4	PTO5	PTO6	PTO7	PTO8	PTO9	PTO10	SB1	SB2	SB3	SB4	SB5	SB6
Beispiel 3: 4 x 200 kHz (kein Richtungs- ausgang)	PTO1											P				
	PTO2												P			
	PTO3													P		
	PTO4														P	
Beispiel 4: 2 x 100 kHz, 2 x 200 kHz (alle mit Richtungs- ausgang)	PTO1	P	D													
	PTO2			P	D											
	PTO3										P	D				
	PTO4												P	D		

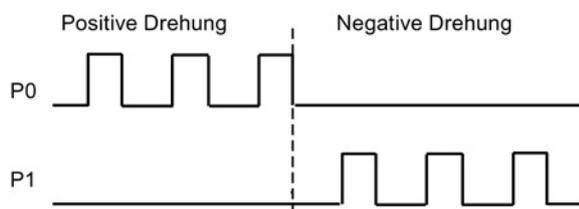
10.1 Phasenlage

Die Phasenlage-Schnittstelle zum Schrittmotor/Servoantrieb bietet die folgenden vier Optionen :

- PTO (Impuls A und Richtung B): Wenn Sie eine PTO-Option (Impuls A und Richtung B) auswählen, dann steuert ein Ausgang (P0) die Impulse und ein Ausgang (P1) steuert die Richtung. P1 ist high (aktiv), wenn Impulse in positiver Richtung erzeugt werden. P1 ist low (inaktiv), wenn Impulse in negativer Richtung erzeugt werden:



- PTO (Vorwärtszählen A und Rückwärtszählen B): Wenn Sie eine PTO-Option (Vorwärtszählen A und Rückwärtszählen B) auswählen, dann gibt ein Ausgang (P0) Impulse für positive Richtungen und ein anderer Ausgang (P1) Impulse für negative Richtungen aus.



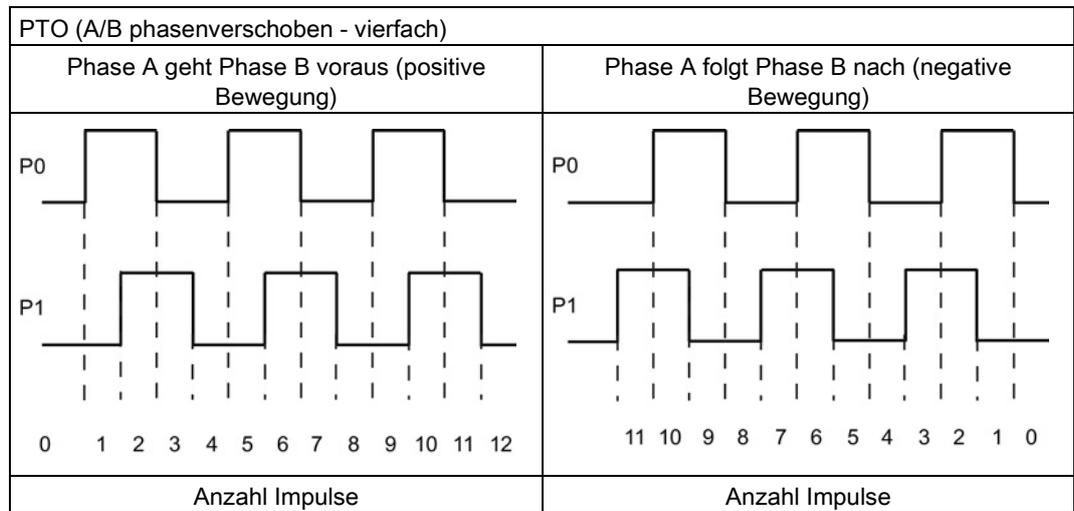
- PTO (A/B phasenverschoben): Wenn Sie eine PTO-Option (A/B phasenverschoben) auswählen, dann geben beide Ausgänge Impulse mit der angegebenen Geschwindigkeit, doch um 90 Grad phasenverschoben aus. Hierbei handelt es sich um eine 1X-Konfiguration, bei der ein Impuls die Zeitdauer zwischen positiven Übergängen von P0 aufweist. In diesem Fall wird die Richtung anhand des Ausgangs ermittelt, der zuerst von 0 nach 1 wechselt. Bei positiver Richtung geht P0 P1 voraus. Bei negativer Richtung geht P1 P0 voraus.

Die Anzahl der erzeugten Impulse basiert auf der Anzahl der 0-nach-1-Übergänge von Phase A. Das Phasenverhältnis legt die Richtung der Bewegung fest:

PTO (A/B phasenverschoben)	
Phase A geht Phase B voraus (positive Bewegung)	Phase A folgt Phase B nach (negative Bewegung)
<p>P0</p> <p>P1</p> <p>0 1 2 3</p>	<p>P0</p> <p>P1</p> <p>2 1 0</p>
Anzahl Impulse	Anzahl Impulse

- PTO (A/B phasenverschoben - vierfach): Wenn Sie eine PTO-Option (A/B phasenverschoben - vierfach) auswählen, dann geben beide Ausgänge Impulse mit der angegebenen Geschwindigkeit, aber um 90 Grad phasenverschoben aus. Bei der vierfachen Option handelt es sich um eine 4X-Konfiguration, bei der ein Impuls dem Übergang aller Ausgänge entspricht (positiv und negativ). In diesem Fall wird die Richtung anhand des Ausgangs ermittelt, der zuerst von 0 nach 1 wechselt. Bei positiver Richtung geht P0 P1 voraus. Bei negativer Richtung geht P1 P0 voraus.

Die vierfache Option basiert auf den positiven und negativen Übergängen sowohl von Phase A und Phase B. Die Anzahl der Übergänge wird von Ihnen konfiguriert. Das Phasenverhältnis (A geht B voraus oder B geht A voraus) legt die Richtung der Bewegung fest.



- PTO (Impuls und Richtung (Richtung abgewählt)): Wenn Sie den Richtungsausgang in einem PTO deaktivieren (Impuls und Richtung (Richtung abgewählt)), dann steuert der Ausgang (P0) die Impulse. Ausgang P1 wird nicht verwendet und steht für andere Zwecke im Programm zur Verfügung. In dieser Betriebsart werden von der CPU nur positive Bewegungssteuerungsbefehle akzeptiert. Wenn Sie diese Betriebsart wählen, erlaubt die Bewegungssteuerung keine unzulässigen negativen Konfigurationen. Sie können einen Ausgang einsparen, wenn Ihre Bewegungssteuerungsanwendung nur eine Richtung aufweist. Die einphasige Option (ein Ausgang) wird in der folgenden Abbildung gezeigt (dabei wird von positiver Polarität ausgegangen):



10.2 Konfigurieren eines Impulsgenerators

1. Hinzufügen eines Technologieobjekts:
 - Erweitern Sie in der Projektnavigation den Knoten "Technologieobjekte" und wählen Sie "Neues Objekt hinzufügen" aus.
 - Wählen Sie das Symbol "Achse" (benennen Sie es ggf. um) und klicken Sie auf "OK", um den Konfigurationseditor für das Achsenobjekt zu öffnen.
 - Rufen Sie die Eigenschaften "PTO für Achssteuerung auswählen" unter "Grundparameter" auf und wählen Sie den gewünschten Impuls aus.

Hinweis

Wenn PTO zuvor noch nicht in den CPU-Eigenschaften konfiguriert wurde, ist PTO so konfiguriert, dass einer der integrierten Ausgänge verwendet wird.

Wenn Sie ein Ausgangs-Signalboard verwenden, wählen Sie die Schaltfläche "Gerätekonfiguration", um die CPU-Eigenschaften aufzurufen. In den "Impulsoptionen" können Sie unter "Parametrierung" die Ausgangsquelle für einen Signalboard-Ausgang konfigurieren.

- Konfigurieren Sie die restlichen Grund- und erweiterten Parameter.
2. Programmieren Ihrer Anwendung: Fügen Sie die Anweisung MC_Power in einen Codebaustein ein.
 - Für den Achseneingang wählen Sie das Technologieobjekt "Achse", das Sie angelegt und konfiguriert haben.
 - Wird der Eingang Enable auf WAHR gesetzt, können die anderen Bewegungssteuerungsanweisungen funktionieren.
 - Wird der Eingang Enable auf FALSCH gesetzt, werden die anderen Bewegungssteuerungsanweisungen storniert.

Hinweis

Fügen Sie nur eine Anweisung MC_Power pro Achse ein.

3. Fügen Sie die anderen Bewegungssteuerungsanweisungen ein, um die erforderliche Bewegung zu erzeugen.

Hinweis

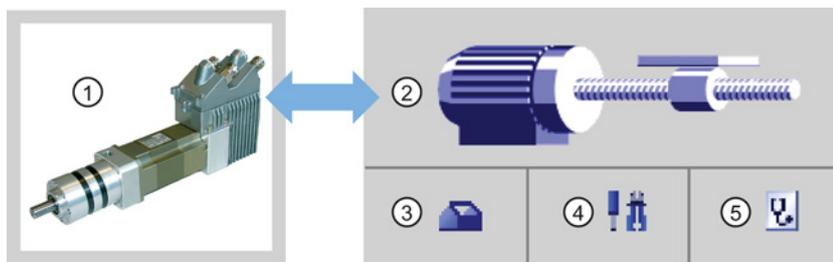
Konfigurieren eines Impulsgenerators für Signalboard-Ausgänge: Wählen Sie die Eigenschaften "Impulsgeneratoren (PTO/PWM)" für eine CPU (in der Gerätekonfiguration) und aktivieren Sie einen Impulsgenerator. Für die S7-1200 CPU V1.0, V2.0, V2.1 und V2.2 sind zwei Impulsgeneratoren verfügbar, während bei der S7-1200 CPU V3.0 und V4.0 vier Impulsgeneratoren zur Verfügung stehen. In demselben Konfigurationsbereich wählen Sie unter "Impulsoptionen" für den Impulsgenerator die Verwendung: "PTO" aus.

Hinweis

Die CPU berechnet Bewegungssteuerungsaufgaben in "Scheiben" bzw. Segmenten von 10 ms. Wenn eine Scheibe ausgeführt wird, wartet die nächste Scheibe in der Warteschlange auf die Ausführung. Wenn Sie die Bewegungssteuerungsaufgabe einer Achse unterbrechen (indem Sie eine andere neue Bewegungssteuerungsaufgabe für die Achse ausführen), kann die neue Bewegungssteuerungsaufgabe maximal 20 ms lang nicht ausgeführt werden (die Restdauer der aktuellen Scheibe plus die Scheibe in der Warteschlange).

10.3 Achse konfigurieren

STEP 7 bietet die Konfigurationswerkzeuge, Inbetriebnahmewerkzeuge und Diagnosewerkzeuge für das Technologieobjekt "Achse".



- | | |
|---------------------|------------------|
| ① Antrieb | ④ Inbetriebnahme |
| ② Technologieobjekt | ⑤ Diagnose |
| ③ Konfiguration | |

Hinweis

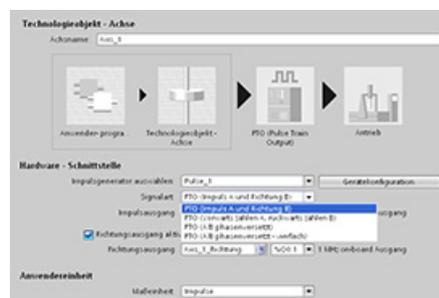
Bei CPUs mit einer Firmwareversion bis V2.2 benötigt die PTO die interne Funktionalität eines schnellen Zählers (HSC). Das bedeutet, dass der jeweilige HSC nicht anderweitig verwendet werden kann.

Die Zuordnung zwischen PTO und HSC ist fest vorgegeben. Wenn PTO1 aktiviert wird, erfolgt die Verbindung mit HSC1. Wenn PTO2 aktiviert wird, erfolgt die Verbindung mit HSC2. Sie können den aktuellen Wert (z. B. in ID1000) nicht überwachen, wenn Impulse auftreten.

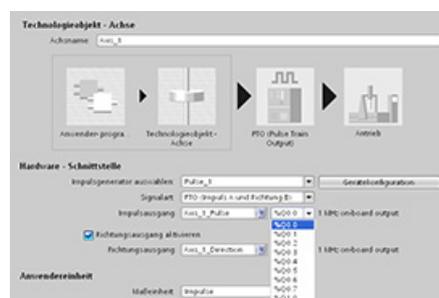
CPUs ab S7-1200 V3.0 haben diese Einschränkung nicht. Alle HSCs bleiben, wenn in diesen CPUs Impulsausgänge konfiguriert sind, für die Verwendung im Programm verfügbar.

Tabelle 10- 5 STEP 7-Werkzeuge für die Bewegungssteuerung

Werkzeug	Beschreibung
Konfiguration	<p>Konfiguriert die folgenden Eigenschaften des Technologieobjekts "Achse":</p> <ul style="list-style-type: none"> • Auswahl des gewünschten PTO und Konfiguration der Antriebsschnittstelle • Eigenschaften der Mechanik und des Übertragungsverhältnisses des Antriebs (bzw. der Maschine oder Anlage) • Eigenschaften der Positionsgrenzwerte, Dynamik und Referenzpunktfahrt <p>Speichern Sie die Konfiguration im Datenbaustein des Technologieobjekts.</p>
Inbetriebnahme	<p>Testet die Funktion Ihrer Achse, ohne dass ein Anwenderprogramm erstellt werden muss. Wenn das Werkzeug gestartet wird, wird das Steuerpanel angezeigt. Die folgenden Befehle sind im Steuerpanel verfügbar:</p> <ul style="list-style-type: none"> • Achse freigeben und deaktivieren • Achse im Tipbetrieb bewegen • Achse absolut und relativ bewegen • Referenzpunktfahrt der Achse durchführen • Fehler quittieren <p>Die Geschwindigkeit und die Beschleunigung/Verzögerung kann für die Bewegungssteuerungsbefehle angegeben werden. Das Steuerpanel zeigt auch den aktuellen Achsenzustand.</p>
Diagnose	Überwachung des aktuellen Zustands und der Fehlerinformationen für die Achse und den Antrieb.



Nachdem Sie das Technologieobjekt für die Achse angelegt haben, konfigurieren Sie die Achse, indem Sie die grundlegenden Parameter definieren, z. B. die PTO und die Konfiguration der Antriebsschnittstelle. Sie konfigurieren auch die anderen Eigenschaften der Achse wie Positionsgrenzwerte, Dynamik und Referenzpunktfahrt.

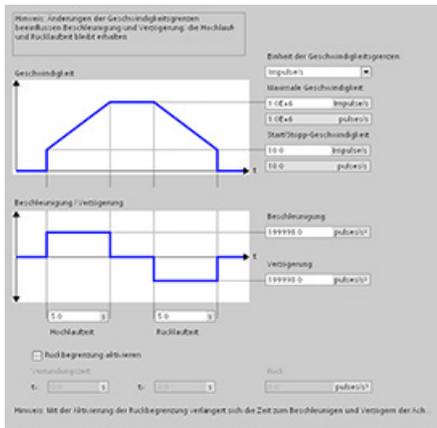


Hinweis

Sie müssen möglicherweise die Werte der Eingangsparameter der Bewegungssteuerungsanweisungen im Anwenderprogramm an die neue Einheit anpassen.



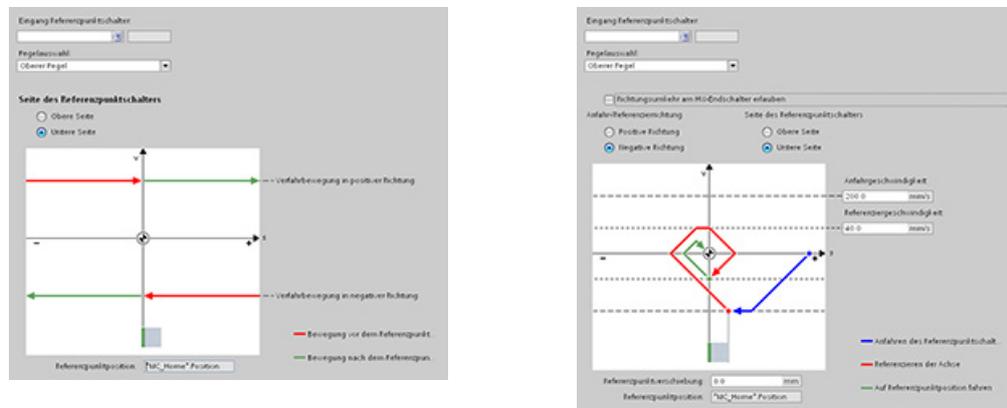
Konfigurieren Sie die Eigenschaften für die Antriebssignale, Antriebsmechanik und Positionsüberwachung (Hardware- und Software-Endschalter).



Sie konfigurieren die Bewegungssteuerungsdynamik und das Verhalten des Not-Aus-Befehls.



Ferner konfigurieren Sie das Verhalten der Referenzpunktfahrt (passiv und aktiv).



Im Steuerpanel "Inbetriebnahme" können Sie die Funktionalität unabhängig von Ihrem Anwenderprogramm testen.

 Klicken Sie auf das Symbol "Anlauf", um die Achse in Betrieb zu nehmen.

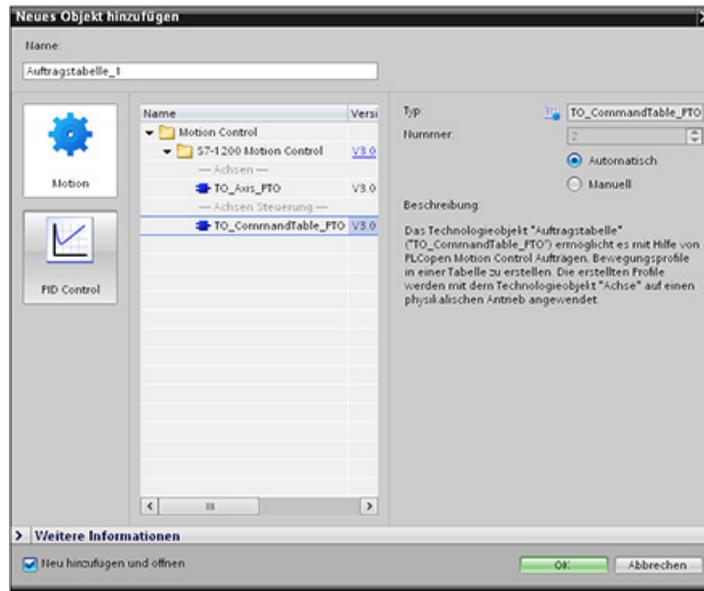
Das Steuerpanel zeigt den aktuellen Zustand der Achse. Sie können nicht nur die Achse aktivieren und deaktivieren, Sie können auch die Positionierung der Achse prüfen (sowohl absolut als auch relativ) und Sie können Geschwindigkeit, Beschleunigung und Verzögerung vorgeben. Weiterhin können Sie die Referenzpunktfahrt und den Tippbetrieb testen. Im Steuerpanel können Sie auch Fehler quittieren.

10.4 TO-Befehlstabelle PTO konfigurieren

Mit Hilfe der Technologieobjekte können Sie eine Anweisung CommandTable konfigurieren.

Hinzufügen eines Technologieobjekts

1. Erweitern Sie in der Projektnavigation den Knoten "Technologieobjekte" und wählen Sie "Neues Objekt hinzufügen" aus.
2. Wählen Sie das Symbol "Befehlstabelle" (benennen Sie es ggf. um) und klicken Sie auf "OK", um den Konfigurationseditor für das Befehlstabellenobjekt zu öffnen.



Schritte für Ihre Anwendung planen

Die gewünschte Bewegungsfolge können Sie im Konfigurationsfenster "Befehlstabelle" erstellen und das Ergebnis in der grafischen Darstellung im Kurvendiagramm prüfen.

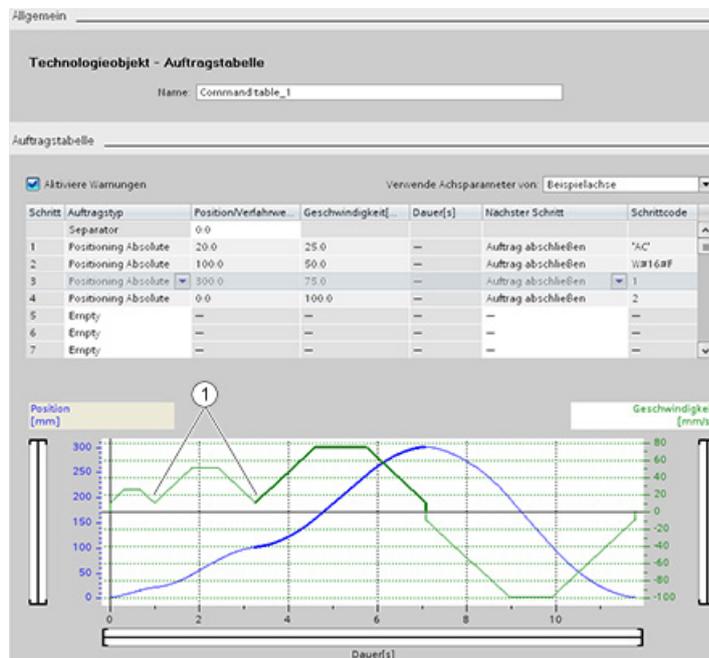
Sie können die Befehlstypen auswählen, die für die Verarbeitung der Befehlstabelle verwendet werden sollen. Bis zu 32 Schritte können eingegeben werden. Die Befehle werden der Reihe nach verarbeitet und erzeugen mühelos ein komplexes Bewegungsprofil.

Tabelle 10- 6 MC_CommandTable-Befehlstypen

Befehlstyp	Beschreibung
Empty	Dieser Befehl dient als Platzhalter für jeden hinzuzufügenden Befehl. Der leere Eintrag wird bei der Verarbeitung der Befehlstabelle ignoriert.
Halt	Der Befehl hält die Achse an. Hinweis: Der Befehl wird nur nach einem Befehl "Velocity setpoint" durchgeführt.
Positioning Relative	Der Befehl positioniert die Achse basierend auf der Distanz. Er bewegt die Achse um die angegebene Distanz mit der angegebenen Geschwindigkeit.
Positioning Absolute	Der Befehl positioniert die Achse basierend auf der Lage. Er bewegt die Achse mit der vorgegebenen Geschwindigkeit an die angegebene Position.

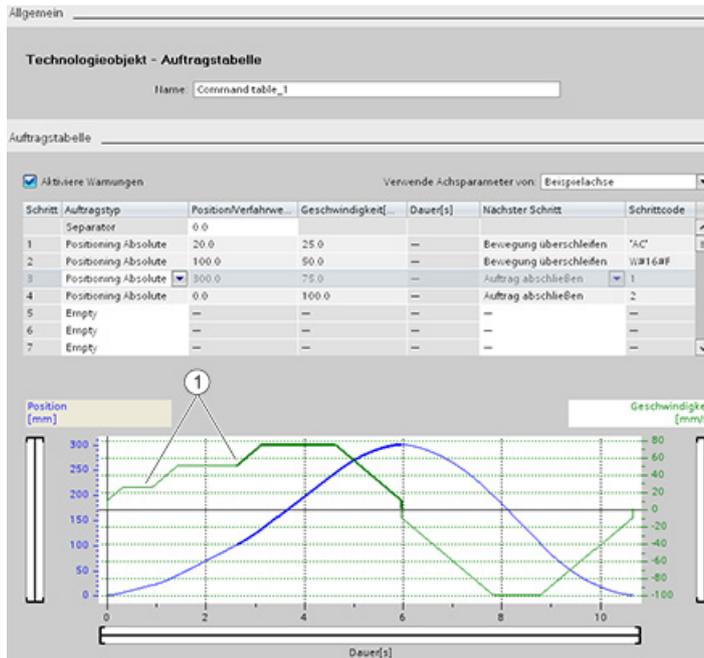
Befehlstyp	Beschreibung
Velocity setpoint	Der Befehl bewegt die Achse mit der angegebenen Geschwindigkeit.
Wait	Der Befehl wartet, bis der angegebene Zeitraum abgelaufen ist. "Wait" stoppt keine aktive Verfahrbewegung.
Separator	Der Befehl fügt eine Trennlinie ("Separator") oberhalb der ausgewählten Linie ein. Die Trennlinie ermöglicht die Definition mehrerer Profile in einer Befehlstabelle.

In der folgenden Abbildung wird "Command complete" als Übergang zum nächsten Schritt verwendet. Diese Art des Übergangs ermöglicht es Ihrem Gerät, auf die Start-/Stoppgeschwindigkeit zu verzögern und dann zu Beginn des nächsten Schritts erneut zu beschleunigen.



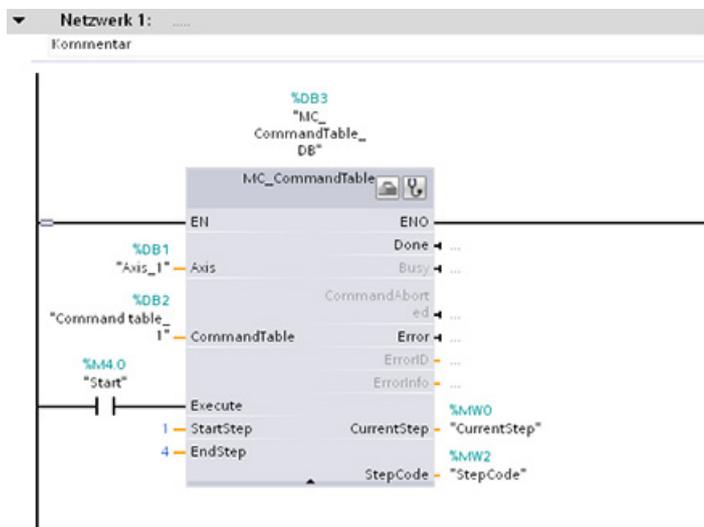
① Die Achse verzögert zwischen den Schritten auf die Start-/Stoppgeschwindigkeit.

In der folgenden Abbildung wird "Blending motion" als Übergang zum nächsten Schritt verwendet. Diese Art des Übergangs ermöglicht es Ihrem Gerät, die Geschwindigkeit bis zum Start des nächsten Schritts beizubehalten, was zu einem glatten Übergang des Geräts von einem Schritt zum nächsten führt. Bei Verwendung von "Blending motion" kann sich die Gesamtdauer für die vollständige Ausführung eines Profils verkürzen. Ohne Blending dauert die Ausführung des Beispiels sieben Sekunden. Mit Blending wird die Ausführungszeit um eine Sekunde auf insgesamt sechs Sekunden reduziert.



① Die Achse bewegt sich ununterbrochen weiter und beschleunigt oder verzögert auf die Geschwindigkeit des nächsten Schritts, wodurch Zeit gespart und mechanischer Verschleiß verringert wird.

Die Funktionsweise Ihrer Befehlstabelle wird, wie im Folgenden gezeigt, von der Anweisung MC_CommandTable gesteuert:



10.5 Bewegungssteuerungsanweisungen

10.5.1 Übersicht MC-Anweisungen

Die Bewegungssteuerungsanweisungen verwenden einen zugehörigen Technologie-Datenbaustein und die dafür zugewiesene PTO (Impulsfolge) der CPU, um die Bewegung einer Achse zu steuern.

- MC_Power (Seite 238) aktiviert und deaktiviert eine Achse für die Bewegungssteuerung.
- MC_Reset (Seite 241) setzt alle Bewegungssteuerungsfehler zurück. Alle Bewegungssteuerungsfehler, die quittiert werden können, werden quittiert.
- MC_Home (Seite 242) stellt die Beziehung zwischen dem Achsensteuerungsprogramm und dem mechanischen Positionierungssystem der Achse her.
- MC_Halt (Seite 245) storniert alle Bewegungsvorgänge und bewirkt den Stopp der Achsenbewegung. Die Halteposition ist nicht definiert.
- MC_MoveAbsolute (Seite 247) startet die Bewegung zu einer absoluten Position. Der Auftrag ist beendet, wenn die Zielposition erreicht ist.
- MC_MoveRelative (Seite 249) startet eine Positionierbewegung relativ zur Startposition.
- MC_MoveVelocity (Seite 251) bewirkt, dass sich die Achse mit der angegebenen Geschwindigkeit bewegt.
- MC_MoveJog (Seite 254) führt den Tippbetrieb zu Test- und Anlaufzwecken aus.
- MC_CommandTable (Seite 257) führt Achssteuerungsbefehle als Bewegungsfolge aus.
- MC_ChangeDynamic (Seite 260) ändert die Dynamikeinstellungen der Achse.
- MC_WriteParam (Seite 262) schreibt bestimmte Parameter, um die Funktionalität der Achse im Anwenderprogramm zu ändern.
- MC_ReadParam (Seite 264) liest bestimmte Parameter, die die aktuelle Position, Geschwindigkeit usw. der im Achseneingang definierten Achse angeben.

CPU-Firmwarestände

Wenn Sie mit einer S7-1200 CPU mit Firmware V4.0 arbeiten, wählen Sie Version V4.0 der Bewegungsanweisungen.

Wenn Sie mit einer S7-1200 CPU mit Firmwareversion V3.0 oder niedriger arbeiten, wählen Sie die entsprechende Version V3.0, V2.0 oder V1.0 der Bewegungsanweisungen.

10.5.2 Anweisung MC_Power (Achse freigeben/sperrn)

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- 7 Anweisung MC_Power

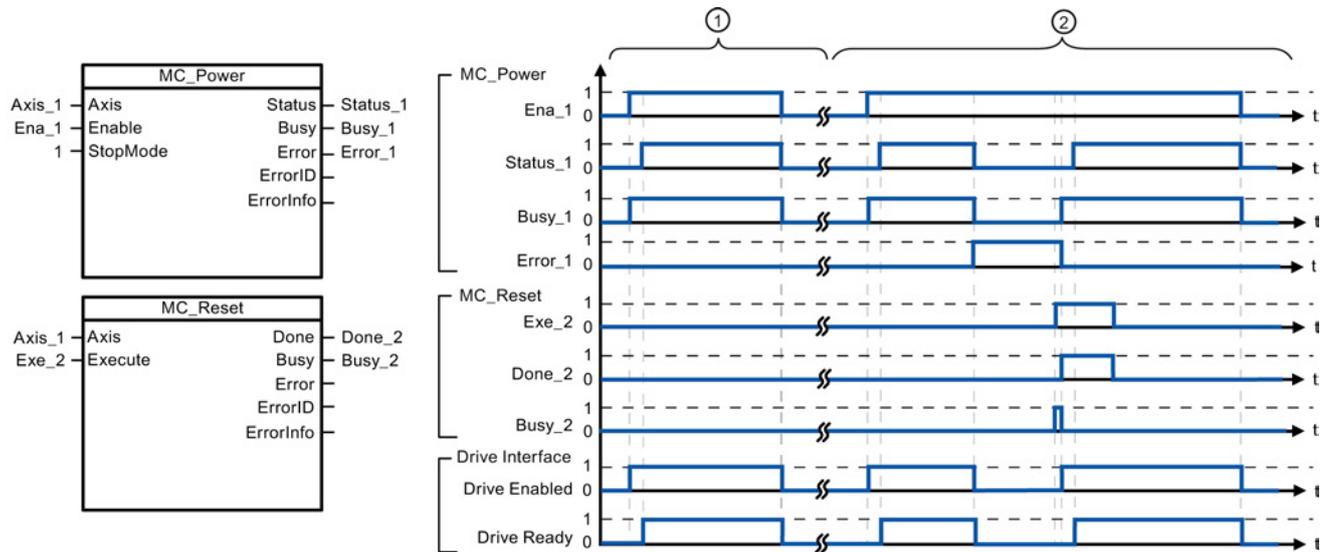
KOP/FUP	SCL	Beschreibung
	<pre>"MC_Power_DB" (Axis:= _multi_fb_in_, Enable:= _bool_in_, StopMode:= _int_in_, Status=> _bool_out_, Busy=> _bool_out_, Error=> _bool_out_, ErrorID=> _word_out_, ErrorInfo=> _word_out_);</pre>	<p>Die Bewegungssteuerungsanweisung MC_Power aktiviert oder deaktiviert eine Achse. 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.

Tabelle 10- 8 Parameter für die MC_Power-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1 Technologieobjekt "Achse"
Enable	IN	Bool <ul style="list-style-type: none"> • FALSE (Standard): Alle aktiven Aufgaben werden entsprechend dem parametrisierten "StopMode" abgebrochen und die Achse wird gestoppt. • TRUE: Die Bewegungssteuerung versucht, die Achse zu aktivieren.

Parameter und Datentyp		Datentyp	Beschreibung
StopMode	IN	Int	<ul style="list-style-type: none"> • 0: Not-Aus: Steht eine Anforderung zur Deaktivierung der Achse an, bremst die Achse mit der konfigurierten Notfallverzögerung. Die Achse wird nach Erreichen des Stillstands deaktiviert. • 1: Sofortiger Stopp: Steht eine Anforderung zur Deaktivierung der Achse an, wird die Achse ohne Verzögerung deaktiviert. Der Impulsausgang wird sofort gestoppt. • 2: Not-Aus mit Rucksteuerung: Steht eine Anforderung zur Deaktivierung der Achse an, bremst die Achse mit der konfigurierten Not-Aus-Verzögerung. Wenn die Rucksteuerung aktiviert ist, wird der konfigurierte Ruck berücksichtigt. Die Achse wird nach Erreichen des Stillstands deaktiviert.
Status	OUT	Bool	Status der Achsenfreigabe: <ul style="list-style-type: none"> • FALSE: Die Achse ist deaktiviert: <ul style="list-style-type: none"> – Die Achse führt keine Bewegungssteuerungsaufgaben aus und nimmt keine neuen Aufgaben an (Ausnahme: Aufgabe MC_Reset). – Die Achse ist nicht am Referenzpunkt. – Beim Deaktivieren wechselt der Zustand erst dann nach FALSCH, wenn die Achse den Stillstand erreicht. • TRUE: Die Achse ist aktiviert: <ul style="list-style-type: none"> – Die Achse ist bereit, Bewegungssteuerungsaufgaben auszuführen. – Beim Aktivieren der Achse wechselt der Zustand erst dann nach WAHR, wenn das Signal "Antrieb bereit" ansteht. Wurde die Antriebsschnittstelle "Antrieb bereit" während der Achsenkonfiguration nicht eingerichtet, wechselt der Zustand sofort nach WAHR.
Busy	OUT	Bool	FALSE: MC_Power ist nicht aktiv. TRUE: MC_Power ist aktiv.
Error	OUT	Bool	FALSE: Kein Fehler TRUE: In der Bewegungssteuerungsanweisung "MC_Power" oder im zugehörigen Technologieobjekt ist ein Fehler aufgetreten. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID



- ① Eine Achse wird aktiviert und dann wieder deaktiviert. Nachdem der Antrieb das Signal "Antrieb bereit" an die CPU zurückgemeldet hat, kann die erfolgreiche Aktivierung über "Status_1" ausgelesen werden.
- ② Nach einer Achsenfreigabe ist ein Fehler aufgetreten, der verursacht hat, dass die Achse deaktiviert wurde. Der Fehler wird behoben und mit "MC_Reset" quittiert. Die Achse wird dann wieder aktiviert.

Um eine Achse mit konfigurierter Antriebsschnittstelle zu aktivieren, gehen Sie wie folgt vor:

1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie den Eingangsparameter "StopMode" mit dem gewünschten Wert. Setzen Sie den Eingangsparameter "Enable" auf WAHR.

Der Freigabeausgang für "Antrieb freigegeben" wechselt nach WAHR, um die Spannung für den Antrieb zu aktivieren. Die CPU wartet auf das Signal "Antrieb bereit" des Antriebs.

Wenn das Signal "Antrieb bereit" am konfigurierten Bereitschaftseingang der CPU verfügbar ist, wird die Achse freigegeben. Der Ausgangsparameter "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable geben den Wert WAHR an.

Um eine Achse ohne konfigurierte Antriebsschnittstelle zu aktivieren, gehen Sie wie folgt vor:

1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie den Eingangsparameter "StopMode" mit dem gewünschten Wert. Setzen Sie den Eingangsparameter "Enable" auf WAHR. Die Achse ist aktiviert. Der Ausgangsparameter "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable geben den Wert WAHR an.

Um eine Achse zu deaktivieren, gehen Sie wie folgt vor:

1. Bringen Sie die Achse zum Stillstand.
Anhand der Technologieobjektvariable <Achsenname>.StatusBits.StandStill können Sie erkennen, wann sich die Achse im Stillstand befindet.
2. Setzen Sie den Eingangsparameter "Enable" nach Erreichen des Stillstands auf FALSCH.
3. Wenn die Ausgangsparameter "Busy" und "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable den Wert FALSCH anzeigen, ist die Deaktivierung der Achse beendet.

10.5.3 Anweisung MC_Reset (Fehler bestätigen)

Tabelle 10- 9 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.

Tabelle 10- 10 Parameter der Anweisung MC_Reset

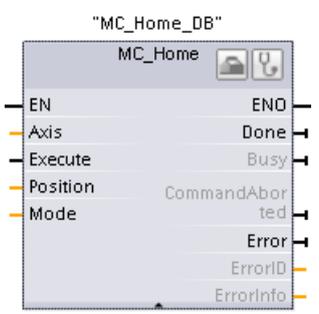
Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1
Execute	IN	Bool
Restart	IN	Bool
		TRUE = Laden der Achsenkonfiguration aus dem Ladespeicher in den Arbeitsspeicher. Der Befehl kann nur bei deaktivierter Achse ausgeführt werden.
		FALSE = Anstehende Fehler werden quittiert
Done	OUT	Bool
		WAHR = Fehler wurde quittiert.
Busy	OUT	Bool
		WAHR = Die Aufgabe wird ausgeführt.
Error	OUT	Bool
		WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUTP	Word
		Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word
		Fehlerinfo-ID für Parameter ErrorID

Um einen Fehler mit MC_Reset zu quittieren, gehen Sie folgendermaßen vor:

1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Starten Sie die Quittierung des Fehlers mit einer steigenden Flanke am Eingangsparameter Execute.
3. Der Fehler wurde quittiert, wenn Done gleich WAHR ist und die Technologieobjektvariable <Achsenname>.StatusBits.Error gleich FALSCH ist.

10.5.4 Anweisung MC_Home (Referenzpunktfahrt der Achse durchführen)

Tabelle 10- 11 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_);</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.

Die folgenden Arten von Referenzpunktfahrten stehen zur Verfügung:

- Direkte Referenzpunktfahrt absolut (Mode = 0): Die aktuelle Achsenposition wird auf den Wert des Parameters "Position" gesetzt.
- Direkte Referenzpunktfahrt relativ (Mode = 1): Die aktuelle Achsenposition wird um den Wert des Parameters "Position" versetzt.
- Passive Referenzpunktfahrt (Mode = 2): Während der passiven Referenzpunktfahrt führt die Anweisung MC_Home keine Referenzpunktfahrtbewegung durch. Die für diesen Schritt erforderliche Verfahrbewegung müssen Sie über andere Bewegungssteuerungsanweisungen implementieren. Wenn der Referenzpunktwechsel erkannt wird, wird die Achse an den Referenzpunkt gefahren.
- Aktive Referenzpunktfahrt (Mode = 3): Die Referenzpunktfahrt wird automatisch durchgeführt.

Tabelle 10- 12 Parameter für die MC_Home-Anweisung

Parameter und Datentyp		Datentyp	Beschreibung
Axis	IN	TO_Axis_PTO	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke
Position	IN	Real	<ul style="list-style-type: none"> • Mode = 0, 2 und 3 (Absolute Position der Achse nach Beendigung der Referenzpunktfahrt) • Mode = 1 (Korrekturwert für die aktuelle Achsenposition) Grenzwerte: $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$
Mode	IN	Int	Art der Referenzpunktfahrt <ul style="list-style-type: none"> • 0: Direkte Referenzpunktfahrt absolut Die neue Achsenposition ist der Positionswert des Parameters "Position". • 1: Direkte Referenzpunktfahrt relativ Die neue Achsenposition ist die aktuelle Achsenposition + Positionswert des Parameters "Position". • 2: Passive Referenzpunktfahrt Referenzpunktfahrt entsprechend der Achsenkonfiguration. Nach der Referenzpunktfahrt wird der Wert des Parameters "Position" als neue Achsenposition eingestellt. • 3: Aktive Referenzpunktfahrt Referenzpunktannäherung entsprechend der Achsenkonfiguration. Nach der Referenzpunktfahrt wird der Wert des Parameters "Position" als neue Achsenposition eingestellt.
Done	OUT	Bool	WAHR = Aufgabe durchgeführt
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID

Hinweis

Die Referenzpunktfahrt der Achse geht unter den folgenden Voraussetzungen verloren

- Deaktivierung der Achse durch die Anweisung MC_Power
- Umschalten zwischen Automatik- und Handbetrieb
- Nach dem Start der aktiven Referenzpunktfahrt (nach erfolgreicher Durchführung der Referenzpunktfahrt steht die Referenzpunktfahrt der Achse erneut zur Verfügung.)
- Nach Aus- und wieder Einschalten der CPU
- Nach Neustart der CPU (RUN-in-STOP oder STOP-in-RUN)

Um die Achse an den Referenzpunkt zu fahren, gehen Sie folgendermaßen vor:

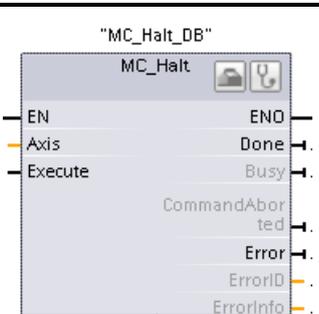
1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie die erforderlichen Eingangsparameter mit Werten und starten Sie die Referenzpunktfahrt mit einer steigenden Flanke am Eingangsparameter "Execute".
3. Wenn der Ausgangsparameter "Done" und die Technologieobjektvariable <Achsenname>.StatusBits.HomingDone den Wert WAHR angeben, ist die Referenzpunktfahrt beendet.

Tabelle 10- 13 Übersteuerungsantwort

Betriebsart	Beschreibung	
0 oder 1	Die Aufgabe MC_Home kann von keiner anderen Bewegungssteuerungsaufgabe abgebrochen werden. Die neue Aufgabe MC_Home bricht keine aktiven Bewegungssteuerungsaufgaben ab. Positionsbezogene Bewegungssteuerungsaufgaben werden nach der Referenzpunktfahrt entsprechend der neuen Referenzpunktposition (Wert am Eingangsparameter Position) wieder aufgenommen.	
2	Die Aufgabe MC_Home kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden: Mode der Aufgabe MC_Home = 2, 3: Die neue Aufgabe MC_Home bricht die folgende aktive Bewegungssteuerungsaufgabe ab. Mode der Aufgabe MC_Home = 2: Positionsbezogene Bewegungssteuerungsaufgaben werden nach der Referenzpunktfahrt entsprechend der neuen Referenzpunktposition (Wert am Eingangsparameter Position) wieder aufgenommen.	
3	Die Aufgabe MC_Home kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden: <ul style="list-style-type: none"> • MC_Home Mode = 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog 	Die neue Aufgabe MC_Home bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab: <ul style="list-style-type: none"> • MC_Home-Modus = 2, 3 • MC_Halt • MC_MoveAbsolute • MC_MoveRelative • MC_MoveVelocity • MC_MoveJog

10.5.5 Anweisung MC_Halt (Achse pausieren)

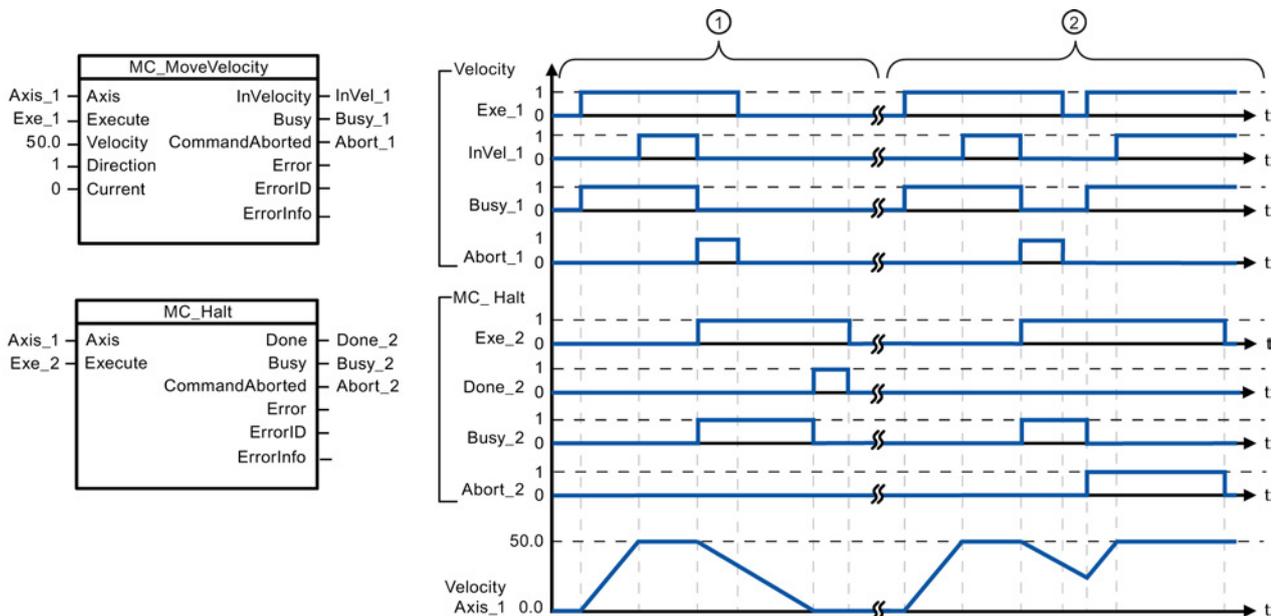
Tabelle 10- 14 Anweisung MC_Halt

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.

Tabelle 10- 15 Parameter für die MC_Halt-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung	
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke
Done	OUT	Bool	WAHR = Nullgeschwindigkeit erreicht
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter ErrorID



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 5,0

- ① Die Achse wird von einer Aufgabe MC_Halt gebremst, bis sie zum Stillstand kommt. Der Stillstand der Achse wird über "Done_2" gemeldet.
- ② Während die Aufgabe MC_Halt die Achse bremst, wird die Aufgabe von einer anderen Bewegungssteuerungsaufgabe abgebrochen. Der Abbruch wird über "Abort_2" gemeldet.

Übersteuerungsantwort

Die Aufgabe MC_Halt kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

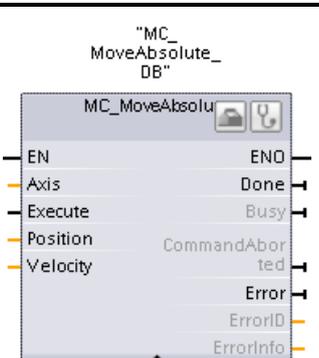
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Die neue Aufgabe MC_Halt bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.6 Anweisung MC_MoveAbsolute (Achse absolut positionieren)

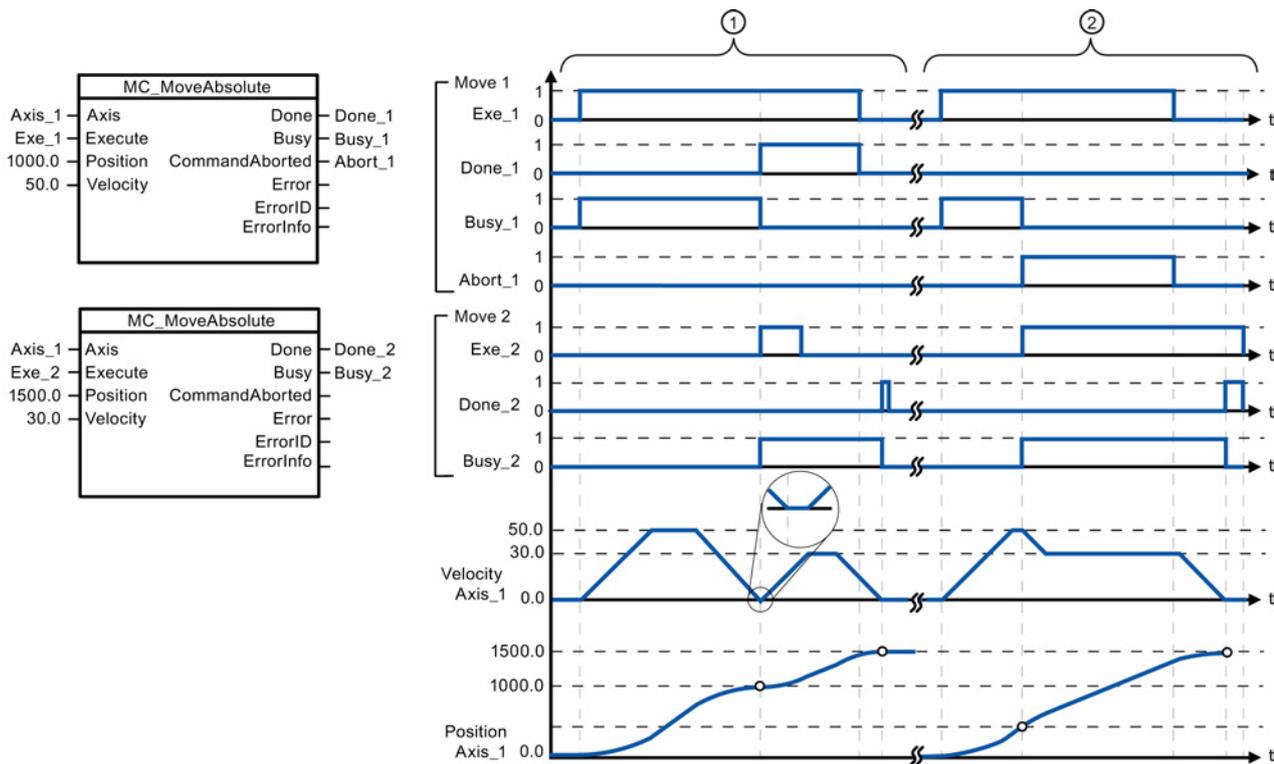
Tabelle 10- 16 Anweisung MC_MoveAbsolute

KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveAbsolute_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Position:=_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_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>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_MoveAbsolute_DB" der Name des Instanz-DBs.

Tabelle 10- 17 Parameter für die MC_MoveAbsolute-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1 Technologieobjekt "Achse"
Execute	IN	Bool Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Position	IN	Real Absolute Zielposition (Standardwert: 0.0) Grenzwerte: $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$
Velocity	IN	Real Achsgeschwindigkeit (Standardwert: 10.0) Diese Geschwindigkeit wird wegen der konfigurierten Beschleunigung und Verzögerung und der anzufahrenden Zielposition nicht immer erreicht. Grenzwerte: Start-/Stoppgeschwindigkeit $\leq \text{Velocity} \leq$ Maximalgeschwindigkeit
Done	OUT	Bool WAHR = Absolute Zielposition erreicht
Busy	OUT	Bool WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Eine Achse wird mit der Aufgabe MC_MoveAbsolute zur absoluten Position 1000,0 gefahren. Wenn die Achse die Zielposition erreicht, wird dies über "Done_1" gemeldet. Wenn "Done_1" = WAHR ist, wird eine andere Aufgabe MC_MoveAbsolute mit der Zielposition 1500,0 gestartet. Wegen der Antwortzeiten (z. B. Zykluszeit des Anwenderprogramms usw.) kommt die Achse kurz zum Stillstand (siehe vergrößerter Ausschnitt). Wenn die Achse die neue Zielposition erreicht, wird dies über "Done_2" gemeldet.
- ② Eine aktive Aufgabe MC_MoveAbsolute wird von einer anderen Aufgabe MC_MoveAbsolute abgebrochen. Der Abbruch wird über "Abort_1" gemeldet. Die Achse wird dann mit der neuen Geschwindigkeit zur neuen Zielposition 1500,0 gefahren. Wenn die neue Zielposition erreicht ist, wird dies über "Done_2" gemeldet.

Übersteuerungsantwort

Die Aufgabe MC_MoveAbsolute kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

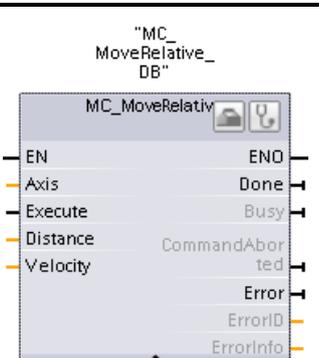
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Die neue Aufgabe MC_MoveAbsolute bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.7 Anweisung MC_MoveRelative (Achse relativ positionieren)

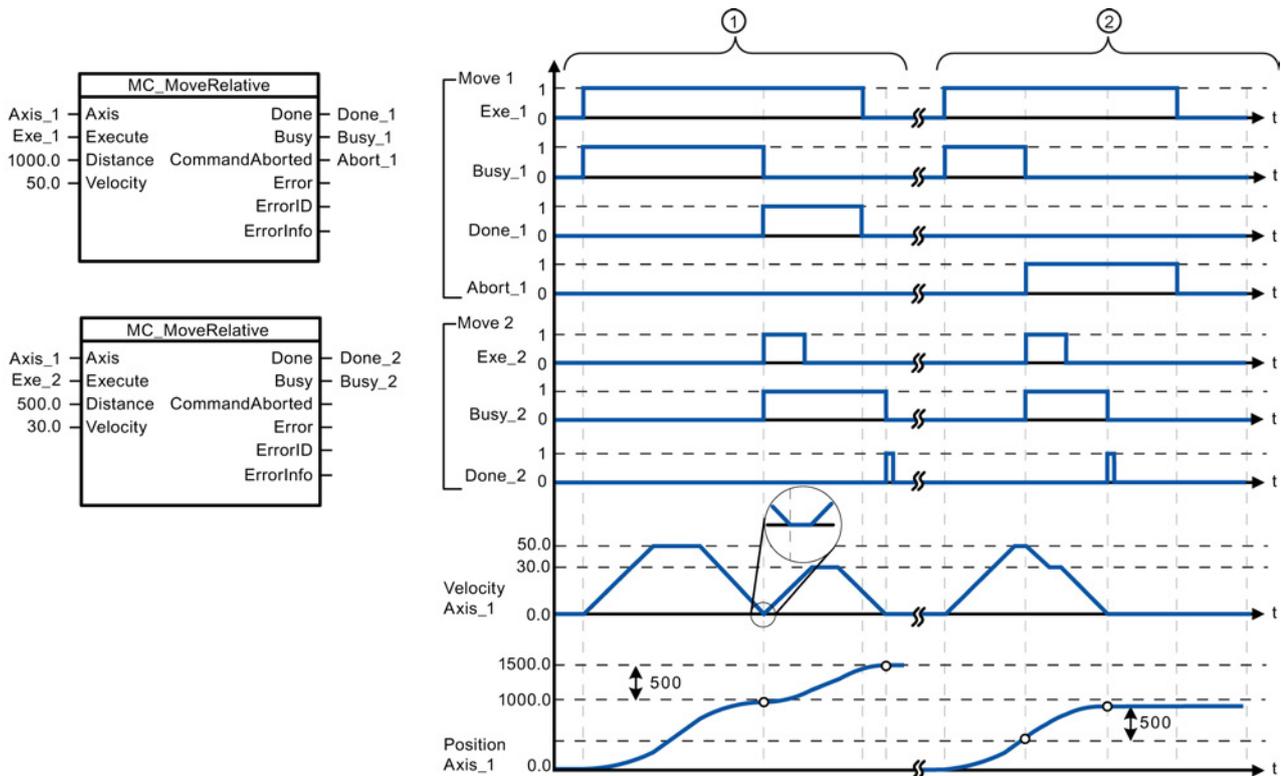
Tabelle 10- 18 Anweisung MC_MoveRelative

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>

- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_MoveRelative_DB" der Name des Instanz-DBs.

Tabelle 10- 19 Parameter für die MC_MoveRelative-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Distance	IN	Real	Verfahrweg für den Positioniervorgang (Standardwert: 0.0) Grenzwerte: $-1,0e^{12} \leq \text{Distance} \leq 1,0e^{12}$
Velocity	IN	Real	Achsgeschwindigkeit (Standardwert: 10.0) Diese Geschwindigkeit wird wegen der konfigurierten Beschleunigung und Verzögerung und des zu fahrenden Wegs nicht immer erreicht. Grenzwerte: Start-/Stoppgeschwindigkeit $\leq \text{Velocity} \leq$ Maximalgeschwindigkeit
Done	OUT	Bool	WAHR = Zielposition erreicht
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Die Achse wird von einer Aufgabe MC_MoveRelative den Weg ("Distance") 1000,0 gefahren. Wenn die Achse die Zielposition erreicht, wird dies über "Done_1" gemeldet. Wenn "Done_1" = WAHR ist, wird eine andere Aufgabe MC_MoveRelative mit dem Verfahrensweg 500,0 gestartet. Wegen der Antwortzeiten (z. B. Zykluszeit des Anwenderprogramms) kommt die Achse kurz zum Stillstand (siehe vergrößerter Ausschnitt). Wenn die Achse die neue Zielposition erreicht, wird dies über "Done_2" gemeldet.
- ② Eine aktive Aufgabe MC_MoveRelative wird von einer anderen Aufgabe MC_MoveRelative abgebrochen. Der Abbruch wird über "Abort_1" gemeldet. Die Achse wird dann mit der neuen Geschwindigkeit den neuen Weg ("Distance") 500,0 gefahren. Wenn die neue Zielposition erreicht ist, wird dies über "Done_2" gemeldet.

Übersteuerungsantwort

Die Aufgabe MC_MoveRelative kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

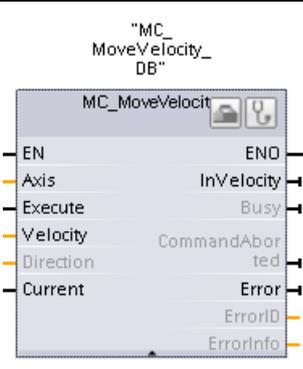
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Die neue Aufgabe MC_MoveRelative bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.8 Anweisung MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen)

Tabelle 10- 20 Anweisung MC_MoveVelocity

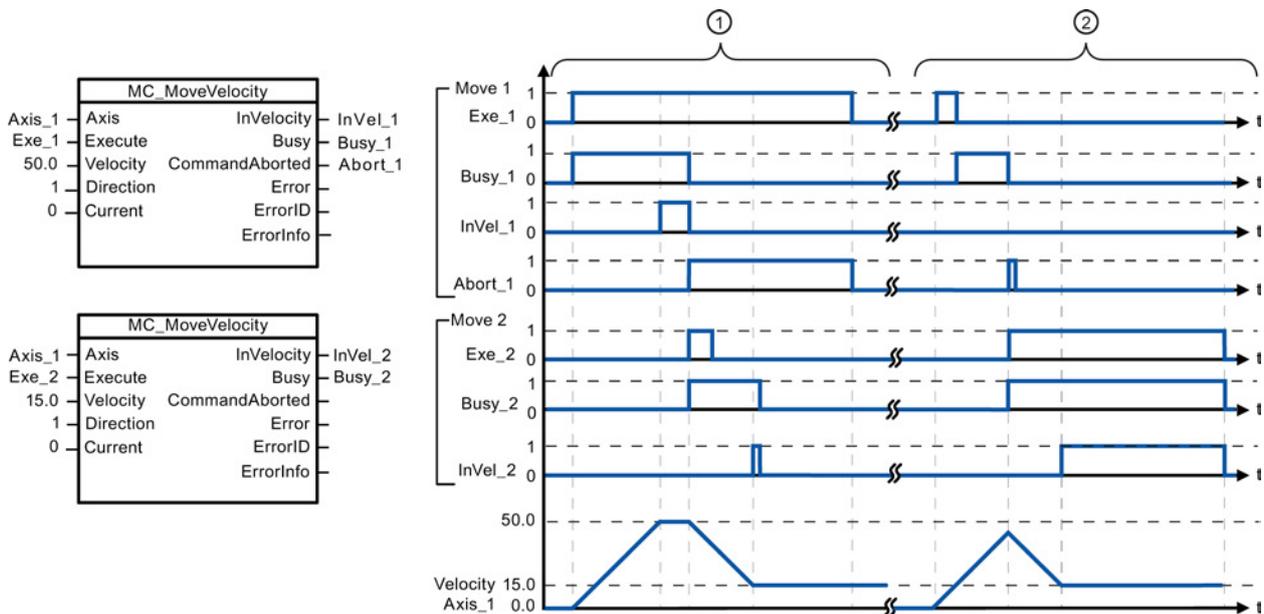
KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveVelocity_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Velocity:=_real_in_, Direction:=_int_in_, Current:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Mit 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.

Tabelle 10- 21 Parameter für die MC_MoveVelocity-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1 Technologieobjekt "Achse"
Execute	IN	Bool Starten der Aufgabe bei einer positiven Flanke (Standardwert: Falsch)
Velocity	IN	Real Geschwindigkeitsangabe für die Achsenbewegung (Standardwert: 10.0) Grenzwerte: Start-/Stoppgeschwindigkeit ≤ Velocity ≤ Maximalgeschwindigkeit (Velocity = 0,0 ist zulässig)
Direction	IN	Int Richtungsangabe: <ul style="list-style-type: none"> • 0: Die Drehrichtung entspricht dem Vorzeichen des Werts im Parameter "Velocity" (Standardwert) • 1: Positive Drehrichtung (das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert). • 2: Negative Drehrichtung (das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert).

Parameter und Datentyp		Datentyp	Beschreibung
Current	IN	Bool	<p>Aktuelle Geschwindigkeit beibehalten:</p> <ul style="list-style-type: none"> • FALSCH: "Aktuelle Geschwindigkeit beibehalten" ist deaktiviert. Die Werte der Parameter "Velocity" und "Direction" werden verwendet. (Standardwert) • WAHR: "Aktuelle Geschwindigkeit beibehalten" ist aktiviert. Die Werte der Parameter "Velocity" und "Direction" werden nicht berücksichtigt. <p>Wenn die Achse die Bewegung mit der aktuellen Geschwindigkeit wiederaufnimmt, gibt der Parameter "InVelocity" den Wert TRUE. aus.</p>
InVelocity	OUT	Bool	<p>WAHR:</p> <ul style="list-style-type: none"> • Wenn "Current" = FALSE: Die im Parameter "Velocity" angegebene Geschwindigkeit wurde erreicht. • Wenn "Current" = TRUE: Die Achse fährt zur Startzeit mit der aktuellen Geschwindigkeit.
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 10,0

- ① Eine aktive Aufgabe MC_MoveVelocity meldet über "InVel_1", dass die Zielgeschwindigkeit erreicht wurde. Sie wird dann von einer anderen Aufgabe MC_MoveVelocity abgebrochen. Der Abbruch wird über "Abort_1" gemeldet. Wenn die neue Zielgeschwindigkeit 15,0 erreicht ist, wird dies über "InVel_2" gemeldet. Die Achse bewegt sich dann mit der neuen konstanten Geschwindigkeit weiter.
- ② Eine aktive Aufgabe MC_MoveVelocity wird von einer anderen Aufgabe MC_MoveVelocity abgebrochen, bevor sie ihre Zielgeschwindigkeit erreicht. Der Abbruch wird über "Abort_1" gemeldet. Wenn die neue Zielgeschwindigkeit 15,0 erreicht ist, wird dies über "InVel_2" gemeldet. Die Achse bewegt sich dann mit der neuen konstanten Geschwindigkeit weiter.

Übersteuerungsantwort

Die Aufgabe MC_MoveVelocity kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Die neue Aufgabe MC_MoveVelocity bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Hinweis

Verhalten bei auf Null gesetzter Geschwindigkeit (Velocity = 0,0)

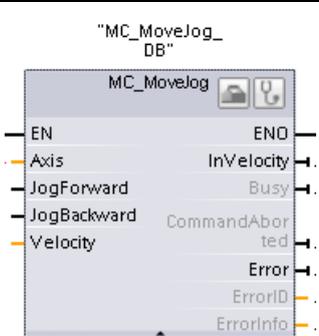
Eine Aufgabe MC_MoveVelocity mit "Velocity" = 0,0 (wie eine Aufgabe MC_Halt) bricht aktive Bewegungssteuerungsaufgaben ab und stoppt die Achse mit der konfigurierten Verzögerung. Wenn die Achse zum Stillstand kommt, gibt der Ausgangsparameter "InVelocity" mindestens einen Programmzyklus lang WAHR an.

"Busy" gibt während des Verzögerungsvorgangs den Wert WAHR an und geht zusammen mit "InVelocity" nach FALSCH. Wenn der Parameter "Execute" = WAHR gesetzt ist, sind "InVelocity" und "Busy" als gespeichert gesetzt.

Wenn die Aufgabe MC_MoveVelocity gestartet wird, wird das Statusbit "SpeedCommand" im Technologieobjekt gesetzt. Das Statusbit "ConstantVelocity" wird beim Stillstand der Achse gesetzt. Beide Bits werden an die neue Situation angepasst, wenn eine neue Bewegungssteuerungsaufgabe gestartet wird.

10.5.9 Anweisung MC_MoveJog (Achse im Tippbetrieb bewegen)

Tabelle 10- 22 Anweisung MC_MoveJog

KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveJog_DB" (Axis:=_multi_fb_in_, JogForward:=_bool_in_, JogBackward:=_bool_in_, Velocity:=_real_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 Tippbetrieb. Diese Anweisung dient üblicherweise zu Test- und Inbetriebnahmezwecken.</p> <p>Um die Anweisung MC_MoveJog zu verwenden, muss die Achse zunächst freigegeben werden.</p>

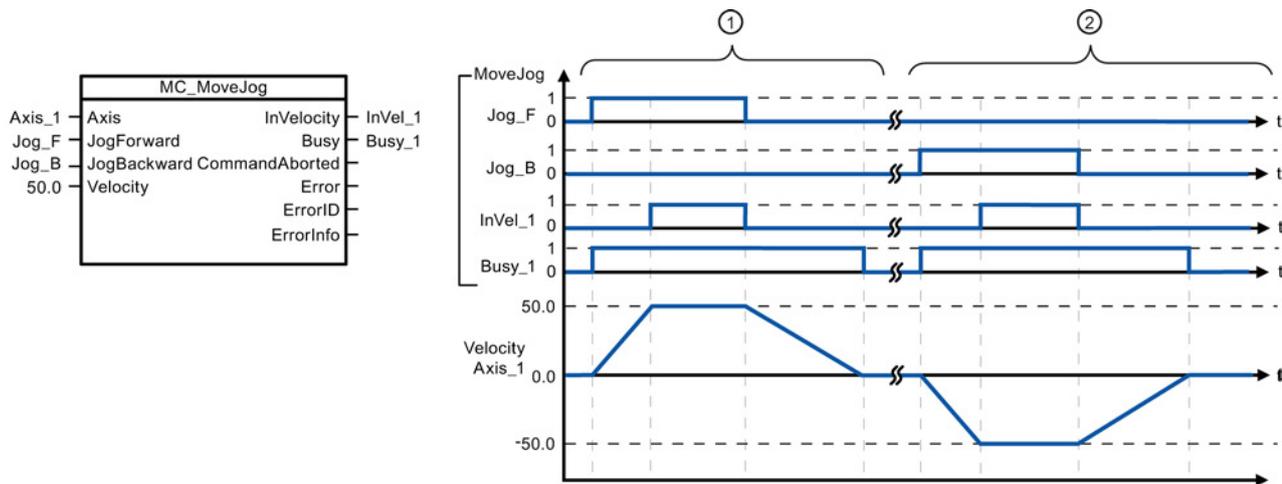
1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

2 Im SCL-Beispiel ist "MC_MoveJog_DB" der Name des Instanz-DBs.

Tabelle 10- 23 Parameter für die MC_MoveJog-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
JogForward ¹	IN	Bool	Solange der Parameter WAHR ist, bewegt sich die Achse mit der im Parameter "Velocity" angegebenen Geschwindigkeit in positiver Richtung. Das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert. (Standardwert: Falsch)
JogBackward ¹	IN	Bool	Solange der Parameter WAHR ist, bewegt sich die Achse mit der im Parameter "Velocity" angegebenen Geschwindigkeit in negativer Richtung. Das Vorzeichen des Werts im Parameter "Velocity" wird ignoriert. (Standardwert: Falsch)
Velocity	IN	Real	Voreingestellte Geschwindigkeit für den Tippbetrieb (Standardwert: 10.0) Grenzwerte: Start-/Stoppgeschwindigkeit $\leq Velocity \leq$ Maximalgeschwindigkeit
InVelocity	OUT	Bool	WAHR = Die im Parameter "Velocity" angegebene Geschwindigkeit wurde erreicht.
Busy	OUT	Bool	WAHR = Die Aufgabe wird ausgeführt.
CommandAborted	OUT	Bool	WAHR = Während der Ausführung wurde die Aufgabe von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUT	Word	Fehler-ID für Parameter "Error" (Standardwert: 0000)
ErrorInfo	OUT	Word	Fehlerinfo-ID für Parameter "ErrorID" (Standardwert: 0000)

¹ Wenn beide Parameter JogForward und JogBackward gleichzeitig WAHR sind, stoppt die Achse mit der konfigurierten Verzögerung. Ein Fehler wird in den Parametern "Error", "ErrorID" und "ErrorInfo" gemeldet.



Die folgenden Werte wurden im Konfigurationsfenster "Dynamik > Allgemein" konfiguriert: Beschleunigung = 10,0 und Verzögerung = 5,0

- ① Die Achse wird im Tippbetrieb über "Jog_F" in positiver Richtung bewegt. Wenn die Zielgeschwindigkeit 50,0 erreicht ist, wird dies über "InVelo_1" gemeldet. Die Achse bremst erneut bis zum Stillstand, nachdem Jog_F zurückgesetzt wird.
- ② Die Achse wird im Tippbetrieb über "Jog_B" in negativer Richtung bewegt. Wenn die Zielgeschwindigkeit 50,0 erreicht ist, wird dies über "InVelo_1" gemeldet. Die Achse bremst erneut bis zum Stillstand, nachdem Jog_B zurückgesetzt wird.

Übersteuerungsantwort

Die Aufgabe MC_MoveJog kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

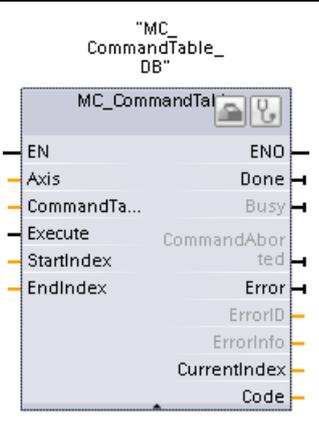
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

Die neue Aufgabe MC_MoveJog bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

10.5.10 Anweisung MC_CommandTable (Achssteuerungsbefehle als Bewegungsfolge ausführen)

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_, StartIndex:= uint_in_, EndIndex:= uint_in_, Done=> bool_out_, Busy=> bool_out_, CommandAborted=> bool_out_, Error=> bool_out_, ErrorID=> word_out_, ErrorInfo=> word_out_, CurrentIndex=> uint_out_, Code=> word_out_);</pre>	<p>Führt eine Reihe einzelner Bewegungen für eine Motorsteuerungsachse 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>

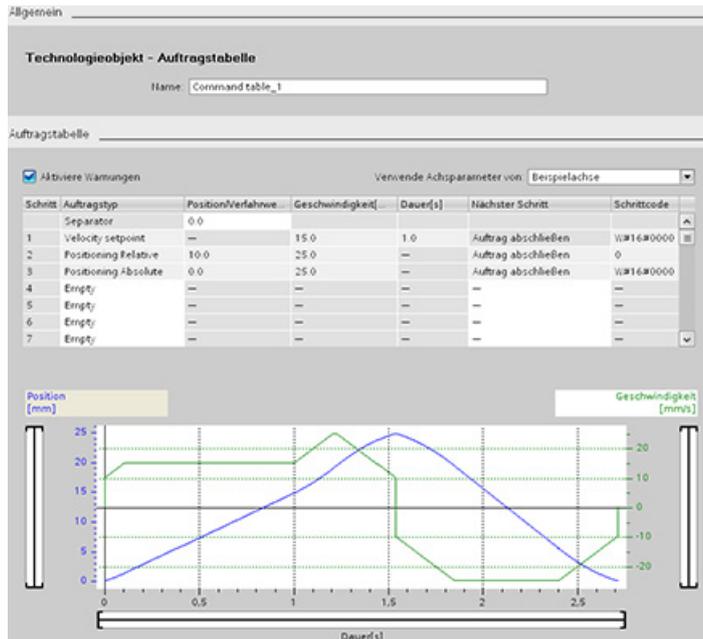
- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_CommandTable_DB" der Name des Instanz-DBs.

Tabelle 10- 25 Parameter für die MC_CommandTable-Anweisung

Parameter und Datentyp	Datentyp	Anfangswert	Beschreibung	
Axis	IN	TO_Axis_1	-	Technologieobjekt "Achse"
Table	IN	TO_CommandTable_1	-	Technologieobjekt "Befehlstabelle"
Execute	IN	Bool	FALSCH	Auftrag bei steigender Flanke starten
StartIndex	IN	Int	1	Verarbeitung der Befehlstabelle mit diesem Schritt starten Grenzwerte: $1 \leq \text{StartIndex} \leq \text{EndIndex}$
EndIndex	IN	Int	32	Verarbeitung der Befehlstabelle mit diesem Schritt beenden Grenzwerte: $\text{StartIndex} \leq \text{EndIndex} \leq 32$
Done	OUT	Bool	FALSCH	Verarbeitung von MC_CommandTable erfolgreich ausgeführt
Busy	OUT	Bool	FALSCH	Operation in Bearbeitung
CommandAborted	OUT	Bool	FALSCH	Die Aufgabe wurde während der Bearbeitung von einer anderen Aufgabe abgebrochen.
Error	OUT	Bool	FALSCH	Ein Fehler ist während der Bearbeitung aufgetreten. Die Ursache wird von den Parametern ErrorID und ErrorInfo. angegeben
ErrorID	OUT	Word	16#0000	Fehler-ID
ErrorInfo	OUT	Word	16#0000	Fehlerinformation

Parameter und Datentyp	Datentyp	Datentyp	Anfangswert	Beschreibung
Step	OUT	Int	0	Schritt wird gerade bearbeitet
Code	OUT	Word	16#0000	Anwenderdefinierte Kennung des in Bearbeitung befindlichen Schritts

Die gewünschte Bewegungsfolge können Sie im Konfigurationsfenster "Befehlstabelle" erstellen und das Ergebnis in der grafischen Darstellung im Kurvendiagramm prüfen.



Sie können die Befehlstypen auswählen, die für die Verarbeitung der Befehlstabelle verwendet werden sollen. Bis zu 32 Aufträge können eingegeben werden. Die Befehle werden der Reihe nach verarbeitet.

Tabelle 10- 26 MC_CommandTable-Befehlstypen

Befehlstyp	Beschreibung
Empty	Dieser Befehl dient als Platzhalter für jeden hinzuzufügenden Befehl. Der leere Eintrag wird bei der Verarbeitung der Befehlstabelle ignoriert.
Halt	Der Befehl hält die Achse an. Hinweis: Der Befehl wird nur nach einem Befehl "Velocity setpoint" durchgeführt.
Positioning Relative	Der Befehl positioniert die Achse basierend auf der Distanz. Er bewegt die Achse um die angegebene Distanz und Geschwindigkeit.
Positioning Absolute	Der Befehl positioniert die Achse basierend auf der Lage. Er bewegt die Achse mit der vorgegebenen Geschwindigkeit an die angegebene Position.
Velocity setpoint	Der Befehl bewegt die Achse mit der angegebenen Geschwindigkeit.
Wait	Der Befehl wartet, bis der angegebene Zeitraum abgelaufen ist. "Wait" stoppt keine aktive Verfahr-bewegung.
Separator	Der Befehl fügt eine Trennlinie ("Separator") oberhalb der ausgewählten Linie ein. Die Trennlinie ermöglicht die Definition mehrerer Profile in einer Befehlstabelle.

Voraussetzungen für die Ausführung von MC_CommandTable:

- Das Technologieobjekt TO_Axis_PTO V2.0 muss ordnungsgemäß konfiguriert sein.
- Das Technologieobjekt TO_CommandTable_PTO muss korrekt konfiguriert sein.
- Die Achse muss freigegeben sein.

Übersteuerungsantwort

Die Aufgabe MC_CommandTable kann von den folgenden Bewegungssteuerungsaufgaben abgebrochen werden:

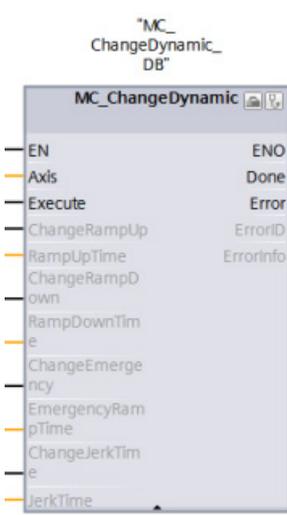
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable

Die neue Aufgabe MC_CommandTable bricht die folgenden aktiven Bewegungssteuerungsaufgaben ab:

- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog
- MC_CommandTable
- Der aktuelle Bewegungssteuerungsauftrag mit dem Start des ersten Befehls "Positioning Relative", "Positioning Absolute", "Velocity setpoint" oder "Halt"

10.5.11 Anweisung MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern)

Tabelle 10- 27 Anweisung MC_ChangeDynamic

KOP/FUP	SCL	Beschreibung
	<pre>"MC_ChangeDynamic_DB" (Execute:=_bool_in_, ChangeRampUp:=_bool_in_, RampUpTime:=_real_in_, ChangeRampDown:=_bool_in_, RampDownTime:=_real_in_, ChangeEmergency:=_bool_in_, EmergencyRampTime:=_real_in_, ChangeJerkTime:=_bool_in_, JerkTime:=_real_in_, Done=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</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.

Tabelle 10- 28 Parameter für die MC_ChangeDynamic-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Starten des Befehls bei einer positiven Flanke. Standardwert: FALSE
ChangeRampUp	IN	Bool	TRUE = Hochlaufzeit in Übereinstimmung mit dem Eingangsparameter "RampUpTime" ändern. Standardwert: FALSE
RampUpTime	IN	Real	Zeit (in Sekunden), während der ohne Ruckbegrenzung vom Stillstand zur konfigurierten Maximalgeschwindigkeit beschleunigt werden soll. Standardwert: 5.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Acceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variablen gezeigt.
ChangeRampDown	IN	Bool	TRUE = Rücklaufzeit in Übereinstimmung mit dem Eingangsparameter "RampDownTime" ändern. Standardwert: FALSE
RampDownTime	IN	Real	Zeit (in Sekunden), während der die Achse ohne Ruckbegrenzung von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand verzögert werden soll. Standardwert: 5.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Deceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variablen gezeigt.

Parameter und Datentyp		Datentyp	Beschreibung
ChangeEmergency	IN	Bool	TRUE = Not-Aus-Rücklaufzeit in Übereinstimmung mit dem Eingangsparameter "EmergencyRampTime" ändern. Standardwert: FALSE
EmergencyRampTime	IN	Real	Zeit (in Sekunden), während der die Achse im Not-Aus-Modus ohne Ruckbegrenzung von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand verzögert werden soll. Standardwert: 2.00 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.EmergencyDeceleration. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variablen gezeigt.
ChangeJerkTime	IN	Bool	TRUE = Glättungszeit in Übereinstimmung mit dem Eingangsparameter "JerkTime" ändern. Standardwert: FALSE
JerkTime	IN	Real	Glättungszeit (in Sekunden) für die Beschleunigungs- und Verzögerungsrampen der Achse. Standardwert: 0.25 Die Änderung beeinflusst die Variable <Achsenname>. Config.DynamicDefaults.Jerk. Die Wirksamkeit der Änderung wird in der Beschreibung dieser Variablen gezeigt.
Done	OUT	Bool	TRUE = Die geänderten Werte wurden in den Technologiedatenbaustein geschrieben. Die Beschreibung der Variablen wird angezeigt, wenn die Änderung wirksam wird. Standardwert: FALSE
Error	OUT	Bool	TRUE = Während der Ausführung des Befehls trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt. Standardwert: FALSE
ErrorID	OUT	Word	Fehler-ID Standardwert: 16#0000
ErrorInfo	IN	Word	Fehlerinformationen. Standardwert: 16#0000

Voraussetzungen für die Ausführung von MC_ChangeDynamic:

- Das Technologieobjekt TO_Axis_PTO V2.0 muss ordnungsgemäß konfiguriert sein.
- Die Achse muss freigegeben sein.

Übersteuerungsantwort

Ein Befehl MC_ChangeDynamic kann von keinem anderen Bewegungssteuerungsbefehl abgebrochen werden.

Der neue Befehl MC_ChangeDynamic bricht keine aktiven Bewegungssteuerungsaufgaben ab.

Hinweis

Für die Eingangsparameter "RampUpTime", "RampDownTime", "EmergencyRampTime" und "RoundingOffTime" können Werte angegeben werden, die dazu führen, dass die resultierenden Achsenparameter "Beschleunigung", "Verzögerung", "Not-Aus-Verzögerung" und "Ruck" außerhalb der zulässigen Grenzwerte liegen.

Stellen Sie sicher, dass Sie die Parameter MC_ChangeDynamic innerhalb der Grenzwerte der Dynamikkonfiguration des Technologieobjekts "Achse" halten.

10.5.12 Anweisung MC_WriteParam (Parameter des Technologieobjekts schreiben)

Mit der Anweisung MC_WriteParam können Sie bestimmte Parameter schreiben, um die Funktionalität der Achse im Anwenderprogramm zu ändern.

Tabelle 10- 29 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.

Sie können in öffentliche Parameter schreiben. Sie können nicht in "MotionStatus" und "StatusBits" schreiben. Die folgende Tabelle zeigt die gültigen Parameter:

Name des beschreibbaren Parameters	Name des beschreibbaren Parameters
Actor.InverseDirection	DynamicDefaults.Acceleration
Actor.DirectionMode	DynamicDefaults.Deceleration
Actor.DriveParameter.PulsesPerDriveRevolution	DynamicDefaults.Jerk
Sensor[1].ActiveHoming.Mode	DynamicDefaults.EmergencyDeceleration
Sensor[1].ActiveHoming.SidelInput	PositionLimitsHW.Active
Sensor[1].ActiveHoming.Offset	PositionLimitsHW.MaxSwitchedLevel
Sensor[1].ActiveHoming.SwitchedLevel	PositionLimitsHW.MinSwitchedLevel
Sensor[1].PassiveHoming.Mode	PositionLimitsSW.Active
Sensor[1].PassiveHoming.SidelInput	PositionLimitsSW.MinPosition
Sensor[1].PassiveHoming.SwitchedLevel	PositionLimitsSW.MaxPosition
Units.LengthUnit	Homing.AutoReversal
Mechanics.LeadScrew	Homing.ApproachDirection
DynamicLimits.MinVelocity	Homing.ApproachVelocity
DynamicLimits.MaxVelocity	Homing.ReferencingVelocity

Tabelle 10- 30 Parameter für die Anweisung MC_WriteParam

Parameter und Datentyp		Datentyp	Beschreibung
PARAMNAME	IN	Variant	Name des Parameters, in den der Wert geschrieben wird
VALUE	IN	Variant	Wert, der in den zugewiesenen Parameter geschrieben werden soll
EXECUTE	IN	Bool	Anweisung starten. Standardwert: FALSE
DONE	OUT	Bool	Wert wurde geschrieben. Standardwert: FALSE
BUSY	OUT	Bool	Bei TRUE wird die Anweisung gerade ausgeführt. Standardwert: FALSE
ERROR	OUT	Real	Bei TRUE ist ein Fehler aufgetreten. Standardwert: FALSE
ERRORID	OUT	Word	ID des Fehlers
ERRORINFO	OUT	Word	Zusatzinformation zur ERRORID

Tabelle 10- 31 Bedingungscode für ERRORID und ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Beschreibung
0	0	Parameter TO-DB einer Achse wurde erfolgreich geändert
8410 _[1]	0028 _[1]	Ungültiger Parameter (Achsparameter TO-DB mit falscher Länge)
8410 _[1]	0029 _[1]	Ungültiger Parameter (kein Achsparameter TO-DB)
8410 _[1]	002B _[1]	Ungültiger Parameter (Achsparameter TO-DB schreibgeschützt)
8410 _[1]	002C _[1]	Gültiger Parameter, aber Achse nicht deaktiviert
Config Error _[2]	Config Error _[2]	Gültiger Parameter (öffentlicher Achsparameter TO-DB schreibgeschützt) nicht im zulässigen Bereich
Config Error _[3]	Config Error _[3]	Gültiger Parameter (öffentlicher Achsparameter TO-DB) nicht im zulässigen Bereich

[1] Fehler in MC_WriteParam

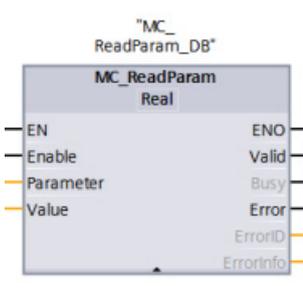
[2] Fehler in MC_Power

[3] Fehler in MC_Power und MC_MoveXXX oder MC_CommandTable

10.5.13 Anweisung MC_ReadParam (Parameter eines 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- 32 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>

- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC_ReadParam_DB" der Name des Instanz-DBs.

Die Anweisung MC_ReadParam funktioniert bei Freigabeverhalten. Solange der Eingang "Enable" wahr ist, liest die Anweisung den angegebenen "Parameter" in den Speicherort "Value".

Der "Position"-Wert der Achse "MotionStatus" wird basierend auf dem aktuellen HSC-Wert an jedem Steuerpunkt im Zyklus aktualisiert.

Der "Velocity"-Wert der Achse "MotionStatus" ist die Befehls-geschwindigkeit am Ende des aktuellen Segments (wird alle ~10 ms aktualisiert). Die Anweisung MC_ReadParam kann auch diesen Wert lesen.

Tritt ein Fehler auf, geht die Anweisung in einen Fehlerzustand, der nur durch eine neue steigende Flanke am Eingang "Enable" zurückgesetzt werden kann.

Tabelle 10- 33 Parameter für die MC_ReadParam-Anweisung

Parameter und Datentyp	Datentyp	Beschreibung	
ENABLE	IN	Bool	Start der Anweisung. Standardwert: FALSE
PARAMETER	IN	Variant	Pointer auf den zu lesenden TO-Parameter
VALID	OUT	Bool	Wenn TRUE, wurde der Wert gelesen. Standardwert: FALSE
BUSY	OUT	Bool	Wenn TRUE, ist die Anweisung in Betrieb. Standardwert: FALSE
ERROR	OUT	Real	Wenn TRUE, ist ein Fehler aufgetreten. Standardwert: FALSE
ERRORID	OUT	Word	ID des Fehlers. Standardwert: 0
ERRORINFO	OUT	Word	Zugehörige Informationen zu ERRORID. Standardwert: 0
VALUE	INOUT	Variant	Pointer auf den Speicherort, an dem der gelesene Wert gespeichert ist

Tabelle 10- 34 Bedingungscode für ERRORID und ERRORINFO

ERRORID (W#16#...)	ERRORINFO (W#16#...)	Beschreibung
0	0	Parameter erfolgreich gelesen
8410	0028	Ungültiger Parameter (falsche Länge)
8410	0029	Ungültiger Parameter (kein TO-DB)
8410	0030	Ungültiger Parameter (nicht lesbar)
8411	0032	Ungültiger Parameter (falscher Wert)

TO-Parameter

Die Achse "MotionStatus" besteht aus vier Werten. Überwachen Sie diese Werte auf Veränderungen, die während der Ausführung des Programms gelesen werden können:

Variablenname	Datentyp	Lesbar über MC_ReadParam
MotionStatus:	Struktur	Nein
• Position	REAL	Ja
• Geschwindigkeit	REAL	Ja
• Entfernung	REAL	Ja
• Zielposition	REAL	Ja

10.6 Funktionsweise der Bewegungssteuerung bei der S7-1200

10.6.1 Für die Bewegungssteuerung verwendete CPU-Ausgänge

Die CPU bietet vier Impulsgeneratoren. Jeder Impulsgenerator bietet einen Impulsausgang und einen Richtungsausgang zum Steuern eines Schrittmotorantriebs oder eines Servomotorantriebs mit Impulsschnittstelle. Der Impulsausgang bietet den Antrieb mit den für die Motorbewegung erforderlichen Impulsen. Der Richtungsausgang steuert die Verfahrrichtung des Antriebs.

Der PTO-Ausgang erzeugt einen Rechteckausgang mit variabler Frequenz. Die Impulserzeugung wird über Konfigurations- und Ausführungsinformationen gesteuert, die über die H/W-Konfiguration und/oder SFCs/SFBs vorgegeben werden.

Während sich die CPU im Betriebszustand RUN befindet, steuern je nach Auswahl des Anwenders entweder die Werte im Prozessabbild oder die Ausgänge des Impulsgenerators die digitalen Ausgänge. Im Betriebszustand STOP steuert der PTO-Generator die Ausgänge nicht.

Integrierte CPU-Ausgänge und Ausgänge eines Signalboards können als Impuls- und Richtungsausgänge verwendet werden. Sie wählen zwischen integrierten CPU-Ausgängen und Ausgängen des Signalboards während der Gerätekonfiguration der Impulsgeneratoren (PTO/PWM) im Register "Eigenschaften". Nur PTO (Pulse Train Output) gilt für die Bewegungssteuerung.

Die folgende Tabelle zeigt die E/A-Standardbelegungen; die vier Impulsgeneratoren können jedoch für jeden Digitalausgang konfiguriert werden.

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 Bewegungssteuerungsanweisungen) konfigurieren, steuern die entsprechenden Ausgangsadressen nicht mehr die Ausgänge. 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 eine anderweitige Nutzung in Ihrem Programm freigegeben werden.

Jeder PTO müssen zwei Ausgänge zugewiesen werden: ein Impulsausgang und ein Richtungsausgang. Es ist möglich, nur den Impulsausgang, aber nicht den Richtungsausgang zu verwenden. Dadurch wird der Richtungsausgang für andere Funktionen im Anwenderprogramm freigegeben. Der Ausgang kann nicht gleichzeitig als PTO-Richtungsausgang und im Anwenderprogramm benutzt werden.

Tabelle 10- 35 Standardadresszuweisungen für die Impuls- und Richtungsausgänge

Verwendung von Ausgängen für die Bewegungssteuerung		
	Impuls	Richtung
PTO1		
Integrierte E/A	A0.0	A0.1
SB-E/A	A4.0	A4.1
PTO2		
Integrierte E/A	A0.2	A0.3
SB-E/A	A4.2	A4.3
PTO3		
Integrierte E/A	A0.4 ¹	A0.5 ¹
SB-E/A	A4.0	A4.1
PTO4		
Integrierte E/A	A0.6 ²	A0.7 ²
SB-E/A	A4.2	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-Funktionen der CPU 1211C, CPU 1212C, CPU 1214C, CPU 1215C und CPU 1217C.

Antriebsschnittstelle

Für die Bewegungssteuerung können Sie optional für eine Antriebsschnittstelle die Zustände "Antrieb freigegeben" und "Antrieb bereit" konfigurieren. Wenn Sie die Antriebsschnittstelle verwenden, können der Digitalausgang für "Antrieb freigegeben" und der Digitaleingang für "Antrieb bereit" frei ausgewählt werden.

Hinweis

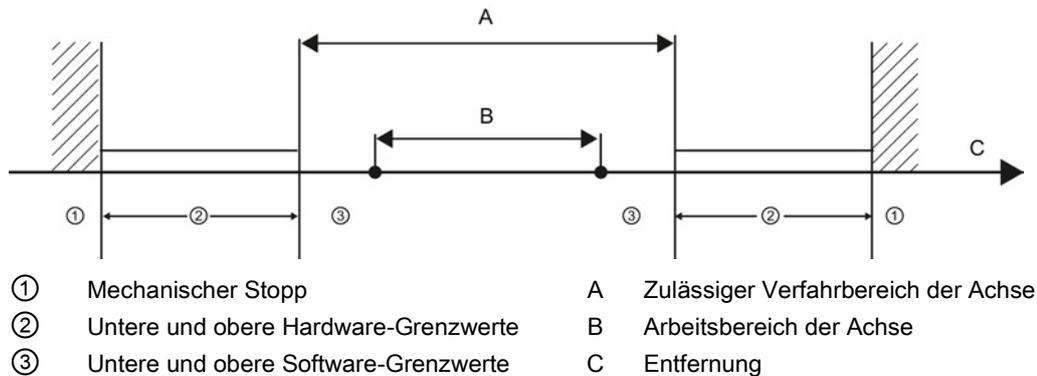
Die Firmware übernimmt die Steuerung über die entsprechenden Impuls- und Richtungsausgänge, wenn die PTO (Pulse Train Output) ausgewählt und einer Achse zugewiesen wurde.

Mit dieser Übernahme der Steuerungsfunktion wird auch die Verbindung zwischen dem Prozessabbild und dem E/A-Ausgang getrennt. Der Anwender hat zwar die Möglichkeit, über das Anwenderprogramm oder die Beobachtungstabelle in das Prozessabbild der Impuls- und Richtungsausgänge zu schreiben, diese Daten werden jedoch nicht an die Ausgänge übertragen. Entsprechend ist es auch nicht möglich, die Ausgänge über das Anwenderprogramm oder die Beobachtungstabelle zu überwachen. Die ausgelesenen Informationen geben lediglich den Wert des Prozessabbilds wieder und entsprechen in keiner Weise dem tatsächlichen Zustand der Ausgänge.

Bei allen anderen CPU-Ausgängen, die nicht permanent von der CPU-Firmware verwendet werden, kann der Zustand der Ausgänge wie üblich über das Prozessabbild gesteuert und überwacht werden.

10.6.2 Hardware- und Software-Endschalter für die Bewegungssteuerung

Mit den Hardware- und Software-Endschaltern begrenzen Sie den zulässigen Verfahrbereich und den Arbeitsbereich Ihrer Achse.



Hardware- und Software-Endschalter müssen vor der Verwendung in der Konfiguration oder im Anwenderprogramm aktiviert werden. Software-Endschalter sind erst nach der Referenzpunktfahrt der Achse aktiv.

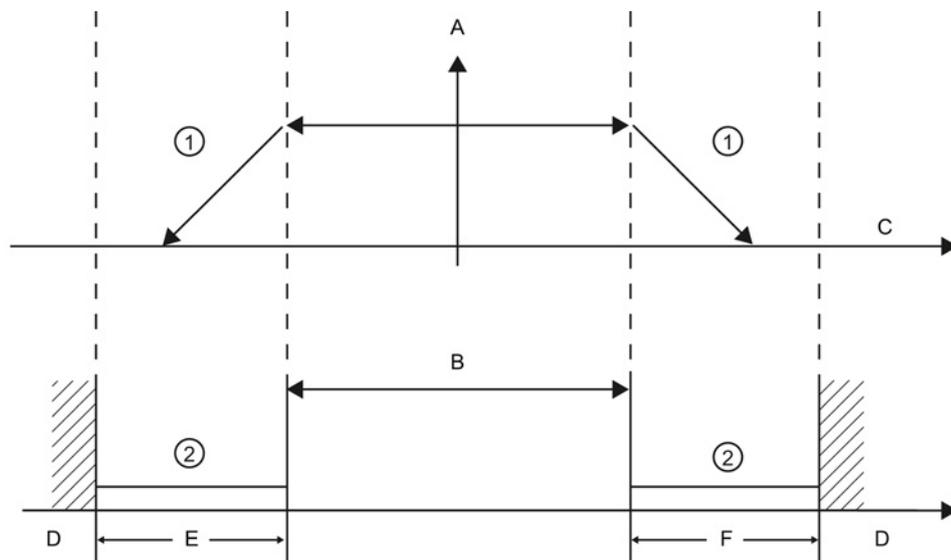
Hardware-Endschalter

Hardware-Endschalter geben den maximalen Verfahrbereich der Achse vor. Hardware-Endschalter sind physikalische Schaltelemente, die an alarmfähige Eingänge der CPU angeschlossen werden müssen. Verwenden Sie nur Hardware-Endschalter, die nach der Anfahrt dauerhaft geschaltet bleiben. Der Schaltzustand kann erst nach der Rückkehr in den zulässigen Verfahrbereich rückgängig gemacht werden.

Tabelle 10- 36 Verfügbare Eingänge für Hardwaregrenzen

Beschreibung	RPS	LIM-	LIM+
Integrierte E/A		E0.0 - E1.5	
SB-E/A		E4.0 - E4.3	

Wenn die Hardware-Endschalter angefahren werden, bremst die Achse mit der konfigurierten Notfallverzögerung bis zum Stillstand. Die angegebene Notfallverzögerung muss ausreichend sein, um die Achse vor dem mechanischen Stopp zuverlässig zu stoppen. Das folgende Diagramm zeigt das Verhalten der Achse nach Anfahrt der Hardware-Endschalter.



- ① Die Achse brems mit der konfigurierten Notfallverzögerung bis zum Stillstand.
 ② Bereich, in dem die Hardware-Endschalter den Zustand "angefahren" signalisieren.
 A [Geschwindigkeit]
 B Zulässiger Verfahrbereich
 C Entfernung
 D Mechanischer Stopp
 E Unterer Hardware-Endschalter
 F Oberer Hardware-Endschalter



WARNUNG

Gefahren beim Ändern der Filterzeit 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.

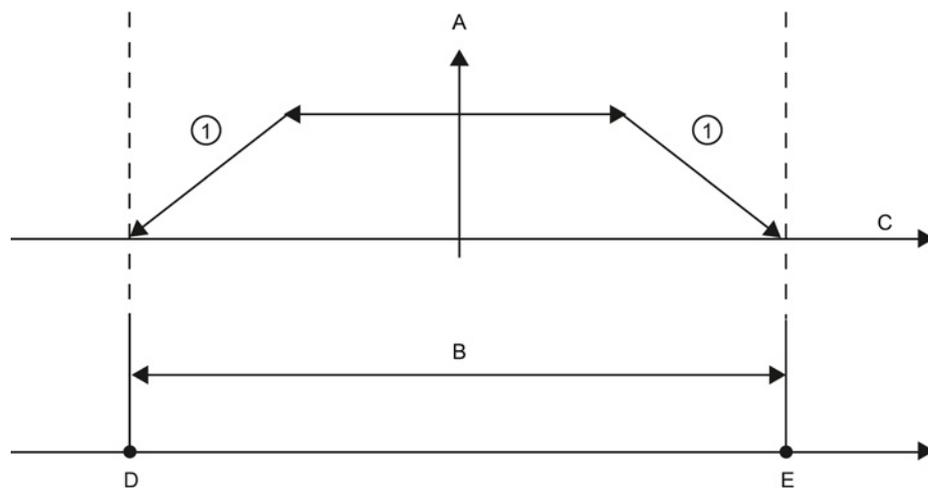
Das Verändern der Filterzeiten kann unerwarteten Betrieb der Maschine bzw. des Prozesses verursachen, was zu tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann.

Um sicherzustellen, dass eine neue Filterzeit sofort wirksam wird, müssen Sie die CPU aus- und wieder einschalten.

Software-Endschalter

Software-Endschalter begrenzen den Arbeitsbereich der Achse. Sie sollten sich relativ zum Verfahrbereich innerhalb der Hardware-Endschalter befinden. Weil die Positionen der Software-Endschalter flexibel eingerichtet werden können, lässt sich der Arbeitsbereich der Achse individuell je nach aktuellem Verfahrprofil einschränken. Im Gegensatz zu Hardware-Endschaltern werden Software-Endschalter ausschließlich über die Software implementiert und benötigen keine eigenen Schaltelemente.

Wenn Software-Endschalter aktiviert werden, wird eine aktive Bewegung an der Position des Software-Endschalters gestoppt. Die Achse wird mit der konfigurierten Verzögerung gebremst. Das folgende Diagramm zeigt das Verhalten der Achse bis zum Erreichen des Software-Endschalters.



- ① Die Achse bremst mit der konfigurierten Verzögerung bis zum Stillstand.
- A [Geschwindigkeit]
- B Arbeitsbereich
- C Entfernung
- D Unterer Software-Endschalter
- E Oberer Software-Endschalter

Verwenden Sie zusätzlich Hardware-Endschalter, wenn sich ein mechanischer Endstopp nach den Software-Endschaltern befindet und die Gefahr mechanischer Schäden besteht.

Weitere Informationen

Ihr Anwenderprogramm kann die Grenzwerte der Hardware- oder Softwareposition durch Aktivieren oder Deaktivieren der Hardware- und Software-Grenzwertfunktionen übersteuern. Die Auswahl erfolgt über den Achsen-DB.

- Um die Hardware-Grenzwertfunktion zu aktivieren, rufen Sie die Variable "Active" (Bool) im DB-Pfad "<Achsenname>/Config/PositonLimits_HW" auf. Der Zustand der Variable "Active" aktiviert oder deaktiviert die Verwendung von Hardware-Positionsgrenzwerten.
- Um die Software-Grenzwertfunktion zu aktivieren, rufen Sie die Variable "Active" (Bool) im DB-Pfad "<Achsenname>/Config/Position Limits_SW" auf. Der Zustand der Variable "Active" aktiviert oder deaktiviert die Verwendung von Software-Positionsgrenzwerten.

Sie können die Software-Positionsgrenzwerte auch mit Ihrem Anwenderprogramm ändern (beispielsweise um die Maschineneinrichtung flexibler zu gestalten oder um die Umrüstzeiten von Maschinen zu verkürzen). Ihr Anwenderprogramm kann neue Werte in die Variablen "MinPosition" und "MaxPosition" (physikalische Einheiten im Format Real) im DB "<Achsenname>/Config/PositionLimits_SW" schreiben.

10.6.3 Referenzpunktfahrt

10.6.3.1 Referenzpunktfahrt für Achsen durchführen

Bei der Referenzpunktfahrt werden die Achsenkoordinaten an die reale, physikalische Position des Antriebs angepasst. (Befindet sich der Antrieb gegenwärtig an Position x, wird die Achse in Position x gebracht.) Bei positionsgesteuerten Achsen beziehen sich die Einträge und Anzeigen für die Position exakt auf diese Achsenkoordinaten.

Hinweis

Die Übereinstimmung zwischen den Achsenkoordinaten und der realen Situation ist äußerst wichtig. Dieser Schritt ist erforderlich, um sicherzustellen, dass die absolute Zielposition der Achse auch exakt mit dem Antrieb erreicht wird.

Die Anweisung MC_Home löst die Referenzpunktfahrt der Achse aus.

Es gibt vier verschiedene Funktionen für die Referenzpunktfahrt. Die ersten beiden Funktionen ermöglichen es dem Anwender, die aktuelle Position der Achse einzustellen, und die beiden zweiten positionieren die Achse in Bezug auf einen Referenzpunktsensor.

- Betriebsart 0 - Direkte Referenzpunktfahrt absolut: Wenn diese Betriebsart ausgeführt wird, wird der Achse genau mitgeteilt, wo sie sich befindet. Die interne Positionsvariable wird auf den Wert des Positionseingangs der Anweisung für die Referenzpunktfahrt gesetzt. Dies wird bei der Kalibrierung und Einrichtung von Maschinen verwendet.

Die Achsenposition wird unabhängig vom Referenzpunktschalter gesetzt. Aktive Verfahrbewegungen werden nicht abgebrochen. Der Wert des Eingangsparameters Position der Anweisung MC_Home wird sofort als Referenzpunkt der Achse gesetzt. Um den Referenzpunkt einer genauen mechanischen Position zuzuweisen, muss sich die Achse zum Zeitpunkt der Referenzpunkteinstellung an dieser Position im Stillstand befinden.

- Betriebsart 1 - Direkte Referenzpunktfahrt relativ: Diese Betriebsart nutzt bei der Ausführung die interne Positionsvariable und fügt den Wert des Positionseingangs der Anweisung für die Referenzpunktfahrt in diese Variable ein. Dies wird typischerweise genutzt, um den Maschinenversatz zu berücksichtigen.

Die Achsenposition wird unabhängig vom Referenzpunktschalter gesetzt. Aktive Verfahrbewegungen werden nicht abgebrochen. Die folgende Aussage gilt für die Achsenposition nach der Referenzpunktfahrt: Neue Achsenposition = aktuelle Achsenposition + Wert des Parameters Position der Anweisung MC_Home.

- Betriebsart 2 - Passive Referenzpunktfahrt: Wenn sich die Achse bewegt und den Referenzpunktschalter überfährt, wird die aktuelle Position als Referenzpunkt gesetzt. Bei dieser Funktion wird der normale Maschinenverschleiß und das Zahnflankenspiel berücksichtigt, um den Bedarf an manuellem Verschleißausgleich zu verhindern. Der Positionseingang der Anweisung für die Referenzpunktfahrt wird wie zuvor zur vom Referenzpunktschalter angegebenen Position addiert, um den Referenzpunkt mühelos zu versetzen.

Während der passiven Referenzpunktfahrt führt die Anweisung MC_Home keine Referenzpunktfahrtbewegung durch. Die für diesen Schritt erforderliche Verfahrbewegung müssen Sie über andere Bewegungssteuerungsanweisungen implementieren. Wenn der Referenzpunktschalter erkannt wird, wird die Achse entsprechend der Konfiguration an den Referenzpunkt gefahren. Aktive Verfahrbewegungen werden beim Start der passiven Referenzpunktfahrt nicht abgebrochen.

- Betriebsart 3 - Aktive Referenzpunktfahrt: Diese Betriebsart ist das präziseste Verfahren für die Referenzpunktfahrt der Achse. Die anfängliche Richtung und Geschwindigkeit der Bewegung werden in den erweiterten Parametern in der Konfiguration des Technologieobjekts unter Referenzpunktfahrt konfiguriert. Dies ist abhängig von der Maschinenkonfiguration. Zudem kann festgelegt werden, ob die steigende oder fallende Flanke des Signals des Referenzpunktschalters der Referenzpunkt ist. Praktisch alle Sensoren haben einen aktiven Bereich, und wenn die Position "Steady State On" als Referenzpunktsignal verwendet wurde, besteht die Möglichkeit eines Fehlers beim Referenzpunkt, weil der aktive Bereich des EIN-Signals einen Entfernungsbereich abdeckt. Durch Verwendung der steigenden oder fallenden Flanke dieses Signals resultiert ein sehr viel präziserer Referenzpunkt. Wie bei allen anderen Betriebsarten wird der Wert des Positioneingangs der Anweisung für die Referenzpunktfahrt zum Hardware-Referenzpunkt addiert.

Bei der aktiven Referenzpunktfahrt führt die Anweisung MC_Home die erforderliche Referenzpunktfahrt durch. Wenn der Referenzpunktschalter erkannt wird, wird die Achse entsprechend der Konfiguration an den Referenzpunkt gefahren. Aktive Verfahrbewegungen werden abgebrochen.

Bei den Betriebsarten 0 und 1 muss die Achse überhaupt nicht bewegt werden. Sie werden typischerweise bei der Einrichtung und Kalibrierung verwendet. Bei den Betriebsarten 2 und 3 muss die Achse bewegt werden und einen Sensor überfahren, der im Technologieobjekt "Achse" als Referenzpunktschalter konfiguriert ist. Der Referenzpunkt kann im Arbeitsbereich der Achse oder außerhalb des normalen Arbeitsbereichs der Achse, muss jedoch innerhalb des Bewegungsbereichs platziert werden.

10.6.3.2 Konfiguration der Parameter für die Referenzpunktfahrt

Sie konfigurieren die Parameter für aktive und passive Referenzpunktfahrt im Konfigurationsfenster "Referenzpunktfahrt". Das Verfahren der Referenzpunktfahrt wird über den Eingangsparameter "Mode" der Bewegungssteuerungsanweisung festgelegt. Hier bedeutet Mode = 2 passive Referenzpunktfahrt und Mode = 3 bedeutet aktive Referenzpunktfahrt.

Hinweis

Stellen Sie anhand einer der folgenden Maßnahmen sicher, dass die Maschine bei Richtungswechsel nicht zu einem mechanischen Endstopp fährt:

- Halten Sie die Anfahrtschwindigkeit gering
 - Vergrößern Sie die konfigurierte Beschleunigung/Verzögerung
 - Vergrößern Sie die Entfernung zwischen Hardware-Endschalter und mechanischem Stopp
-

Tabelle 10- 37 Konfigurationsparameter für die Referenzpunktfahrt der Achse

Parameter	Beschreibung
Eingang Referenzpunktschalter (Aktive und passive Referenzpunktfahrt)	<p>Wählen Sie den digitalen Eingang für den Referenzpunktschalter in der Klappliste aus. Der Eingang muss alarmfähig sein. Die integrierten Eingänge der CPU und die Eingänge eines gesteckten Signalboards können als Eingänge für den Referenzpunktschalter ausgewählt werden.</p> <p>Die Standardfilterzeit für die digitalen Eingänge beträgt 6,4 ms. Wenn die digitalen Eingänge als Referenzpunktschalter verwendet werden, kann dies zu unerwünschten Verzögerungen und damit zu Ungenauigkeiten führen. Abhängig von der verringerten Geschwindigkeit und dem Ausmaß des Referenzpunktschalters wird der Referenzpunkt möglicherweise nicht erkannt. Die Filterzeit kann in der Gerätekonfiguration der digitalen Eingänge unter "Eingangsfiler" eingestellt werden.</p> <p>Die angegebene Filterzeit muss kleiner als die Dauer des Eingangssignals am Referenzpunktschalter sein.</p>
Automatische Umkehr nach Erreichen der Hardware-Endschalter (Nur bei aktiver Referenzpunktfahrt)	<p>Aktivieren Sie das Kontrollkästchen, um den Hardware-Endschalter als Umkehrnocken für die Referenzpunktfahrt zu nutzen. Die Hardware-Endschalter müssen für die Richtungskehr konfiguriert und aktiviert sein.</p> <p>Wenn der Hardware-Endschalter während der aktiven Referenzpunktfahrt erreicht wird, bremst die Achse mit der konfigurierten Verzögerung (nicht mit der Notfallverzögerung) und kehrt die Richtung um. Der Referenzpunktschalter wird dann in umgekehrter Richtung erfasst.</p> <p>Wenn die Richtungskehr nicht aktiv ist und die Achse den Hardware-Endschalter während der aktiven Referenzpunktfahrt erreicht, wird die Referenzpunktfahrt mit einem Fehler abgebrochen und die Achse mit der Notfallverzögerung gebremst.</p>
Anfahrriechung (Aktive und passive Referenzpunktfahrt)	<p>Bei der Richtungsauwahl legen Sie die "Anfahrriechung" für die aktive Referenzpunktfahrt zum Suchen des Referenzpunktschalters sowie die Richtung der Referenzpunktfahrt fest. Die Richtung der Referenzpunktfahrt legt die Verfahrriechung der Achse zum Anfahren der konfigurierten Seite des Referenzpunktschalters fest, um die Referenzpunktfahrt durchzuführen.</p>
Referenzpunktschalter (Aktive und passive Referenzpunktfahrt)	<ul style="list-style-type: none"> • Aktive Referenzpunktfahrt: Geben Sie an, ob die Achse auf der linken oder rechten Seite des Referenzpunktschalters referenziert werden soll. Je nach Startposition der Achse und der Konfiguration der Parameter für die Referenzpunktfahrt kann sich die Abfolge der Referenzpunktfahrt vom Diagramm im Konfigurationsfenster unterscheiden. • Passive Referenzpunktfahrt: Bei der passiven Referenzpunktfahrt müssen die Verfahrbewegungen zum Zweck der Referenzpunktfahrt vom Anwender über Bewegungssteuerungsbefehle implementiert werden. Die Seite des Referenzpunktschalters, auf der die Referenzpunktfahrt erfolgt, ist von den folgenden Faktoren abhängig: <ul style="list-style-type: none"> – Konfigurierte Anfahrriechung – Konfigurierter Referenzpunktschalter – Aktuelle Verfahrriechung während der passiven Referenzpunktfahrt
Anfahrriechwindigkeit (Nur bei aktiver Referenzpunktfahrt)	<p>Geben Sie die Geschwindigkeit an, mit der der Referenzpunktschalter während der Referenzpunktfahrt gesucht werden soll.</p> <p>Grenzwerte (unabhängig von der ausgewählten Anwendereinheit): Start-/Stoppriechwindigkeit ≤ Anfahrriechwindigkeit ≤ Maximalriechwindigkeit</p>

Parameter	Beschreibung
Verringerte Geschwindigkeit (Nur bei aktiver Referenzpunktfahrt)	Geben Sie die Geschwindigkeit an, mit der die Achse den Referenzpunktschalter für die Referenzpunktfahrt anfährt. Grenzwerte (unabhängig von der ausgewählten Anwendereinheit): Start-/Stoppgeschwindigkeit \leq verringerte Geschwindigkeit \leq Maximalgeschwindigkeit
Referenzpunktversatz (Nur bei aktiver Referenzpunktfahrt)	Wenn der gewünschte Referenzpunkt von der Position des Referenzpunktschalters abweicht, kann der Referenzpunktversatz in diesem Feld angegeben werden. Ist der Wert ungleich 0, führt die Achse die folgenden Vorgänge nach der Referenzpunktfahrt am Referenzpunktschalter durch: <ol style="list-style-type: none"> 1. Bewegen der Achse mit verringerter Geschwindigkeit um den Wert des Referenzpunktversatzes. 2. Wenn die Position des Referenzpunkts erreicht ist, wird die Achsenposition auf den absoluten Referenzpunkt gesetzt. Der absolute Referenzpunkt wird über den Parameter "Position" der Bewegungssteuerungsanweisung "MC_Home" angegeben. Grenzwerte (unabhängig von der ausgewählten Anwendereinheit): $-1,0e12 \leq$ Referenzpunktversatz $\leq 1,0e12$

Tabelle 10- 38 Die Referenzpunktfahrt beeinflussende Faktoren

Beeinflussende Faktoren:			Ergebnis:
Konfiguration Anfahrrichtung	Konfiguration Referenzpunktschalter	Aktuelle Verfahrrichtung	Referenzpunktfahrt zu Referenzpunktschalter
Positiv	"Linke (negative) Seite"	Positive Richtung	Links
		Negative Richtung	Rechts
Positiv	"Rechte (positive) Seite"	Positive Richtung	Rechts
		Negative Richtung	Links
Negativ	"Linke (negative) Seite"	Positive Richtung	Rechts
		Negative Richtung	Links
Negativ	"Rechte (positive) Seite"	Positive Richtung	Links
		Negative Richtung	Rechts

10.6.3.3 Abfolge bei der aktiven Referenzpunktfahrt

Sie aktive Referenzpunktfahrt starten Sie mit der Bewegungssteuerungsanweisung "MC_Home" (Eingangsparameter Mode = 3). Der Eingangsparameter "Position" gibt in diesem Fall die absoluten Koordinaten des Referenzpunkts an. Alternativ können Sie die aktive Referenzpunktfahrt zu Testzwecken im Steuerpanel starten.

Das folgende Diagramm zeigt ein Beispiel einer Kennlinie für eine aktive Referenzpunktfahrt mit den folgenden Konfigurationsparametern:

- "Anfahrrichtung" = "Positive Anfahrrichtung"
- "Referenzpunktschalter" = "Rechte (positive) Seite"
- Wert des "Referenzpunktversatzes" > 0

Tabelle 10- 39 Geschwindigkeitskennlinie für die Referenzpunktfahrt bei der Bewegungssteuerung

Funktionsweise		Hinweise	
		A	Anfahrgeschwindigkeit
		B	Verringerte Geschwindigkeit
		C	Referenzpunktkoordinaten
		D	Referenzpunktversatz
①	Suchphase (blaues Kennliniensegment): Wenn die aktive Referenzpunktfahrt startet, beschleunigt die Achse auf die konfigurierte "Anfahrgeschwindigkeit" und sucht bei dieser Geschwindigkeit nach dem Referenzpunktschalter.		
②	Referenzpunktfahrt (roter Kennlinienabschnitt): Wenn der Referenzpunktschalter erkannt ist, bremst die Achse in diesem Beispiel und kehrt um, um auf der konfigurierten Seite des Referenzpunktschalters bei der konfigurierten "verringerten Geschwindigkeit" an den Referenzpunkt zu fahren.		
③	Verfahren zum Referenzpunkt (grünes Kennliniensegment): Nach der Referenzpunktfahrt am Referenzpunktschalter verfährt die Achse bei "verringerten Geschwindigkeit" zu den "Referenzpunktkoordinaten". Beim Erreichen der "Referenzpunktkoordinaten" wird die Achse an dem Positionswert gestoppt, der im Eingangsparameter Position der Anweisung MC_Home angegeben wurde.		

Hinweis

Wenn die Referenzpunktsuche nicht wie erwartet funktioniert, prüfen Sie die den Hardware-Grenzwerten oder dem Referenzpunkt zugewiesenen Eingänge. Möglicherweise wurden die Flankenalarme für diese Eingänge in der Gerätekonfiguration deaktiviert.

Prüfen Sie die Konfigurationsdaten des betroffenen Technologieobjekts "Achse", um festzustellen, welche Eingänge ggf. zu "HW Low Limit Switch Input", "HW High Limit Switch Input" und "Input reference point switch" zugewiesen sind. Öffnen Sie dann die Gerätekonfiguration der CPU und untersuchen Sie jeden der zugewiesenen Eingänge. Prüfen Sie, ob "Erkennung steigende Flanke aktivieren" und "Erkennung fallende Flanke aktivieren" beide ausgewählt sind. Sind diese Eigenschaften nicht ausgewählt, löschen Sie die angegebenen Eingänge in der Achsenkonfiguration und wählen sie erneut aus.

10.7 Inbetriebnahme

Diagnosefunktion "Status- und Fehlerbits"

Mit der Diagnosefunktion "Status- und Fehlerbits" können Sie die wichtigsten Status- und Fehlermeldungen der Achse überwachen. Die Diagnosefunktionsanzeige ist im Online-Modus bei aktiver Achse in der Betriebsart "Manuelle Steuerung" und "Automatiksteuerung" verfügbar.

Tabelle 10- 40 Status der Achse

Status	Beschreibung
Freigegeben	Die Achse ist freigegeben und bereit, über Bewegungssteuerungsaufgaben gesteuert zu werden. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Enable)
Referenzpunktfahrt durchgeführt	Die Achse ist am Referenzpunkt und kann absolute Positionierungsaufgaben der Bewegungssteuerungsanweisung MC_MoveAbsolute durchführen. Die Achse muss sich für die relative Referenzpunktfahrt nicht am Referenzpunkt befinden. Besondere Situationen: <ul style="list-style-type: none"> • Während der aktiven Referenzpunktfahrt ist der Status FALSCH. • Wenn für eine Achse, die sich am Referenzpunkt befindet, eine passive Referenzpunktfahrt durchgeführt wird, wird der Status während der passiven Referenzpunktfahrt auf WAHR gesetzt. (Variable des Technologieobjekts: <Achsenname>.StatusBits.HomingDone)
Fehler	Im Technologieobjekt "Achse" ist ein Fehler aufgetreten. Weitere Informationen zum Fehler sind in der Automatiksteuerung an den Parametern ErrorID und ErrorInfo der Bewegungssteuerungsanweisungen verfügbar. Im Handbetrieb zeigt das Feld "Letzter Fehler" im Steuerpanel ausführliche Informationen zur Fehlerursache an. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Error)
Steuerpanel aktiv	Der Modus "Manuelle Steuerung" wurde im Steuerpanel aktiviert. Das Steuerpanel hat die Steuerpriorität über das Technologieobjekt "Achse". Die Achse kann nicht über das Anwenderprogramm gesteuert werden. (Variable des Technologieobjekts: <Achsenname>.StatusBits.ControlPanelActive)

Tabelle 10- 41 Antriebsstatus

Status	Beschreibung
Antrieb bereit	Der Antrieb ist betriebsbereit. (Variable des Technologieobjekts: <Achsenname>.StatusBits.DriveReady)
Fehler	Der Antrieb hat nach Ausfall des Signals "Bereit" einen Fehler gemeldet. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.DriveFault)

Tabelle 10- 42 Status der Achsbewegung

Status	Beschreibung
Stillstand	Die Achse ist im Stillstand. (Variable des Technologieobjekts: <Achsenname>.StatusBits.StandStill)
Beschleunigung	Die Achse beschleunigt. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Acceleration)
Konstante Geschwindigkeit	Die Achse fährt mit konstanter Geschwindigkeit. (Variable des Technologieobjekts: <Achsenname>.StatusBits.ConstantVelocity)
Verzögerung	Die Achse verzögert (verlangsamt die Fahrt). (Variable des Technologieobjekts: <Achsenname>.StatusBits.Deceleration)

Tabelle 10- 43 Status des Bewegungsmodus

Status	Beschreibung
Positionierung	Die Achse führt eine Positionierungsaufgabe der Bewegungssteuerungsanweisung "MC_MoveAbsolute" oder "MC_MoveRelative" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.PositioningCommand)
Geschwindigkeitsbefehl	Die Achse führt eine Aufgabe mit festgelegter Geschwindigkeit der Bewegungssteuerungsanweisung "MC_MoveVelocity" oder "MC_MoveJog" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.SpeedCommand)
Referenzpunktfahrt	Die Achse führt eine Referenzpunktfahrt Aufgabe der Bewegungssteuerungsanweisung "MC_Home" oder des Steuerpanels aus. (Variable des Technologieobjekts: <Achsenname>.StatusBits.Homing)

Tabelle 10- 44 Fehlerbits

Fehler	Beschreibung
Min. Softwaregrenze erreicht	Der untere Software-Endschalter wurde erreicht. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMinReached)
Min. Softwaregrenze überschritten	Der untere Software-Endschalter wurde überschritten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMinExceeded)
Max. Softwaregrenze erreicht	Der obere Software-Endschalter wurde erreicht. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMaxReached)
Max. Softwaregrenze überschritten	Der obere Software-Endschalter wurde überschritten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SwLimitMaxExceeded)
Negative Hardwaregrenze	Der untere Hardware-Endschalter wurde angefahren. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwLimitMin)
Positive Hardwaregrenze	Der obere Hardware-Endschalter wurde angefahren. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwLimitMax)
PTO wird bereits verwendet	Eine zweite Achse verwendet die gleiche PTO und ist mit "MC_Power" freigegeben. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.HwUsed)
Konfigurationsfehler	Das Technologieobjekt "Achse" wurde falsch konfiguriert oder änderbare Konfigurationsdaten wurden während der Laufzeit des Anwenderprogramms fehlerhaft geändert. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.ConfigFault)
Allgemeiner Fehler	Ein interner Fehler ist aufgetreten. (Variable des Technologieobjekts: <Achsenname>.ErrorBits.SystemFault)

Diagnosefunktion "Bewegungsstatus"

Mit der Diagnosefunktion "Bewegungsstatus" überwachen Sie den Bewegungsstatus der Achse. Die Diagnosefunktionsanzeige ist im Online-Modus bei aktiver Achse in der Betriebsart "Manuelle Steuerung" und "Automatiksteuerung" verfügbar.

Tabelle 10- 45 Bewegungsstatus

Status	Beschreibung
Zielposition	Das Feld "Zielposition" zeigt die aktuelle Zielposition einer aktiven Positionierungsaufgabe der Bewegungssteuerungsanweisung "MC_MoveAbsolute" oder "MC_MoveRelative" oder des Steuerpanels an. Der Wert der "Zielposition" ist nur während der Ausführung einer Positionierungsaufgabe gültig. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.TargetPosition)
Aktuelle Position	Das Feld "Aktuelle Position" zeigt die aktuelle Position der Achse an. Wenn sich die Achse nicht am Referenzpunkt befindet, zeigt der Wert den Positionswert relativ zur Freigabeposition der Achse an. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.Position)
Aktuelle Geschwindigkeit	Das Feld "Aktuelle Geschwindigkeit" zeigt die aktuelle Geschwindigkeit der Achse an. (Variable des Technologieobjekts: <Achsenname>.MotionStatus.Velocity)

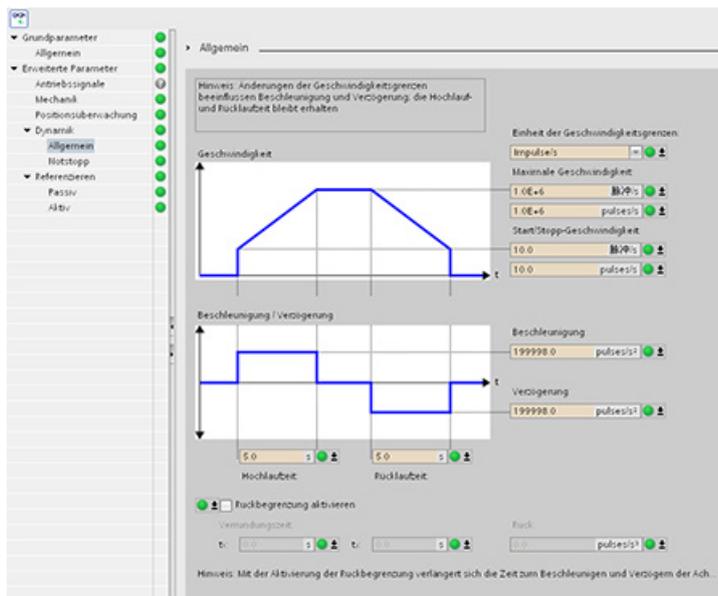
Tabelle 10- 46 Dynamikgrenzen

Dynamikgrenze	Beschreibung
Geschwindigkeit	Das Feld "Geschwindigkeit" zeigt die konfigurierte maximale Geschwindigkeit der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicLimits.MaxVelocity)
Beschleunigung	Das Feld "Beschleunigung" zeigt die aktuell konfigurierte Beschleunigung der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicDefaults.Acceleration)
Verzögerung	Das Feld "Verzögerung" zeigt die aktuell konfigurierte Verzögerung der Achse an. (Variable des Technologieobjekts: <Achsenname>.Config.DynamicDefaults.Deceleration)

Startwertsteuerung bei der Bewegungssteuerung

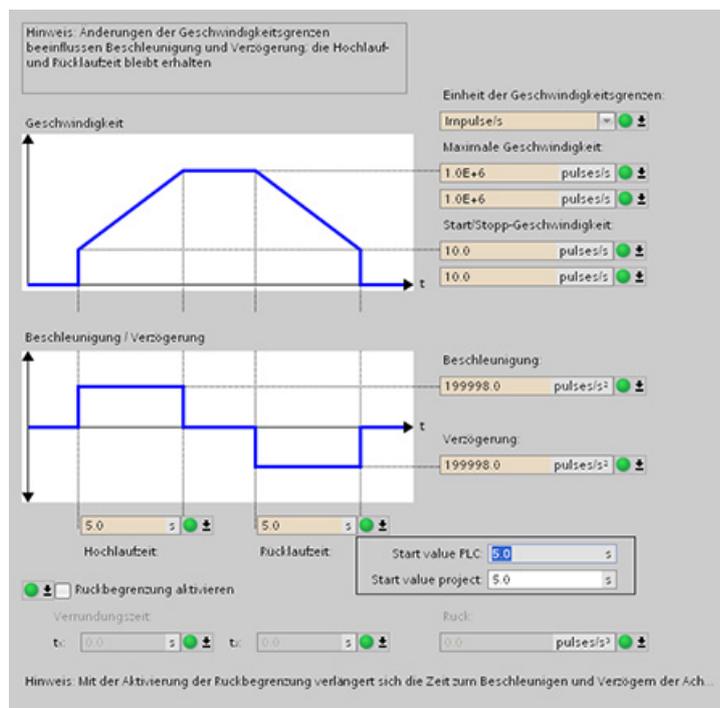
Sie können die Istwerte der Bewegungssteuerungs-Konfigurationsparameter so ändern, dass das Verhalten des Prozesses im Online-Modus optimiert werden kann.

Öffnen Sie die Technologieobjekte für die Bewegungssteuerung und deren Konfigurationsobjekt. Um die Startwertsteuerung aufzurufen, klicken Sie auf das Brillensymbol in der linken oberen Ecke des Dialogfelds:



Nun können Sie die Werte beliebiger Konfigurationsparameter Ihrer Bewegungssteuerung wie in nachstehender Abbildung gezeigt ändern.

Sie können für jeden Parameter den Istwert mit dem Startwert des Projekts (offline) und dem Startwert des PLCs (online) vergleichen. Dies ist erforderlich, um die Online-/Offline-Unterschiede des Technologieobjekt-Datenbausteins (TO-DB) zu vergleichen und die Werte zu kennen, die beim nächsten Wechsel von STOP nach START des PLCs als aktuelle Werte verwendet werden. Außerdem erhalten Sie durch ein Vergleichssymbol einen visuellen Hinweis, um Online-/Offline-Unterschiede schnell zu erkennen:

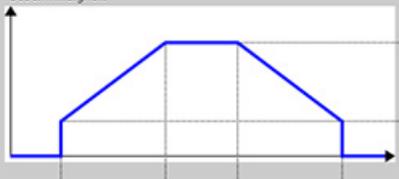


Die Abbildung oben zeigt den Bildschirm der Bewegungssteuerungsparameter mit Vergleichssymbolen. Die Symbole weisen darauf hin, welche Werte sich zwischen Online- und Offline-Projekt unterscheiden. Ein grünes Symbol kennzeichnet die Werte, die gleich sind. Ein blau-orangefarbenes Symbol zeigt an, dass die Werte voneinander abweichen.

Außerdem können Sie auf die Parameterschaltfläche mit dem Abwärtspfeil klicken, um ein kleines Fenster anzuzeigen, in dem für jeden Parameter der Startwert des Projekts (offline) und der Startwert des PLCs (online) angezeigt werden.

Hinweis: Änderungen der Geschwindigkeitsgrenzen beeinflussen Beschleunigung und Verzögerung, die Hochlauf- und Rücklaufzeit bleibt erhalten

Geschwindigkeit

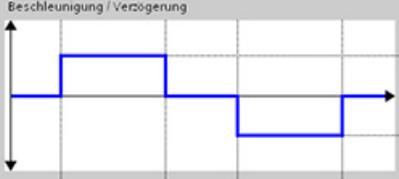


Einheit der Geschwindigkeitsgrenzen
Impulse/s

Maximale Geschwindigkeit
1.0E+6 impulse/s
1.0E+6 pulses/s

Start/Stop-Geschwindigkeit
10.0 impulse/s
10.0 pulses/s

Beschleunigung / Verzögerung



Beschleunigung
199998.0 pulses/s²

Verzögerung
199998.0 pulses/s²

Hochlaufzeit: 5.0 s
Rücklaufzeit: 5.0 s

Rückbegrenzung aktivieren

Verrundungszeit: t_r: 0.0 s
Ruck: 0.0 pulses/s³

Start value FLC: 5.0 s
Start value project: 5.1 s

Hinweis: Mit der Aktivierung der Rückbegrenzung verlängert sich die Zeit zum Beschleunigen und Verzögern der Ach...

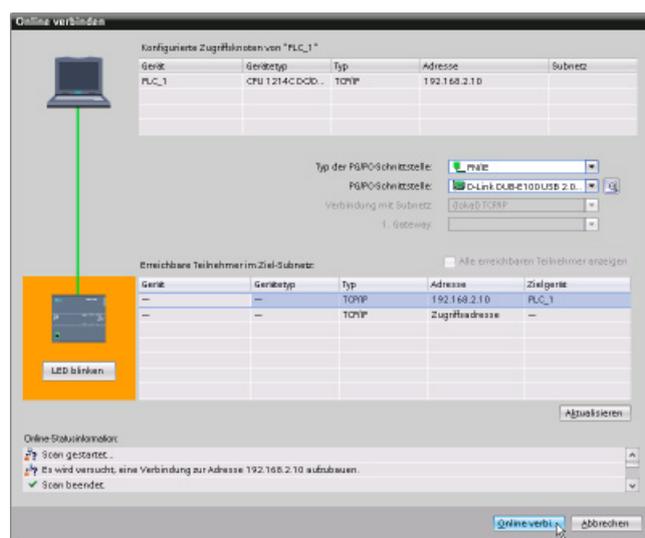
Einfaches Arbeiten mit den Online-Tools

11.1 Online-Verbindung mit einer CPU herstellen

Eine Online-Verbindung zwischen dem Programmiergerät und der CPU ist zum Laden von Programmen und Projektdaten sowie für die folgenden Tätigkeiten erforderlich:

- Testen von Anwenderprogrammen
- Anzeigen und Ändern des Betriebszustands der CPU (Seite 286)
- Anzeigen und Einstellen von Datum und Uhrzeit der CPU (Seite 296)
- Anzeigen der Modulinformationen
- Vergleichen und Synchronisieren (Seite 295) von Offline- und Online-Programmbausteinen
- Laden von Programmbausteinen in die und aus der CPU
- Anzeigen von Diagnose und Diagnosepuffer (Seite 296)
- Mit einer Beobachtungstabelle (Seite 288) das Anwenderprogramm durch Beobachten und Steuern von Werten testen
- Mit einer Forcetable Werte in der CPU forcen (Seite 290)

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 auf dieser Schnittstelle herstellen.

Ihr Programmiergerät ist nun mit der CPU verbunden. Orangefarbene Rahmen weisen auf eine Online-Verbindung hin. Nun können Sie die Tools unter "Online & Diagnose" in der Projektnavigation und die Taskcard "Online-Tools" verwenden.

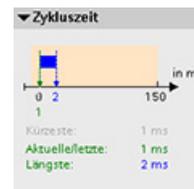
11.2 Interaktion mit der Online-CPU

Das in der Projektansicht in der Taskcard "Online-Tools" enthaltene Bedienpanel zeigt den Betriebszustand der Online-CPU an. Über das Bedienpanel können Sie auch den Betriebszustand der Online-CPU ändern. Mit der Schaltfläche auf dem Bedienpanel ändern Sie den Betriebszustand (STOP bzw. RUN). Außerdem enthält das Bedienpanel eine Schaltfläche MRES zum Umräumen des Speichers.

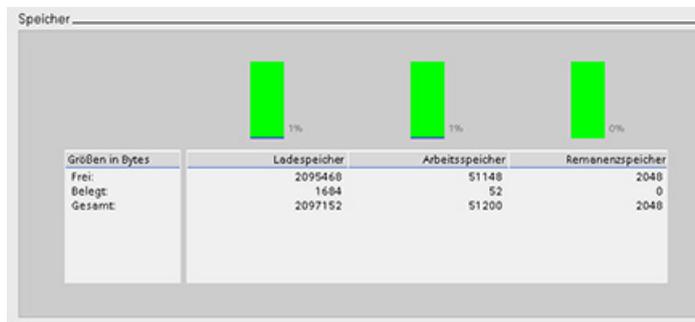


Der aktuelle Betriebszustand der CPU wird durch die Farbe der RUN/STOP-Anzeige angegeben: Gelb steht für den Betriebszustand STOP, Grün für RUN.

Um das Bedienpanel verwenden zu können, müssen Sie eine Online-Verbindung zwischen STEP 7 und der CPU herstellen. Nachdem Sie die CPU in der Gerätekonfiguration selektiert oder einen Codebaustein in der Online-CPU angezeigt haben, können Sie das Bedienpanel über die Taskcard "Online-Tools" aufrufen.



Sie können die Zykluszeit einer Online-CPU überwachen.



Außerdem können Sie die Speicherauslastung der CPU anzeigen.

11.3 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.

Variablen- und 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.

11.4 Einfaches Anzeigen des Status des Anwenderprogramms

Sie können den Zustand der Variablen auch 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.

11.5 Beobachtungstabelle zur Überwachung der CPU verwenden

Mit Hilfe einer Beobachtungstabelle können Sie Datenpunkte beobachten und ändern, während die CPU Ihr Anwenderprogramm ausführt. Bei den Datenpunkten kann es sich um Eingänge (E), Ausgänge (A), Merker (M) oder Peripherieeingänge (wie "On:P" oder "E 3.4:P" handeln. Sie können die physischen Ausgänge (wie A0.0: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 nicht den Programmablauf. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU. Sie können auch mit der Funktion "Wert ändern" die Ausführung Ihres Anwenderprogramms testen.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Beobachten mit Trigger	Steuern mit Trigger	Steuerwert
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"/>

Hinweis

Die vom schnellen Zähler (HSC), von der Impulsdauermodulation (PWM) und von der Impulsfolge (PTO) verwendeten E/A werden während der Konfiguration zugewiesen. Wenn diesen Geräten digitale E/A zugewiesen wurden, können die Adresswerte der zugewiesenen E/A nicht durch die Funktion "Forcen" der Beobachtungstabelle geändert werden.

Mit einer Beobachtungstabelle können Sie die Werte der einzelnen Variablen beobachten und ändern. Dabei gibt es folgende Möglichkeiten:

- Am Anfang oder Ende des Zyklus
- Wenn CPU in den Betriebszustand STOP geht
- "Dauerhaft" (dabei wird der Wert nach einem Wechsel von STOP in RUN nicht zurückgesetzt)



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.

Diese Beobachtung der Variablen setzt jedoch eine Online-Verbindung zur CPU voraus. Zum Ändern der Variablen stehen die folgenden Möglichkeiten zur Verfügung:

- "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.
- Mit "PA freischalten" können Sie die Peripherieausgänge aktivieren, wenn sich die CPU im Betriebszustand STOP befindet. Diese Funktion ist nützlich, um die Verschaltung der Ausgangsmodule zu prüfen.

Die verschiedenen Funktionen können über die Schaltflächen am oberen Rand einer Beobachtungstabelle ausgewählt werden. Geben Sie den Namen der zu beobachtenden Variablen ein und wählen Sie in der Klappliste ein Anzeigeformat. Besteht eine Online-Verbindung zu der CPU, wird durch Anklicken der Schaltfläche "Beobachten" der Istwert der Datenpunkt im Feld "Beobachtungswert" angezeigt.

11.6 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 Peripherieeingang 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 Ausgangs 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 Ausgangs 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.

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.

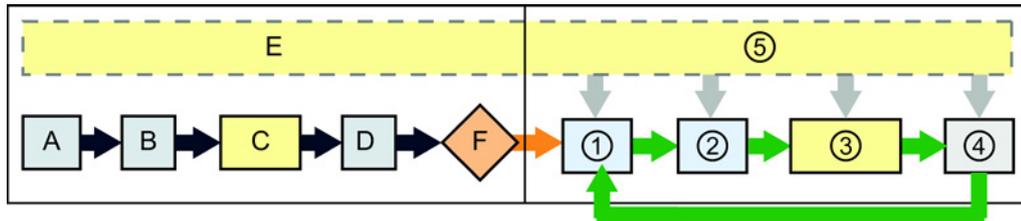
Wenn ein Eingang oder Ausgang in der Forcetabelle geforct wird, werden die Forceaktionen Teil des Anwenderprogramms. Auch wenn Sie die Programmiersoftware schließen, bleiben die geforcten Werte im ausgeführten CPU-Programm solange aktiv, bis Sie wieder in der Programmiersoftware online gehen und die 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 Forcelfunktion nicht beeinflusst.
- B Die Initialisierung der Ausgangswerte wird von der Forcelfunktion nicht beeinflusst.
- C Während der Ausführung der Anlauf-OBs schaltet die CPU den Forcwert 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 Forcwert bei der Aktualisierung der Ausgänge auf.
- ② Beim Lesen der physischen Eingänge wendet die CPU die Forcwerte an, kurz bevor die Eingänge in den Speicherbereich E kopiert werden.
- ③ Während der Ausführung der Anwenderprogramms (Programmzyklus-OBs) schaltet die CPU den Forcwert 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 Forcelfunktion nicht beeinflusst.
- ⑤ Die Verarbeitung von Alarmen während eines beliebigen Teils des Zyklus wird nicht beeinflusst.

11.7 Online-Werte eines DBs erfassen, um die Startwerte zurückzusetzen

Sie können die aktuellen Werte, die in einer Online-CPU beobachtet werden, erfassen und daraus Startwerte für einen globalen DB machen.

- Sie benötigen eine Online-Verbindung zur CPU.
- Die CPU muss sich im Betriebszustand RUN befinden.
- Sie müssen den DB in STEP 7 geöffnet haben.



Mit der Schaltfläche "Momentabbild der beobachteten Werte anzeigen" erfassen Sie die aktuellen Werte der ausgewählten Variablen im DB. Sie können diese Werte dann in die Spalte "Startwert" des DBs kopieren.

1. Klicken Sie im DB-Editor auf die Schaltfläche "Alle Variablen beobachten". Die Spalte "Beobachtungswert" zeigt die aktuellen Datenwerte an.
2. Klicken Sie auf die Schaltfläche "Momentabbild der beobachteten Werte anzeigen", um die aktuellen Werte in der Spalte "Momentabbild" anzuzeigen.
3. Klicken Sie auf die Schaltfläche "Alle beobachten", um die Beobachtung der Daten in der CPU zu beenden.
4. Kopieren Sie einen Wert aus der Spalte "Momentabbild" für eine Variable.
 - Wählen Sie einen zu kopierenden Wert aus.
 - Klicken Sie mit der rechten Maustaste auf den ausgewählten Wert, um das Kontextmenü aufzurufen.
 - Wählen Sie den Befehl "Kopieren".
5. Fügen Sie den kopierten Wert in die entsprechende Spalte "Startwert" der Variablen ein. (Klicken Sie mit der rechten Maustaste auf die Zelle und wählen Sie im Kontextmenü den Befehl "Einfügen".)
6. Speichern Sie das Projekt, um die kopierten Werte als neue Startwerte für den DB zu speichern.
7. Übersetzen Sie den DB und laden Sie ihn in die CPU. Der DB verwendet die neuen Startwerte, nachdem die CPU in den Betriebszustand RUN geht.

Hinweis

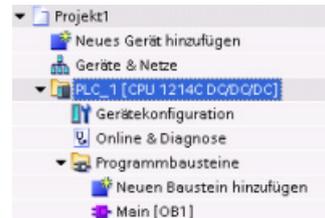
Die Werte, die in der Spalte "Beobachtungswert" angezeigt werden, sind immer aus der CPU kopiert. STEP 7 prüft nicht, ob alle Werte aus demselben Zyklus der CPU stammen.

11.8 Elemente des Projekts kopieren

Sie können die Programmbausteine einer Online-CPU oder einer an Ihr Programmiergerät angeschlossenen Memory Card auch kopieren.

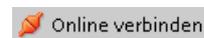
Bereiten Sie das Offline-Projekt für die kopierten Programmbausteine vor:

1. Fügen Sie eine CPU hinzu, die der Online-CPU entspricht.
2. Erweitern Sie den CPU-Knoten, so dass der Ordner "Programmbausteine" angezeigt wird.



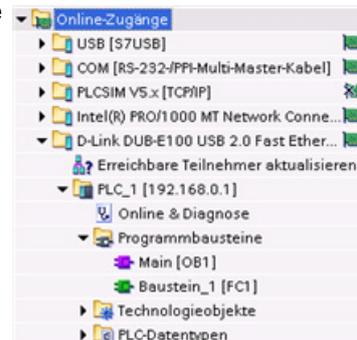
Um die Programmbausteine aus der Online-CPU in das Offline-Projekt zu laden, gehen Sie folgendermaßen vor:

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.



Als Alternative zur vorherigen Vorgehensweise gehen Sie folgendermaßen vor:

1. Erweitern Sie in der Projektnavigation den Knoten "Online-Zugänge", um die Programmbausteine in der Online-CPU auszuwählen.
2. Erweitern Sie den Knoten für das Netzwerk und doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren".
3. Erweitern Sie den Knoten für die CPU.
4. Ziehen Sie den Ordner "Programmbausteine" mit der Maus von der Online-CPU in den Ordner "Programmbausteine" Ihres Offline-Projekts.
5. Aktivieren Sie im Dialog "Vorschau für das Laden von Gerät" das Kontrollkästchen "Fortfahren", und klicken Sie dann auf "Laden von Gerät".



Nach dem Ladevorgang werden alle Programmbausteine, Technologiebausteine und Variablen im Offline-Bereich angezeigt.



Hinweis

Sie können die Programmbausteine aus der Online-CPU in ein vorhandenes Programm kopieren. Der Ordner "Programmbausteine" des Offline-Projekts muss nicht leer sein. Das vorhandene Programm wird jedoch gelöscht und durch das Anwenderprogramm aus der Online-CPU ersetzt.

11.9 Vergleichen von Offline- und Online-CPU's

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.



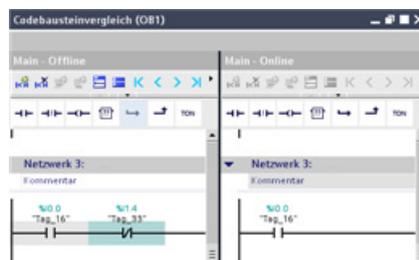
Wählen Sie die CPU in Ihrem Projekt aus.

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.



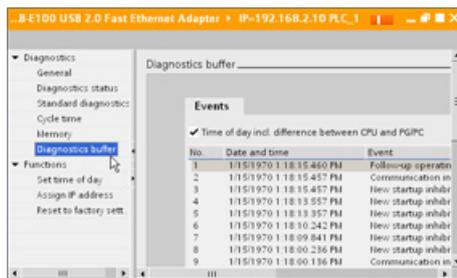
Klicken Sie in der Spalte "Vergleichen mit" mit der rechten Maustaste auf ein Objekt und wählen Sie "Detaillierten Vergleich starten", um die Codebausteine nebeneinander anzuzeigen.

Bei diesem Detailvergleich werden die Unterschiede zwischen den Codebausteinen der Online-CPU und den Codebausteinen der CPU in Ihrem Projekt hervorgehoben.

11.10 Diagnoseereignisse anzeigen

Die CPU bietet einen Diagnosepuffer, der für jedes Diagnoseereignis einen Eintrag enthält, z. B. für den Wechsel des CPU-Betriebszustands oder für Fehler, die von der CPU oder den Modulen festgestellt wurden. Für den Zugriff auf den Diagnosepuffer müssen Sie online sein.

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.

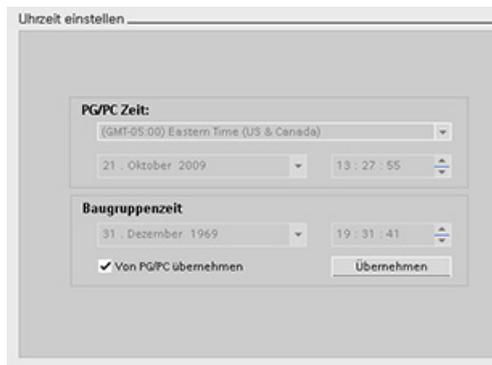


In diesem Puffer werden bis zu 50 aktuelle Ereignisse gespeichert, solange die CPU eingeschaltet ist. Ist der Puffer voll, wird immer das älteste Ereignis durch ein neues Ereignis überschrieben.

Bei einer Unterbrechung der Stromversorgung werden die letzten zehn Ereignisse gespeichert.

11.11 Einstellen der IP-Adresse und der Uhrzeit

Sie können die IP-Adresse 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.

11.12 Auf Werkseinstellungen zurücksetzen

Unter den folgenden Bedingungen können Sie eine S7-1200 auf die ursprünglichen Werkseinstellungen zurücksetzen:

- Es ist keine Memory Card in der CPU gesteckt.
- 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 Sicht "Online und Diagnose" der CPU.
2. Wählen Sie im Ordner "Funktionen" den Eintrag "Auf Werkseinstellungen zurücksetzen".
3. Aktivieren Sie das Kontrollkästchen "IP-Adresse beibehalten", wenn Sie die IP-Adresse beibehalten möchten, oder das Kontrollkästchen "IP-Adresse zurücksetzen", wenn Sie die IP-Adresse löschen möchten.
4. Klicken Sie auf die Schaltfläche "Zurücksetzen".
5. Bestätigen Sie die Eingabeaufforderung mit "OK".

Ergebnis

Das Modul wird ggf. in den Betriebszustand STOP versetzt und auf die Werkseinstellungen zurückgesetzt:

- Der Arbeitsspeicher und interne Ladespeicher sowie alle Operandenbereiche werden gelöscht.
- Alle Parameter werden auf ihre Standardwerte zurückgesetzt.
- Der Diagnosepuffer wird gelöscht.
- Die Uhrzeit wird zurückgesetzt.
- Die IP-Adresse wird je nach vorgenommener Einstellung beibehalten oder gelöscht. (Die MAC-Adresse ist fest zugewiesen und wird niemals geändert.)

11.13 Firmware aktualisieren

Sie können die Firmware der angeschlossenen CPU mit den Online- und Diagnosetools von STEP 7 aktualisieren.

Für ein Firmware-Update gehen Sie folgendermaßen vor:

1. Öffnen Sie die Online- und Diagnoseansicht der angeschlossenen CPU.
2. Wählen Sie im Ordner "Funktionen" die Option "Firmware-Update".
3. Klicken Sie auf die Such-Schaltfläche und navigieren Sie zum Speicherort der Datei mit dem Firmware-Update. Dies kann ein Speicherort auf Ihrer Festplatte sein, in den Sie eine Firmware-Update-Datei für S7-1200 (<http://support.automation.siemens.com/WW/view/de/34612486/133100>) von der Service- und Support-Webseite (<http://www.siemens.com/automation/>) heruntergeladen haben.
4. Wählen Sie eine Datei, die für Ihr Modul geeignet ist. In der Tabelle werden die kompatiblen Module für die ausgewählte Datei angezeigt.
5. 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

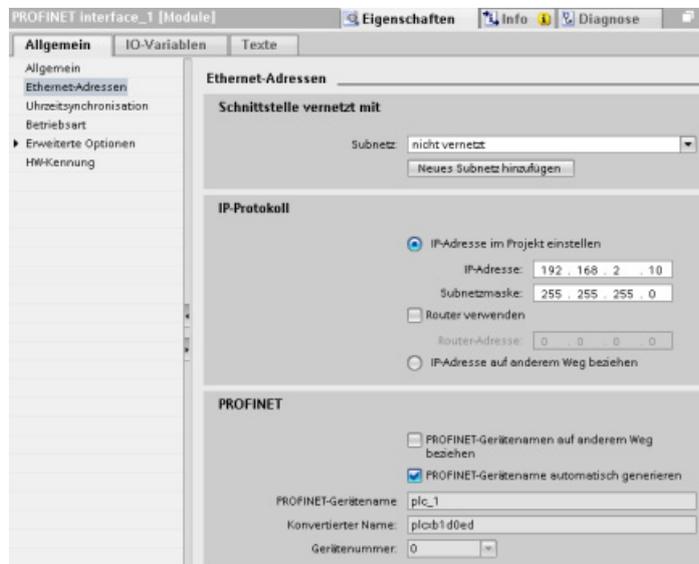
Wollen Sie das Modul mit der neuen Firmware nicht starten, bleibt die alte Firmware aktiv, bis das Modul, zum Beispiel durch Aus- und Einschalten, zurückgesetzt wird. Die neue Firmware wird erst nach dem Zurücksetzen des Moduls aktiv.

Sie können ein Firmware-Update auch mit einer der folgenden Methoden ausführen:

- Mit einer Memory Card
- Über die Standard-Webseite "Modulinformationen" des Webservers (Seite 210)

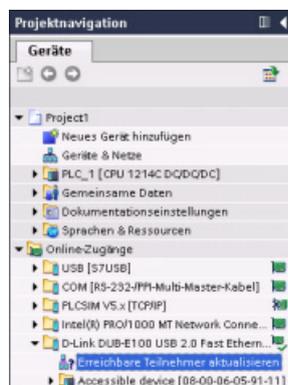
11.14 Laden einer permanenten IP-Adresse in eine Online-CPU

Um eine IP-Adresse zuzuweisen, gehen Sie folgendermaßen vor:



- Richten Sie die IP-Adresse für die CPU (Seite 89) ein.
- Die Konfiguration speichern und in die CPU laden

Die IP-Adresse und Subnetzmaske der CPU muss mit der IP-Adresse und Subnetzmaske des Programmiergeräts kompatibel sein. Die IP-Adresse und Subnetzmaske für Ihre CPU erfahren Sie von Ihrem Netzwerkspezialisten.



Wenn die CPU bisher noch nicht konfiguriert wurde, können Sie die IP-Adresse auch über die "Online-Zugänge" festlegen.

Eine IP-Adresse, die Sie zusammen mit der Gerätekonfiguration geladen haben, geht beim Ausschalten des Zielsystems nicht verloren.

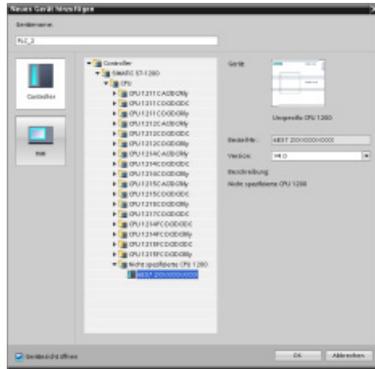
Nachdem Sie die Gerätekonfiguration einschließlich der IP-Adresse geladen haben, erscheint die IP-Adresse unter dem Ordner "Online-Zugänge".

11.15 Verwendung der "nicht spezifizierten CPU" zum Laden der Hardwarekonfiguration aus dem Gerät

Wenn Sie über eine physische CPU verfügen, die Sie an das Programmiergerät anschließen können, lässt sich die Konfiguration der Hardware problemlos aus dem Gerät laden.

11.15 Verwendung der "nicht spezifizierten CPU" zum Laden der Hardwarekonfiguration aus dem Gerät

Zunächst müssen Sie die CPU mit dem Programmiergerät verbinden und ein neues Projekt anlegen.

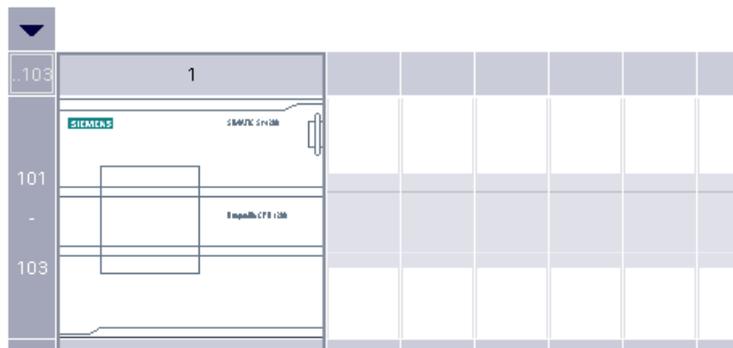


Fügen Sie in der Gerätekonfiguration (Projektansicht oder Portalansicht) ein neues Gerät hinzu, wählen Sie jedoch anstelle einer bestimmten CPU die "nicht spezifizierte CPU". STEP 7 legt eine nicht spezifizierte CPU an.



Nach dem Anlegen der nicht spezifizierten CPU können Sie die Hardwarekonfiguration von der Online-CPU laden.

- 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 haben, lädt STEP 7 die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Die IP-Adresse wird **nicht** geladen. Sie müssen die "Gerätekonfiguration" aufrufen, um die IP-Adresse manuell zu konfigurieren.

11.16 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 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

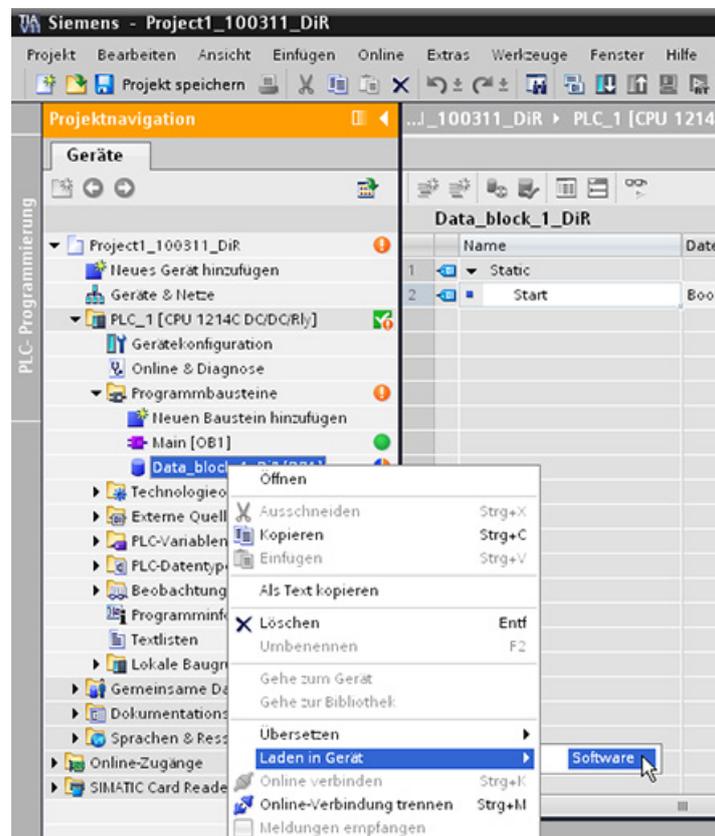
Befindet sich die CPU im Betriebszustand RUN und wurden Programmänderungen vorgenommen, versucht STEP 7 zunächst stets, die Änderungen im Betriebszustand RUN zu laden. Möchten Sie dies verhindern, müssen Sie die CPU in STOP versetzen.

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.

11.16.1 Ändern des Programms im Betriebszustand RUN

Um Programmänderungen in RUN vorzunehmen, vergewissern Sie sich zuerst, dass die CPU und das Programm die Voraussetzungen erfüllen, und gehen Sie dann wie folgt vor:

- 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".



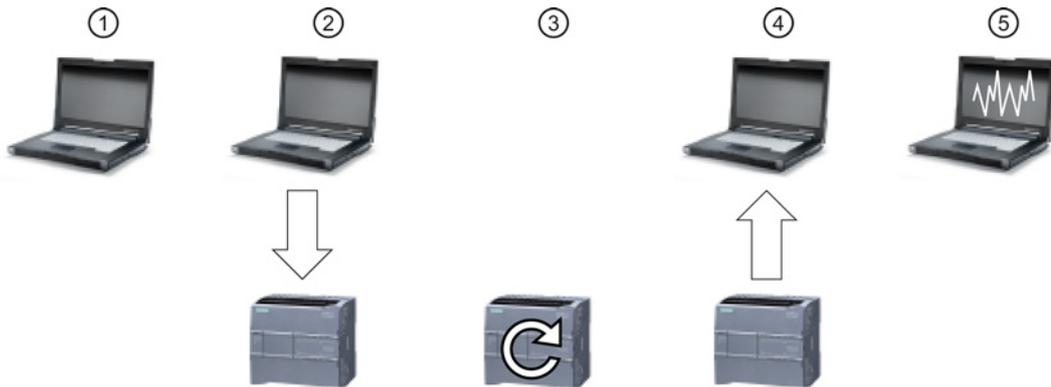
Wenn das Programm erfolgreich übersetzt wird, lädt STEP 7 das Programm in die CPU.

- Wenn STEP 7 Sie auffordert, das Programm zu laden oder den Vorgang abubrechen, klicken Sie auf "Laden", um das Programm in die CPU zu laden.

11.17 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 Trace-Daten 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.

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 aufzuzeichnenden Datenwerte, die Aufzeichnungsdauer und -häufigkeit sowie die Auslösebedingung festlegen.
- 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 Mit den Tools in STEP 7 können Sie die Daten analysieren, grafisch darstellen und speichern.

Beachten Sie, dass die Trace-Funktion erst ab Version V4.0 der S7-1200 CPU verfügbar ist.

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.

IO-Link – ganz einfach

12.1 Überblick über die IO-Link-Technologie

IO-Link ist eine von der PROFIBUS-Nutzerorganisation (PNO) definierte innovative Kommunikationstechnologie für Sensoren und Aktoren. IO-Link ist ein internationaler Standard nach IEC 61131-9. Er basiert auf einer Punkt-zu-Punkt-Verbindung zwischen den Sensoren und Aktoren (Slaves) und der Steuerung (Master). Es handelt sich also nicht um ein Bussystem, sondern um die Weiterentwicklung einer herkömmlichen Punkt-zu-Punkt-Verbindung.

Von den angeschlossenen Sensoren/Aktoren werden neben zyklischen Betriebsdaten auch umfangreiche Parameter- und Diagnosedaten übertragen. Für die Datenübertragung wird das gleiche 3-Leiter-Verbindungskabel verwendet wie für die Standardsensorik üblich.

12.2 Komponenten eines IO-Link-Systems

Ein IO-Link-System besteht aus IO-Link-Geräten (üblicherweise Sensoren, Aktoren oder einer Kombination aus beiden), einem 3-Leiter-Standardkabel für Sensoren/Aktoren und einem IO-Link-Master. Bei dem Master kann es sich um ein Gerät mit beliebiger Konstruktion und Schutzart handeln.

Ein IO-Link-Master kann einen oder mehrere Ports umfassen. Der SM 1278 4xIO-Link Master verfügt über vier Ports. An jeden Port kann ein IO-Link-Gerät oder ein Standardsensor/-aktor angeschlossen sein. IO-Link ist ein Punkt-zu-Punkt-Kommunikationssystem.

12.3 Nach dem Anlauf

Beim Anlauf befindet sich das IO-Link-Gerät stets im SIO-Modus (Standard-I/O-Modus). Die Ports des Masters können verschiedene Konfigurationen aufweisen. Weitere Informationen finden Sie im Kapitel zu IO-Link im S7-1200-Systemhandbuch.

Befindet sich ein Port im SIO-Modus, verhält sich der Master an diesem Port wie ein normaler Digitaleingang. Befindet sich der Port im IO-Link-Modus (Kommunikationsmodus), versucht der Master, das angeschlossene IO-Link-Gerät zu finden. Dieser Prozess wird "Wake up" genannt.

Beim Wake-up sendet der Master ein definiertes Signal und wartet auf die Antwort des Slaves. Anfänglich versucht der Master dies mit der höchstmöglichen Baudrate. Sind die Versuche nicht erfolgreich, fährt der Master mit niedrigerer Baudrate fort. Der Master spricht das Gerät mit jeder Baudrate drei Mal an. Das Gerät unterstützt jeweils nur eine definierte Baudrate. Empfängt der Master eine Antwort (d. h., das Gerät ist "aufgewacht"), beginnen beide die Kommunikation. Zunächst werden dabei die Kommunikationsparameter ausgetauscht, danach beginnt der zyklische Austausch von Prozessdaten.

Wird das Slave-Gerät im Betrieb entfernt, erkennt der Master den Kommunikationsabbruch, meldet diesen der Steuerung mit Feldbusgenauigkeit und versucht in regelmäßigen Abständen erneut, das Gerät aufzuwecken. Nach einem weiteren erfolgreichen Wake-up werden die Kommunikationsparameter erneut ausgelesen, gegebenenfalls validiert, und anschließend wird der zyklische Kommunikationskanal erneut gestartet.

12.4 IO-Link-Protokoll

Über das IO-Link-System können drei Arten von Daten ausgetauscht werden:

- Zyklische Prozessdaten (Eingänge, Ausgänge der Prozessdaten) → zyklische Daten
- Geräteparameter (Datenobjekte auf Anforderung) → azyklische Daten
- Ereignisse → azyklische Daten

Das IO-Link-Gerät sendet die Daten erst, nachdem es vom IO-Link-Master dazu aufgefordert wurde. Prozessdaten werden nach dem IDLE-Telegramm des Masters gesendet und der Master fordert Daten und Ereignisse der Geräteparameter explizit an.

12.5 Konfiguration des Feldbusses

Der IO-Link-Master erscheint auf dem Feldbus als normaler Feldbusknoten und wird über die entsprechenden Gerätebeschreibung im zugehörigen Netzwerkkonfigurator eingebunden. Diese Dateien beschreiben die Kommunikationseigenschaften und andere Merkmale des IO-Link-Masters wie z. B. die Anzahl der Ports. Sie enthalten keine Angabe über die angeschlossenen IO-Link-Geräte.

Anhand der IO-Link-Gerätebeschreibung (IO-Link Device Description, IODD) jedoch können Sie eine vollständige transparente Darstellung der Systemarchitektur bis zum IO-Link-Gerät erstellen. Sie können mithilfe der IODD und des IO-Link-Konfigurationswerkzeugs S7-PCT konfigurieren, welches IO-Link-Gerät an welchen Port Ihres IO-Link-Masters angeschlossen ist.

Ausführliche Hinweise zur Konfiguration finden Sie im S7-PCT-Hilfesystem sowie im S7-1200 System Manual.

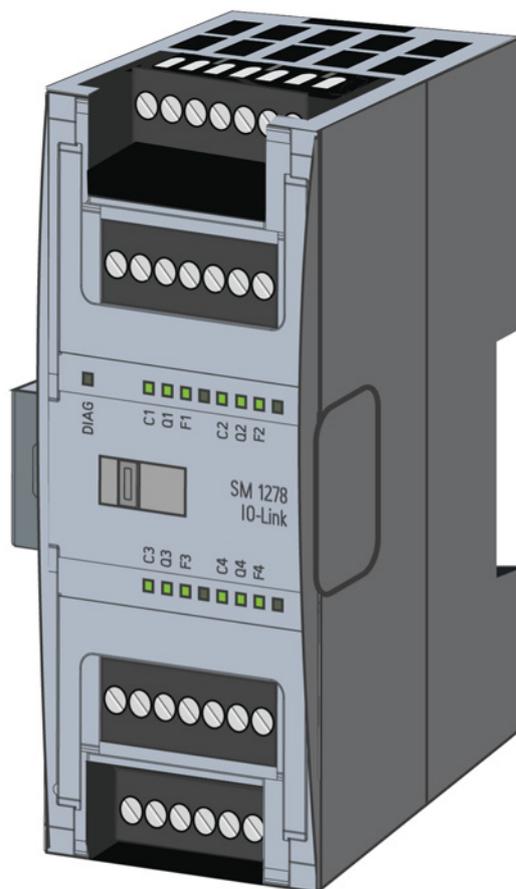
12.6 IO-Link und Ihr STEP 7-Programm

Der IO-Link-Master programmiert die azyklische Kommunikation mit dem IO-Link-Gerät über den Funktionsbaustein (FB) für den IOL_CALL-Aufruf in Ihrem S7-1200-Steuerungsprogramm in STEP 7. Der Funktionsbaustein IOL_CALL zeigt den von Ihrem Programm verwendeten IO-Link-Master sowie die vom Master für den Datenaustausch verwendeten Ports an.

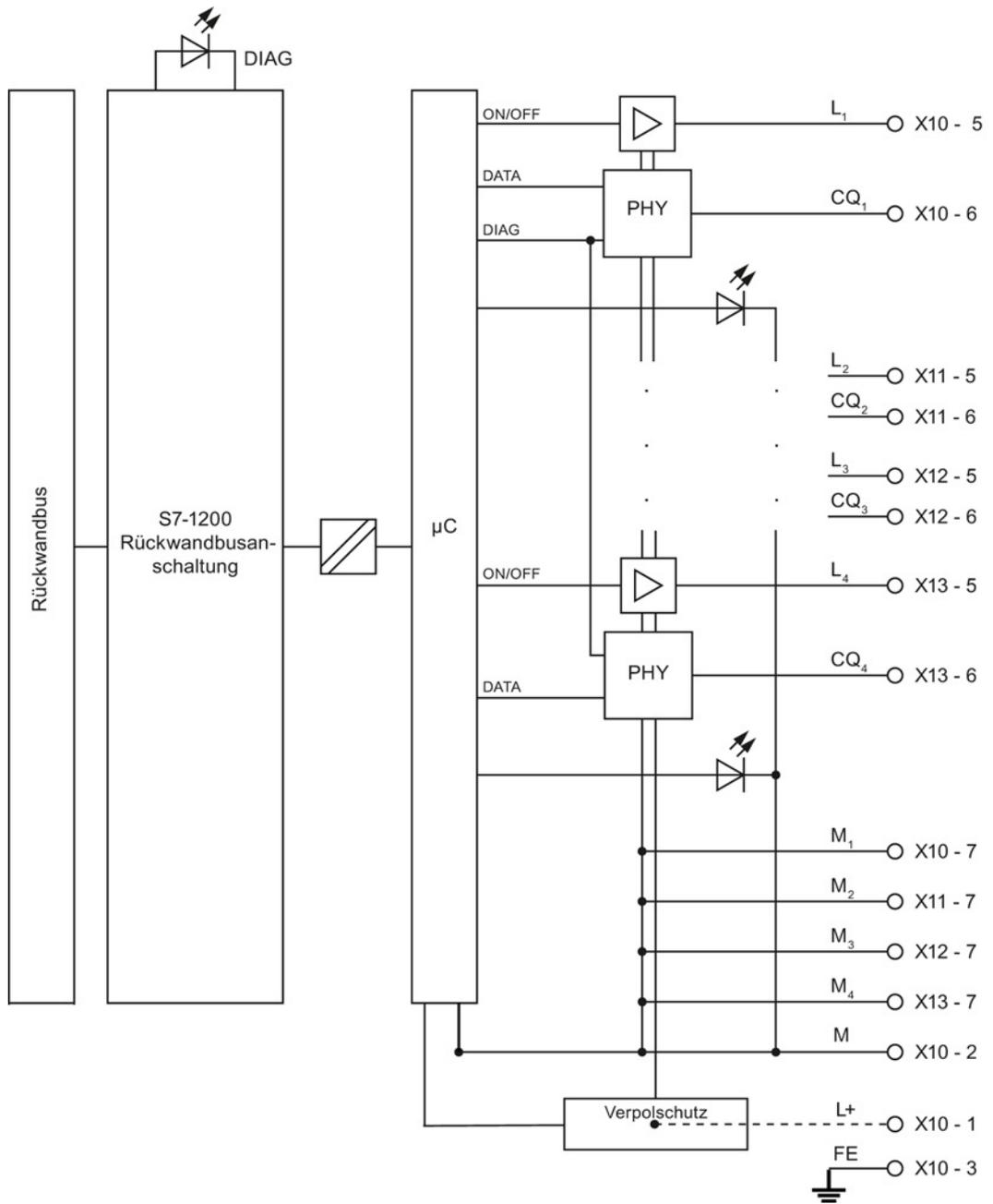
Auf der Siemens Industry-Website für Online-Support (<http://support.automation.siemens.com>) finden Sie ausführliche Informationen zum Arbeiten mit dem Funktionsbaustein IOL_CALL. Geben Sie im Suchfeld der Website "IO-Link" ein, um Informationen über IO-Link-Produkte und deren Einsatz aufzurufen.

12.7 Der SM 1278 4xIO-Link-Master

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. Sie können bis zu vier IO-Link-Geräte (3-Leiter-Anschluss) oder vier Standardaktoren oder Standarddrehgeber anschließen.

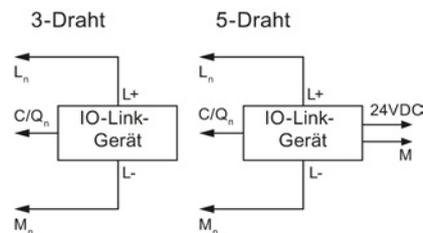


Schaltbild des SM 1278 4xIO-Link-Masters

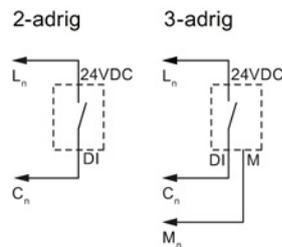


Beispiele für Anschlüsse

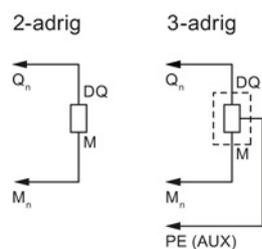
Die folgende Abbildung zeigt die Konfiguration des IO-Link-Betriebszustands (3-Leiter und 5-Leiter); dabei ist n = Portnummer:



Die folgende Abbildung zeigt die Konfiguration des DI-Betriebszustands (2-Leiter und 3-Leiter); dabei ist n = Portnummer:



Die folgende Abbildung zeigt die Konfiguration des DO-Betriebszustands (2-Leiter und 3-Leiter), dabei ist n = Portnummer:



Ausführliche Informationen zur Verwendung und Konfiguration des SM 1278 4xIO-Link-Masters

Ausführliche Informationen zum SM 1278 4xIO-Link Master einschließlich Abbildungen, Verbindungen, Parametrierung, Diagnosealarmen usw. finden Sie im S7-1200-Systemhandbuch.

Technische Daten

A.1 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 61131-2:2007 Speicherprogrammierbare Steuerungen - Betriebsmittelanforderungen und Prüfungen
- EU-Richtlinie 2004/108/EG (EMV-Richtlinie) "Elektromagnetische Verträglichkeit"
 - Störaussendung
EN 61000-6-4:2007: Industriebereich
 - Funkentstörung
EN 61000-6-2:2005: Industriebereich
- EG-Richtlinie 94/9/EG (ATEX) "Geräte und Schutzsysteme zur bestimmungsgemäßen Verwendung in explosionsgefährdeten Bereichen"
 - EN 60079-15:2005: Schutzart 'n':

Die CE-Konformitätserklärung steht allen zuständigen Behörden zur Verfügung bei der:

Siemens AG
IA AS RD ST PLC Amberg
Werner-von-Siemens-Str. 50
D92224 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



Factory Mutual Research (FM)

Zertifizierungsnorm Klasse Nummer 3600 und 3611

Zugelassen für den Einsatz in:

Class I, Division 2, Gas Group A, B, C, D, Temperature Class T3C Ta = 60 °C

Class I, Zone 2, IIC, Temperature Class T3 Ta = 60 °C

Canadian Class I, Zone 2 Installation nach CEC 18-150

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.
--

ATEX-Zulassung



Die ATEX-Zulassung gilt nur für DC-Varianten. Die ATEX-Zulassung gilt nicht für AC- und Relaisvarianten.

EN 60079-0:2009: Explosionsfähige Atmosphäre - Allgemeine Anforderungen

EN 60079-15:2010: Elektrische Betriebsmittel für gasexplosionsgefährdete Bereiche; Schutzart 'nA'

II 3 G Ex nA IIC T4 oder T3 Gc

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.

Im Aufbau muss gewährleistet sein, dass kurzzeitige Störspannungen (Transienten) auf weniger als 119 V begrenzt sind. Siehe "Stoßwellenfestigkeit" in diesem Abschnitt.

WICHTIGE AUSNAHME: Die Anzahl der Eingänge und Ausgänge, die gleichzeitig eingeschaltet sein dürfen, entnehmen Sie den technischen Daten. Einige Modelle sind auf $T_a = 60\text{ °C}$ herabgesetzt.

C-Tick-Zulassung



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Normen nach AS/NZS 2064 (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 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:

- ABS (American Bureau of Shipping)
- BV (Bureau Veritas)
- DNV (Det Norske Veritas)
- GL (Germanischer Lloyd)
- LRS (Lloyds Register of Shipping)
- Class NK (Nippon Kaiji Kyokai)
- Korean Register of Shipping

Industrienumgebungen

Das Automatisierungssystem S7-1200 wurde für den Einsatz in Industrienumgebungen entwickelt.

Tabelle A- 1 Industrienumgebungen

Anwendungsgebiet	Anforderungen an die Störaussendung	Anforderungen an die Störfestigkeit
Industrie	EN 61000-6-4:2007	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 Elektrostatistische 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

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 EN 61000-4-5 enthalten, wobei die Betriebsgrenzen in EN 61000-6-2 festgelegt werden. S7-1200 DC-CPU's und Signalmodule erfordern externen Schutz, um den sicheren Betrieb aufrecht zu erhalten, wenn Stoßspannungen gemäß Definition in dieser Norm anliegen.

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:

- Versorgungsspannung, 24 V DC: DEHN, Inc., Typ BVT AD 24, Bestellnummer 918 402
- Digitaleingänge, 24 V DC: DEHN, Inc., Typ DCO RK E 24, Bestellnummer 919 988
- Digitalausgänge, 24 V DC: DEHN, Inc., Typ DCO RK E 24, Bestellnummer 919 988
- Analogeingänge: DEHN, Inc., Typ BXT ML4 BD12, Bestellnummer 920 342
- Analogausgänge: DEHN, Inc., Typ DCO RK E 12, Bestellnummer 919 987
- Ethernet: DEHN, Inc., Typ DPA M CAT6 RJ45S 48, Bestellnummer 929 100
- RS-232: DEHN, Inc., Typ DCO RK E 12, Bestellnummer 919 987
- RS-485: DEHN, Inc., Typ BXT ML2 BE HFS 5, Bestellnummer 920 270
- Relaisausgänge: Kein Schutz erforderlich

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 55011, 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
Abgestrahlte Störaussendungen EN 55011, 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

Umgebungsbedingungen

Tabelle A-4 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, 5 Zyklen
EN 60068-2-32 Freier Fall	0,3 m, 5 Mal, in Versandverpackung
Atmosphärischer Druck	1080 bis 660 hPa (entspricht einer Höhe von -1000 bis 3500 m)

Tabelle A-5 Betriebsbedingungen

Umgebungsbedingungen - Betrieb	
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	1080 bis 795 hPa (entspricht einer Höhe von -1000 bis 2000m)
Konzentration von Schmutzstoffen	SO ₂ : < 0,5 ppm; H ₂ S: < 0,1 ppm; rel. LF < 60% nicht kondensierend
EN 60068-2-14, Test Nb, Temperaturveränderung	5 °C bis 55 °C, 3 °C/Minute
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 9-150 Hz Schalttafeleinbau: 7,0 mm von 5 bis 9 Hz; 2 G von 9 bis 150 Hz 10 Ablenkungen je Achse, 1 Oktave/Minute

A.1 Allgemeine technische Daten

Tabelle A-6 Hochspannungsisolationsprüfung

Hochspannungsisolationsprüfung	
Stromkreis mit 24V-DC-/5VDC-Nennspannung	520 V DC (Typprüfung der optischen Potentialtrennungsgrenzen)
115-V-AC-/230VAC-Stromkreis an Erde	1500 V AC
115-V-AC-/230VAC-Stromkreis an 115-V-AC-/230-VAC-Stromkreis	1500 V AC
115-V-AC-/230VAC-Stromkreis an 24-V-DC-/5V-DC-Stromkreis	1500 V AC (3000 V AC / 4242 V DC Typprüfung)
Ethernet-Port an 24-V-DC-/5-V-DC-Stromkreise und Erde ¹	1500 V AC (nur Typprüfung)

¹ Die Potentialtrennung des Ethernet-Ports ist dafür ausgelegt, die Gefahr während kurzfristiger Netzausfälle von gefährlichen Spannungen zu begrenzen. Sie entspricht nicht den Sicherheitsanforderungen der routinemäßigen Potentialtrennung von AC-Netzspannungen.

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 bis 264 V AC, 47 bis 63 Hz

Hinweis

Wenn ein mechanischer Kontakt die Ausgangsspannung zur S7-1200 CPU oder einem digitalen Signalmodul 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 tödlichen oder schweren Verletzungen und/oder Sachschaden führen kann. Dies 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 Schaden oder unvorhersehbaren Betrieb verursachen, was zu Tod oder schwerer Körperverletzung und/oder Sachschaden 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 der Last angepasster externer Schutzkreis verlängert die Lebensdauer der Kontakte. Die typische Lebensdauer von Öffnerkontakten beträgt etwa ein Drittel der Lebensdauer eines Schließers bei induktiver und Lampenlast.

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	12,0 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			
	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	

A.2 CPU-Module

Eine vollständige Liste der Module für die S7-1200 finden Sie im S7-1200 Systemhandbuch oder auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

Tabelle A- 9 Allgemeine technische Daten

Allgemeine technische Daten		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Bestellnummer	• AC/DC/Relais	• 6ES7 211-1BE40-0XB0	• 6ES7 212-1BE40-0XB0	• 6ES7 214-1BG40-0XB0	• 6ES7 215-1BG40-0XB0	• -
	• DC/DC/Relais	• 6ES7 211-1HE40-0XB0	• 6ES7 212-1HE40-0XB0	• 6ES7 214-1HG40-0XB0	• 6ES7 215-1HG40-0XB0	• -
	• DC/DC/DC	• 6ES7 211-1AE40-0XB0	• 6ES7 212-1AE40-0XB0	• 6ES7 214-1AG40-0XB0	• 6ES7 215-1AG40-0XB0	• 6ES7 217-1AG40-0XB0
Abmessungen B x H x T (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75	130 x 100 x 75	150 x 100 x 75
Gewicht	• AC/DC/Relais	• 420 Gramm	• 425 Gramm	• 475 Gramm	• 585 Gramm	-
	• DC/DC/Relais	• 380 Gramm	• 385 Gramm	• 435 Gramm	• 550 Gramm	-
	• DC/DC/DC	• 370 Gramm	• 370 Gramm	• 415 Gramm	• 520 Gramm	530 Gramm
Leistungserlust	• AC/DC/Relais	• 10 W	• 11 W	• 14 W	• 14 W	-
	• DC/DC/Relais	• 8 W	• 9 W	• 12 W	• 12 W	-
	• DC/DC/DC	• 8 W	• 9 W	• 12 W	• 12 W	12 W
Verfügbare Strom (5 VDC) für SM- und CM-Bus		max. 750 mA	max. 1000 mA	max. 1600 mA	max. 1600 mA	max. 1.600 mA
Verfügbare Strom (24 V DC) Geberspannung		max. 300 mA	max. 300 mA	max. 400 mA	max. 400 mA	max. 400 mA
Stromaufnahme digitaler Eingang (24 V DC)		4 mA/Eingang	4 mA/Eingang	4 mA/Eingang	4 mA/Eingang	4 mA / Eingang

Tabelle A- 10 CPU-Merkmale

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Anwenderspeicher					
• Arbeitsspeicher	• 30 KB	• 50 KB	• 75 KB	• 100 KB	• 125 KB
• Ladespeicher	• 1 MB	• 1 MB	• 4 MB	• 4 MB	• 4 MB
• Remanenter Speicher	• 10 KB	• 10 KB	• 10 KB	• 10 KB	• 10 KB
Integrierte digitale E/A	6 Eingänge	8 Eingänge	14 Eingänge	14 Eingänge	14 Eingänge
Siehe technische Daten (Seite 331).	4 Ausgänge	6 Ausgänge	10 Ausgänge	10 Ausgänge	10 Ausgänge
Integrierte analoge E/A	2 Eingänge	2 Eingänge	2 Eingänge	2 Eingänge	2 Eingänge
Siehe technische Daten (Seite 341).				2 Ausgänge	2 Ausgänge

CPU-Merkmale		CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Größe des Prozessabbilds						
• Eingänge		• 1024 Byte	• 1024 Byte	• 1024 Byte	• 1024 Byte	• 1024 Byte
• Ausgänge		• 1024 Byte	• 1024 Byte	• 1024 Byte	• 1024 Byte	• 1024 Byte
Merker (M)		4096 Byte	4096 Byte	8192 Byte	8192 Byte	8192 Byte
Temporärer (lokaler) Speicher		<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich zugehöriger FBs und FCs). • 6 KB für jede andere Alarmprioritätsstufe (einschließlich FBs und FCs) 				
SM-Erweiterung		Keine	max. 2 SMs	max. 8 SMs	max. 8 SMs	max. 8 SMs
Zusätzliche SB, CB oder BB		max. 1	max. 1	max. 1	max. 1	max. 1
Zusätzliche CM		max. 3	max. 3	max. 3	max. 3	max. 3
Schnelle Zähler	Summe	Bis zu 6 konfiguriert für die Verwendung beliebiger integrierter oder SB-Eingänge				
	1 MHz	--	--	--	--	Eb.2 bis Eb.5 (Differenz)
	100/180 kHz	Ea.0 bis Ea.5	Ea.0 bis Ea.5	Ea.0 bis Ea.5	Ea.0 bis Ea.5	Ea.0 bis Ea.5
	30/120 kHz	--	Ea.6 bis Ea.7	Ea.6 bis Eb.5	Ea.6 bis Eb.5	Ea.6 bis Eb.1
Impulsausgänge ²	Summe	Bis zu 4 konfiguriert für die Verwendung beliebiger integrierter oder SB-Ausgänge				
	1 MHz	--	--	--	--	Aa.0 bis Aa.3 (Differenz)
	100 kHz	Aa.0 bis Aa.3	Aa.0 bis Aa.3	Aa.0 bis Aa.3	Aa.0 bis Aa.3	Aa.4 bis Ab.1
	30 kHz	--	Aa.4 bis Aa.5	Aa.4 bis Ab.1	Aa.4 bis Ab.1	--
Eingänge für Impulsabgriff	6	8	14	14	14	
Verzögerungsalarme	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	
Weckalarme	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	4 mit Auflösung von 1 ms	
Flankenalarme	6 steigend und 6 fallend	8 steigend und 8 fallend	12 steigend und 12 fallend	12 steigend und 12 fallend	12 steigend und 12 fallend	
Optional mit SB	10 steigend und 10 fallend	12 steigend und 12 fallend	16 steigend und 16 fallend	16 steigend und 16 fallend	16 steigend und 16 fallend	

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C	CPU 1215C	CPU 1217C
Echtzeituhr					
• Genauigkeit	• +/- 60 Sekunden/Monat				
• Pufferung (wartungsfreier Hochleistungskondensator)	• Typ. 20 Tage/min. 12 Tage bei 40 °C				
Ausführungsgeschwindigkeit					
• Boolescher Wert	• 0,08 µs/Anweisung				
• Wort übertragen	• 1,7 µs/Anweisung				
• Realzahlenarithmetik	• 2,3 µs/Anweisung				

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 Kommunikation

Technische Daten	CPU 1211C, CPU 1212C, CPU 1214C	CPU 1215C, CPU 1217C
Kommunikation	1 Ethernet-Port	2 Ethernet-Ports
• Datenraten	• 10/100 MBit/s	• 10/100 MBit/s
• Elektrische Trennung (externes Signal zu PLC-Logik)	• Wandler potenzialgetrennt, 1500 V DC • CAT5e geschirmt	• Wandler potenzialgetrennt, 1500 V DC • CAT5e geschirmt
• Kabelart		
Geräte	• 4 HMI • 1 PG	• 4 HMI • 1 PG
Ethernet-Verbindungen ¹	8 (aktiv oder passiv)	8 (aktiv oder passiv)
CPU-zu-CPU- Kommunikation über S7 (GET/PUT)	• 8 (Client) • 3 (Server)	• 8 (Client) • 3 (Server)

¹ Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV.

Tabelle A- 12 Schaltplan der CPU 1214C AC/DC/Relais

CPU 1214C AC/DC/Relais	
<p>The diagram shows the terminal block connections for the CPU 1214C AC/DC/Relais. On the left, there are AC terminals labeled L1, N, and L2, with a note '120-240VAC'. Next to them are DC terminals labeled L+, M, and N(-), with a note '24VDC'. A series of terminals for digital inputs are labeled '24VDC INPUTS' and divided into 'DI a' (terminals 0-5) and 'DI b' (terminals 0-5). To the right are 'ANALOG INPUTS' labeled 'AI' (terminals 0-1). Below these is a terminal block 'X12' for 'RELAY OUTPUTS', divided into 'DO a' (terminals 1L, 0, .1, .2, .3, .4) and 'DO b' (terminals 2L, .5, .6, .7, .0, .1). The terminal block is identified as '214-1BG40-0XB0' with a 'X12 3' label. At the bottom, there are terminals for L(+) and N(-) connected to a power source.</p>	<p>① 24-V-DC-Geberverspannung. Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.</p>
	<p>② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.</p>
	<p>Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C des S7-1200 Systemhandbuchs (Ersatzteile) finden Sie die Bestellnummer.</p>
<p>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 fungieren und muss nicht geerdet werden. Für die Klemmen L1 und N (L2) ist keine Polarisierung erforderlich.</p>	

Tabelle A- 13 Schaltplan der CPU 1214C DC/DC/DC

CPU 1214C DC/DC/DC	
	<p>① 24-V-DC-Geberspannung. Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.</p>
	<p>② Bei stromziehenden Eingängen "-" an "M" anschließen (abgebildet). Bei stromliefernden Eingängen "+" an "M" anschließen.</p>
	<p>Hinweis 1: X11-Steckverbinder müssen vergoldet sein. Im Anhang C des S7-1200 Systemhandbuchs (Ersatzteile) finden Sie die Bestellnummer.</p>

A.3 Digitale Erweiterungsmodule

Eine vollständige Liste der Module für die S7-1200 finden Sie im S7-1200 Systemhandbuch oder auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

A.3.1 SB 1221, SB 1222 und SB 1223 Digitaleingabe/-ausgabe (DI, DO und DI/DO)

Tabelle A- 14 SB 1221 Digitaleingabe (DI) und SB 1222 Digitalausgabe (DO)

Allgemeines	SB 1221 4 DI (200 kHz)	SB 1222 4 DO (200 kHz)						
Bestellnummer	<ul style="list-style-type: none"> 24 V DC: 6ES7 221-3BD30-0XB0 5 V DC: 6ES7 221-3AD30-0XB0 	<ul style="list-style-type: none"> 24 V DC: 6ES7 222-1BD30-0XB0 5 V DC: 6ES7 222-1AD30-0XB0 						
Abmessungen B x H x T (mm)	38 x 62 x 21	38 x 62 x 21						
Gewicht	35 Gramm	35 Gramm						
Leistungsverlust	<ul style="list-style-type: none"> 24 V DC: 1,5 W 5 V DC: 1,0 W 	0,5 W						
Stromaufnahme	<table border="0"> <tr> <td>SM-Bus</td> <td>40 mA</td> <td>35 mA</td> </tr> <tr> <td>24 V DC</td> <td> <ul style="list-style-type: none"> 24 V DC: 7 mA / Eingang + 20 mA 5 V DC: 15 mA / Eingang + 15 mA </td> <td>15 mA</td> </tr> </table>	SM-Bus	40 mA	35 mA	24 V DC	<ul style="list-style-type: none"> 24 V DC: 7 mA / Eingang + 20 mA 5 V DC: 15 mA / Eingang + 15 mA 	15 mA	
SM-Bus	40 mA	35 mA						
24 V DC	<ul style="list-style-type: none"> 24 V DC: 7 mA / Eingang + 20 mA 5 V DC: 15 mA / Eingang + 15 mA 	15 mA						
Eingänge/Ausgänge	4 Eingänge (stromliefernd)	4 Ausgänge (elektronisch, MOSFET)						

Tabelle A- 15 SB 1223 Kombinationsmodule mit digitalen Eingängen/Ausgängen

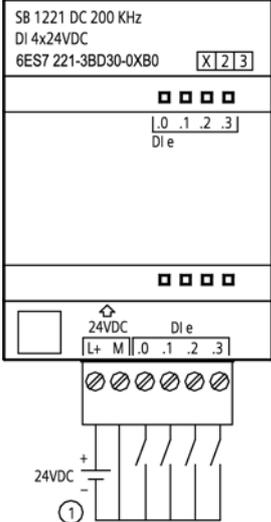
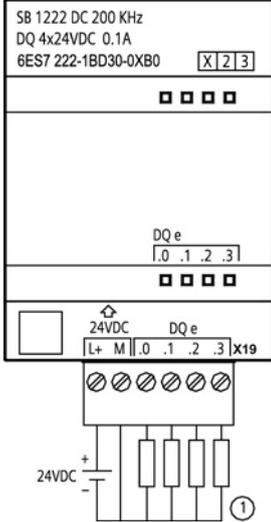
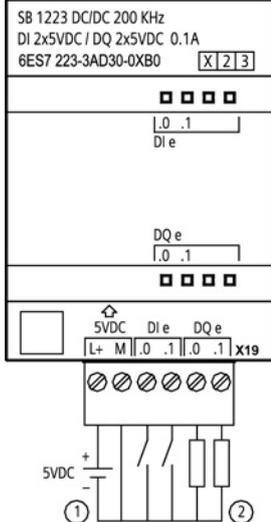
Allgemeines	SB 1223 DI / DO (200 kHz)	SB 1223 2 DI / 2 DO
Bestellnummer	<ul style="list-style-type: none"> 24 V DC: 6ES7 223-3BD30-0XB0 5 V DC: 6ES7 223-3AD30-0XB0 	24 V DC: 6ES7 223-0BD30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21	38 x 62 x 21
Gewicht	35 Gramm	40 Gramm
Leistungsverlust	<ul style="list-style-type: none"> 24 V DC: 1,0 W 5 V DC: 0,5 W 	24 V DC: 1,0 W
Stromaufnahme	SM-Bus	50 mA
	24 V DC	<ul style="list-style-type: none"> 24 V DC: 7 mA / Eingang + 20 mA 5 V DC: 15 mA / Eingang + 15 mA
Eingänge/Ausgänge	2 Eingänge (stromliefernd)	2 Eingänge (IEC Typ 1 stromziehend)
	2 Ausgänge (elektronisch, MOSFET)	2 Ausgänge (elektronisch, MOSFET)

Hinweis

Die schnellen SBs (200 kHz) nutzen "stromliefernde" Eingänge. Das Standard-SB (20 kHz) nutzt "stromziehende" Eingänge. Siehe technische Daten der digitalen Eingänge und Ausgänge (Seite 331).

Die schnellen (200 kHz) Ausgänge (SB 1222 und SB 1223) können stromliefernd oder stromziehend sein. Bei stromliefernden Ausgängen "Load" an "-" anschließen (gezeigt). 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, ist der Standardwert für dieses Modul 0 V, was bedeutet, dass eine stromziehende Last eingeschaltet wird.

Tabelle A- 16 Schaltpläne der digitalen SBs

SB 1221 Eingabemodul	SB 1222 Ausgabemodul	SB 1223 Eingabe-/Ausgabemodul
<p style="text-align: center;">SB 1221 DI 4 (200 kHz)</p> 	<p style="text-align: center;">SB 1222 DO 4 (200 kHz)</p> 	<p style="text-align: center;">SB 1223 DI 2 / DO 2 (200 kHz)</p> 
<p>① Unterstützt nur stromliefernde Eingänge.</p>	<p>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.</p>	<p>① Unterstützt nur stromliefernde Eingänge.</p> <p>② 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.</p>

Hinweis

Die schnellen (200 kHz) SBs (SB 1221 und SB 1223) unterstützen nur stromziehende Eingänge. Das Standard-SB 1223 unterstützt nur stromliefernde Eingänge.

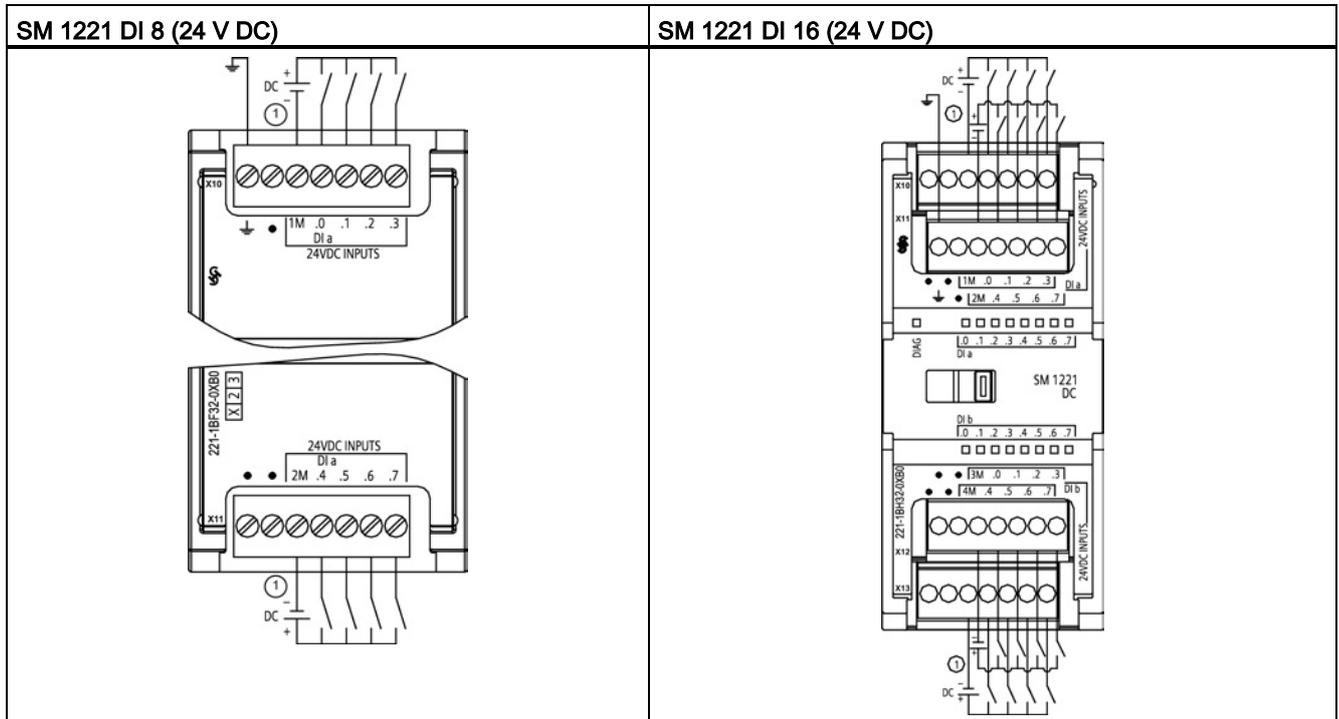
Die schnellen (200 kHz) Ausgänge (SB 1222 und SB 1223) können stromliefernd oder stromziehend sein. Bei stromliefernden Ausgängen "Load" an "-" anschließen (gezeigt). Bei stromziehenden Ausgängen "Load" an "+" anschließen. Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben 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.

A.3.2 SM 1221 Digitaleingabe (DI)

Tabelle A- 17 SM 1221 Digitaleingabe (DI)

Technische Daten	SM 1221 DI 8, 24 V DC	SM 1221 DI 16, 24 V DC
Bestellnummer	6ES7 221-1BF32-0XB0	6ES7 221-1BH32-0XB0
Anzahl der Eingänge (DI)	8	16
Siehe technische Daten (Seite 331).		
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	170 Gramm	210 Gramm
Leistungsverlust	1,5 W	2,5 W
Stromaufnahme	SM-Bus	105 mA
	24 V DC	4 mA/Eingang
		130 mA
		4 mA/Eingang

Tabelle A- 18 Schaltplan der Digitaleingabemodule SM 1221 (DI)



① Bei stromziehenden Eingängen "-" an "M" anschließen (s. Abbildung). Bei stromliefernden Eingängen "+" an "M" anschließen.

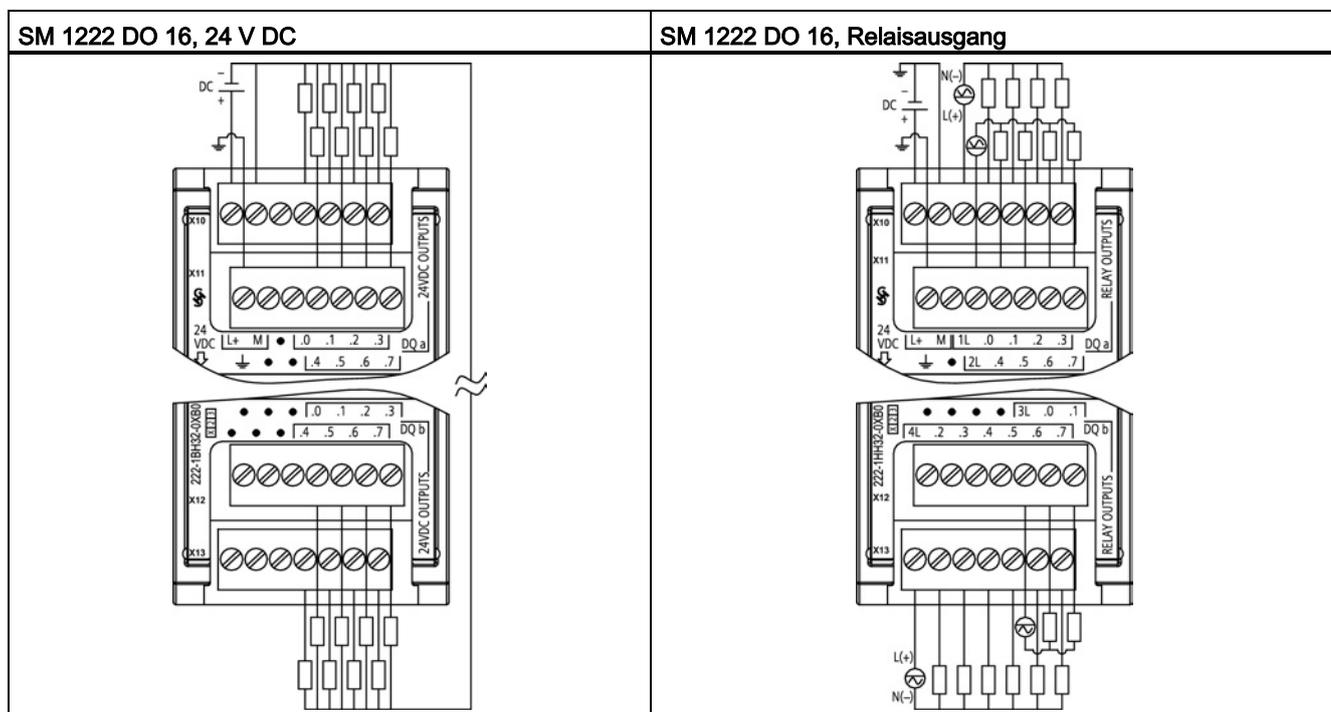
A.3.3 SM 1222 Digitalausgabe (DO)

Tabelle A- 19 SM 1222 Digitalausgabe (DO)

Technische Daten	SM 1222 DO (Relais)	SM 1222 DO (24 V DC)
Bestellnummer	<ul style="list-style-type: none"> • DO 8: 6ES7 222-1HF32-0XB0 • DO 8: Umschaltung: 6ES7 222-1XF32-0XB0 • DO 16: 6ES7 222-1HH32-0XB0 	<ul style="list-style-type: none"> • DO 8: 6ES7 222-1BF32-0XB0 • DO 16: 6ES7 222-1BH32-0XB0
Anzahl der Ausgänge (DO) Siehe technische Daten (Seite 331).	<ul style="list-style-type: none"> • 8 (DO 8 und DO 8 Umschaltung) • 16 (DO 16) 	<ul style="list-style-type: none"> • 8 (DO 8) • 16 (DO 16)
Abmessungen B x H x T (mm)	<ul style="list-style-type: none"> • DO 8 und DO 16: 45 x 100 x 75 • DO 8 Umschaltung: 70 x 100 x 75 	45 x 100 x 75
Gewicht	<ul style="list-style-type: none"> • DO 8: 190 Gramm • DO 8 Umschaltung: 310 Gramm • DO 16: 260 Gramm 	<ul style="list-style-type: none"> • DO 8: 180 Gramm • DO 16: 220 Gramm

Technische Daten		SM 1222 DO (Relais)	SM 1222 DO (24 V DC)
Leistungsverlust		<ul style="list-style-type: none"> DO 8: 4,5 W DO 8 Umschaltung: 5 W DO 16: 8,5 W 	<ul style="list-style-type: none"> DO 8: 1,5 W DO 16: 2,5 W
Stromaufnahme	SM-Bus	<ul style="list-style-type: none"> DO 8: 120 mA DO 8 Umschaltung: 140 mA DO 16: 135 mA 	<ul style="list-style-type: none"> DO 8: 120 mA DO 16: 140 mA
	24 V DC	<ul style="list-style-type: none"> DO 8 und DO 16: 11 mA/Relaispule DO 8 Umschaltung: 16,7 mA/Relaispule 	<ul style="list-style-type: none"> DO 8: -- DO 16: --

Tabelle A- 20 Schaltplan der Digitalausgabemodule SM 1222 (DO)

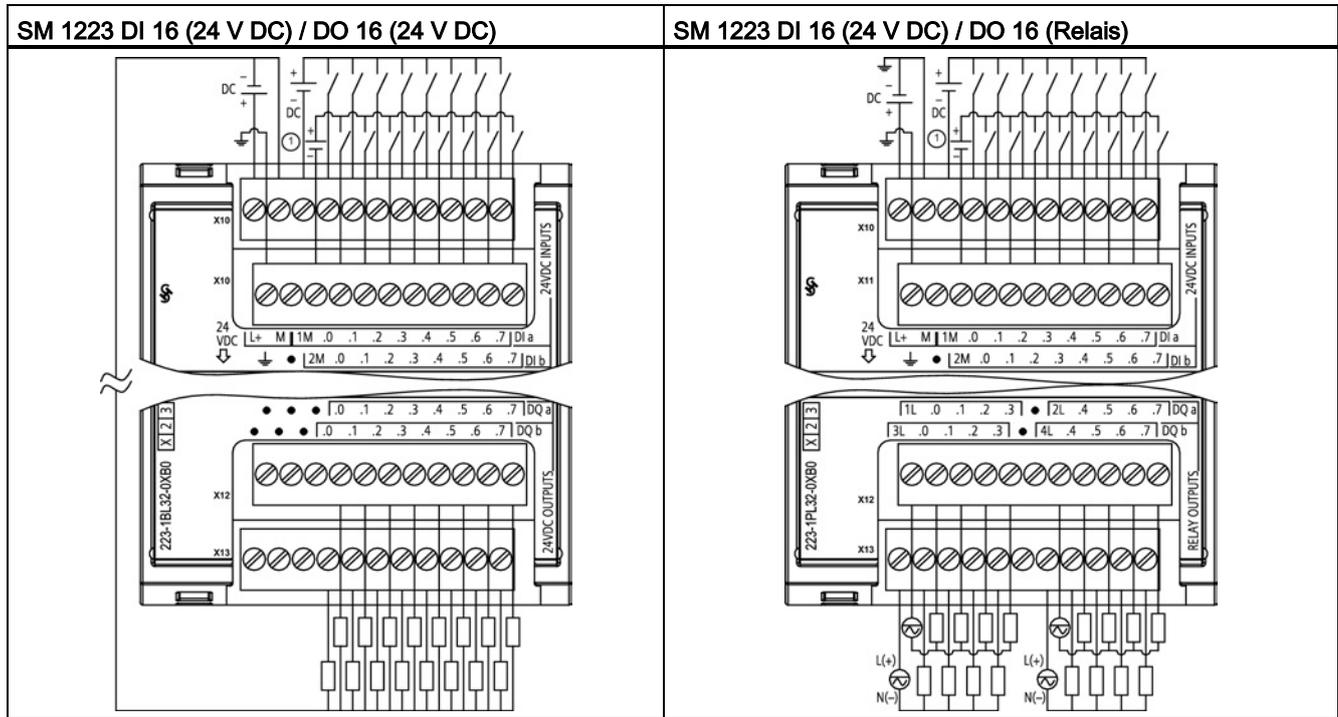


A.3.4 SM 1223 V-DC-Digitaleingabe/-ausgabe (DI/DO)

Tabelle A- 21 SM 1223 Kombinationsmodul Digitaleingabe/-ausgabe (DI/DO)

Technische Daten	SM 1223 DI (24 V DC) / DO (Relais)	SM 1223 DI (24 V DC) / DO (24 V DC)								
Bestellnummer	DI 8 / DO 8: 6ES7 223-1PH32-0XB0 DI 16 / DO 16: 6ES7 223-1PL32-0XB0	DI 8 / DO 8: 6ES7 223-1BH32-0XB0 DI 8 / DO 8: 6ES7 223-1BL32-0XB0								
Anzahl der Eingänge/Ausgänge (DI/DO) Siehe technische Daten (Seite 331).	<ul style="list-style-type: none"> Eingänge: 8 oder 16 (24 V DC) Ausgänge: 8 oder 16 (Relais) 	<ul style="list-style-type: none"> Eingänge: 8 oder 16 (24 V DC) Ausgänge: 8 oder 16 (24 V DC) 								
Abmessungen B x H x T (mm)	<ul style="list-style-type: none"> DI 8 / DO 8: 45 x 100 x 75 DI 16 / DO 16: 70 x 100 x 75 	<ul style="list-style-type: none"> DI 8 / DO 8: 45 x 100 x 75 DI 16 / DO 16: 70 x 100 x 75 								
Gewicht	<ul style="list-style-type: none"> DI 8 / DO 8: 230 Gramm DI 16 / DO 16: 350 Gramm 	<ul style="list-style-type: none"> DI 8 / DO 8: 210 Gramm DI 16 / DO 16: 310 Gramm 								
Leistungsverlust	<ul style="list-style-type: none"> DI 8 / DO 8: 5,5 W DI 16 / DO 16: 10 W 	<ul style="list-style-type: none"> DI 8 / DO 8: 2,5 W DI 16 / DO 16: 4,5 W 								
Stromaufnahme	<table border="0"> <tr> <td>SM-Bus</td> <td> <ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 180 mA </td> </tr> <tr> <td>24 V DC</td> <td>4 mA / Eingang 11 mA / Relaispule</td> </tr> </table>	SM-Bus	<ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 180 mA 	24 V DC	4 mA / Eingang 11 mA / Relaispule	<table border="0"> <tr> <td></td> <td> <ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 185 mA </td> </tr> <tr> <td></td> <td>4 mA / Eingang</td> </tr> </table>		<ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 185 mA 		4 mA / Eingang
SM-Bus	<ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 180 mA 									
24 V DC	4 mA / Eingang 11 mA / Relaispule									
	<ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 185 mA 									
	4 mA / Eingang									

Tabelle A- 22 Schaltpläne des SM 1223 Kombinationsmoduls DI/DO



Ⓛ Bei stromziehenden Eingängen "-" an "M" anschließen (s. Abbildung). Bei stromliefernden Eingängen "+" an "M" anschließen.

A.3.5 SM 1223 120/230 V-AC-Eingabe/Relaisausgang

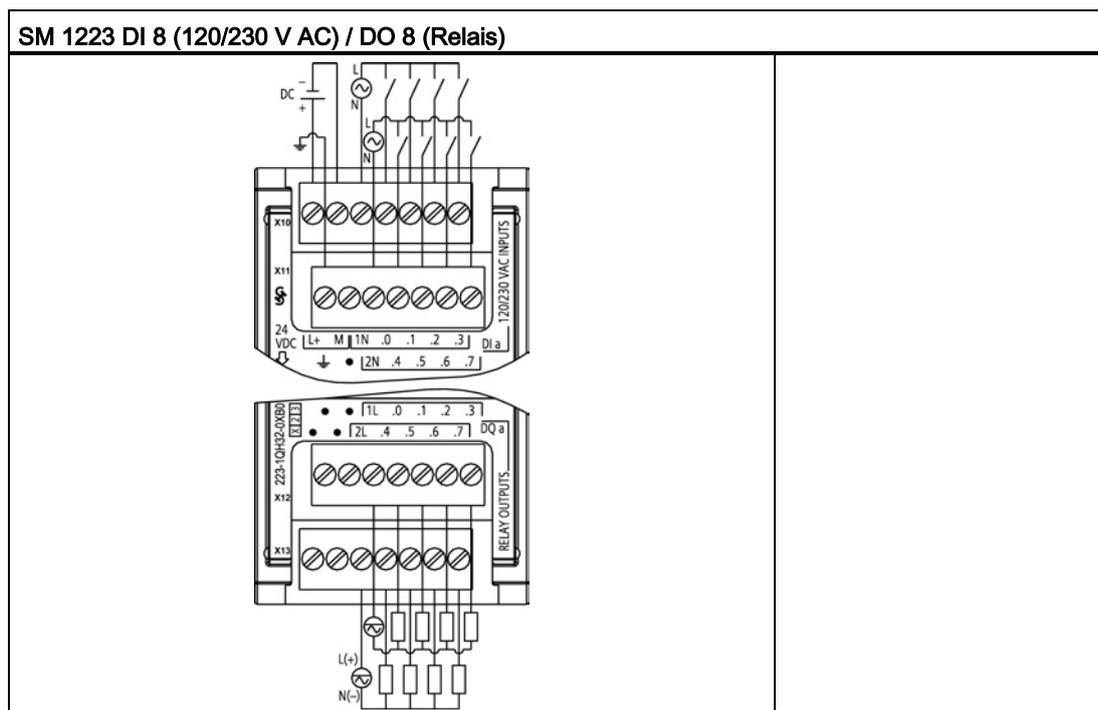
Tabelle A- 23 SM 1223 Kombinationsmodul V-AC-Digitaleingabe/-ausgabe (DI/DO)

Technische Daten	SM 1223 DI (120/230 V AC) / DO (Relais)
Bestellnummer	DI 8 / DO 8: 6ES7 223-1QH32-0XB0
Anzahl der Eingänge/Ausgänge (DI/DO)	Eingänge: 8 (120/230 V AC) Siehe technische Daten für 120/230 V AC-Eingänge (Seite 333). Ausgänge: 8 (Relais) Siehe technische Daten für die Digitalausgänge (Seite 334).
Abmessungen B x H x T (mm)	45 x 100 x 75
Gewicht	190 Gramm
Leistungsverlust	7,5 W
Stromaufnahme	SM-Bus 120 mA
	24 V DC 11 mA / Relaisspule

Hinweis

Das Signalmodul SM 1223 DI 8 x 120/230 V AC, DO 8 x Relais (6ES7 223-1QH32-0XB0) ist für den Einsatz in Umgebungen gemäß Class 1, Division 2, Gas Group A, B, C, D, Temperature Class T4 Ta = 40 °C zugelassen.

Tabelle A- 24 Schaltplan des SM 1223 DI 8 (120/230 V AC) / DO 8 (Relais)



A.4 Technische Daten der digitalen Eingänge und Ausgänge

A.4.1 24-V-DC-Digitaleingabe (DI)

Tabelle A- 25 Technische Daten der digitalen Eingänge (DI)

Technische Daten	CPU, SM und SB	Schnelles SB (200 kHz)
Typ	<ul style="list-style-type: none"> CPU und SM: IEC Typ 1 stromziehend (stromziehend/stromliefernd) SB 1223: IEC Typ 1 stromziehend (nur stromziehend) 	SB 1221 200 kHz und SB 1223 200 kHz: Quelle
Nennspannung	24 V DC bei 4 mA, Nennwert	24 V DC SB: 24 V DC bei 7 mA, Nennwert 5 V DC SB: 5 V DC bei 15 mA, Nennwert

Technische Daten

A.4 Technische Daten der digitalen Eingänge und Ausgänge

Technische Daten	CPU, SM und SB	Schnelles SB (200 kHz)
Zulässige Dauerspannung	max. 30 V DC	24 V DC SB: 28,8 V DC 5 V DC SB: 6 V DC
Stoßspannung	35 V DC für 0,5 s	24 V DC SB: 35 V DC für 0,5 s 5 V DC SB: 6 V
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	24 V DC SB: L+ minus 10 V DC bei 2,9 mA 5 V DC SB: L+ minus 2,0 V DC bei 5,1 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA	24 V DC SB: L+ minus 5 V DC bei 1,4 mA 5 V DC SB: L+ minus 1,0 V DC bei 2,2 mA
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	<ul style="list-style-type: none"> • CPU: 1 • SM 1221 DI 8: 2 • SM 1221 DI 16: 4 • SB 1223 DI 2: 1 • SM 1223: 2 	<ul style="list-style-type: none"> • SB 1221 DI 4: 1 • SB 1223 DI 2: 1
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	<ul style="list-style-type: none"> • SM 1221 und SM 1223 DI 8: 8 • SM 1221 und SM 1223 DI 16: 16 • SB 1223 DI 2: 2 	<ul style="list-style-type: none"> • SB 1221 DI 4: 4 • SB 1223 DI 2: 2
Leitungslänge (Meter)	<ul style="list-style-type: none"> • 500 m geschirmt, 300 m ungeschirmt • CPU: 50 m geschirmt für HSC 	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- 26 HSC Eingangstaktfrequenzen (max.)

Technische Daten	Einphasenzähler	A/B-Zähler
CPU 1211C	100 kHz	80 kHz
CPU 1212C	100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Ea.7)	80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Ea.7)
CPU 1214C, CPU 1215C	100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Ea.5)	80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Ea.5)
CPU 1217C	1 MHz (DEb.2 bis DEb.5)	1 MHz (DEb.2 bis DEb.5)
Schnelles SB (200 kHz)	200 kHz	160 kHz
SB mit Standardgeschwindigkeit	30 kHz	20 kHz

¹ Pegel logisch 1 = 15 bis 26 V DC

A.4.2 120/230 V-AC-Digitaleingabe

Tabelle A- 27 120/230 V-AC-Digitaleingabe

Technische Daten	SM	
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 für 1 Minute	
Potenzialgetrennte Gruppen ¹	4	
Eingabeverzögerungszeiten	<ul style="list-style-type: none"> • Typisch: 0,2 bis 12,8 ms, vom Anwender einstellbar • Maximum: -- 	
Anschluss 2-Draht-Näherungssensor (Bero) (max.)	1 mA	
Leitungslänge	Ungeschirmt	300 Meter
	Geschirmt	500 Meter
Anzahl gleichzeitig eingeschalteter Eingänge	8	

¹ Kanäle in einer Gruppe müssen die gleiche Phase haben.

A.4.3 Digitalausgänge (DO)

Tabelle A- 28 Technische Daten der digitalen Ausgänge (DO)

Technische Daten	Relais (CPU und SM)	24 V DC (CPU, SM und SB)	200 kHz 24 V DC (SB)
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)	MOSFET, elektronisch (stromziehend/stromliefernd)
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 ¹ 4,25 bis 6,0 V DC ²
Signal logisch 1 bei max. Strom	-/-	min. 20 V DC	L+ minus 1,5 V ¹ L+ minus 0,7 V ²
Signal logisch 0 bei 10 kΩ Last	-/-	CPU: min. 20 V DC, max. 0,1 V DC SB: max. 0,1 V DC SM DC: max. 0,1 V DC	max. 1,0 V DC ¹ max. 0,2 V DC ²
Strom (max.)	2,0 A	0,5 A	0,1 A
Lampenlast	30 W DC/200 W AC	SB: 5 W	-/-
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω	max. 11 Ω ¹ oder max. 7 Ω ²
Widerstand bei AUS	-/-	-/-	max. 6 Ω ¹ oder max. 0,2 Ω ²
Kriechstrom pro Ausgang	-/-	max. 10 µA	-/-
Frequenz Impulsgenerator	CPU: -/- ³	CPU: max. 100 kHz, min. 2 Hz ⁴ SB: max. 20 kHz, min. 2 Hz ⁵	max. 200 kHz, min. 2 Hz
Einschaltstrom	7 A bei geschlossenen Kontakten	CPU: 8 A für max. 100 ms SB: 5 A für max. 100 ms SM: 8 A für max. 100 ms	0,11 A
Überlastschutz	Nein	Nein	Nein
Elektrische Trennung (Feld zu Logik)	Spule zu Kontakt: 1500 V AC für 1 Minute Spule zu Logik: Keine	500 V AC für 1 Minute	500 V AC für 1 Minute
Potentialgetrennte Gruppen	<ul style="list-style-type: none"> • CPU 1211C: 1 • CPU 1212C: 2 • CPU 1214C: 2 • CPU 1215C: 2 • SM DO 8: 2 • SM DO 8 Umschaltung: 8 • SM DO 16: 4 	<ul style="list-style-type: none"> • CPU: 1 • SB: 1 • SM (DO 8): 1 • SM (DO 16): 1 	¹⁵
Isolationswiderstand	min. 100 MΩ, wenn neu	-/-	-/-
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	-/-	-/-

Technische Daten	Relais (CPU und SM)	24 V DC (CPU, SM und SB)	200 kHz 24 V DC (SB)
Strom je Leiter	CPU: SM-Relais: <ul style="list-style-type: none"> SM 1222: 10 A (DO 8 und DO 16) SM 1223 DI 8 / DO 8 Relais: 10 A SM 1223 DI 16 / DO 16 Relais: 8 A 	CPU: <ul style="list-style-type: none"> SB: 1 A SM DO 8: 4 A SM DO 16: 8 A 	0,4 A
Induktive Klemmspannung	-/-	L+ minus 48 V, 1 W Verlustleistung	Keine
Maximale Schaltfrequenz Relais	1 Hz	-/-	-/-
Schaltverzögerung	max. 10 ms	CPU: <ul style="list-style-type: none"> Aa.0 bis Aa.3: max. 1,0 μs von Aus nach Ein; max. 3,0 μs von Ein nach Aus Aa.4 bis Ab.1: max. 50 μs von Aus nach Ein; max. 200 μs von Ein nach Aus SB: max. 2 μ s von Aus nach Ein; max. 10 μ s von Ein nach Aus SM: max. 50 μ s von Aus nach Ein; max. 200 μ s von Ein nach Aus	1,5 μ s + 300 s steigend ¹ 1,5 μ s + 300 ns fallend ¹ 200 ns + 300 ns steigend ² 200 ns + 300 ns fallend ²
Mechanische Lebensdauer (ohne Last)	Relais: 10.000.000 Schaltspiele auf/zu	-/-	-/-
Lebensdauer der Kontakte bei Nennlast	Relais: 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)
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	500 m geschirmt, 150 m ungeschirmt	50 m, geschirmtes, verdrilltes Leiterpaar

¹ SB, 24 V DC, 200 KHz

² SB, 5 V DC, 200 KHz

³ Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

⁴ Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

⁵ SB 1223, 200 kHz, DI 2/DO 2: Keine Trennung gegen Eingänge

A.5 Analoge Erweiterungsmodule

Eine vollständige Liste der Module für die S7-1200 finden Sie im S7-1200 Systemhandbuch oder auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

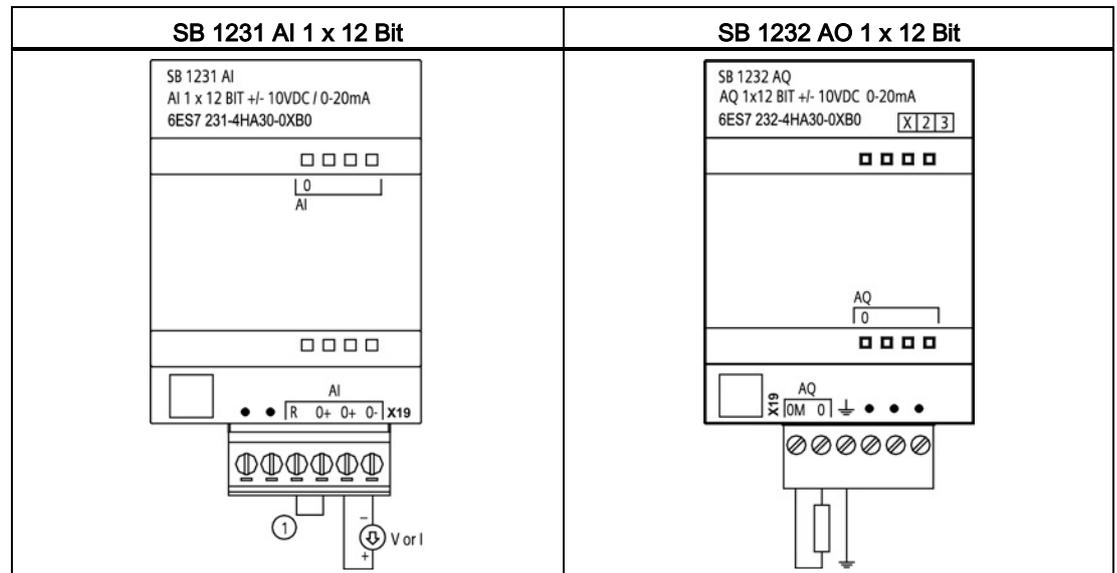
A.5.1 SB 1231 und SB 1232 Analogeingabe (AI) und Analogausgabe (AO)

Tabelle A- 29 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 12 Bit ¹	SB 1232 AO 1 x 12 Bit
Bestellnummer	6ES7 231-4HA30-0XB0	6ES7 232-4HA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
Gewicht	35 Gramm	40 Gramm
Leistungsverlust	0,4 W	1,5 W
Stromaufnahme (SM-Bus)	55 mA	15 mA
Stromaufnahme (24 V DC)	Keine	40 mA (ohne Last)
Anzahl der Eingänge/Ausgänge	1	1
Typ	Spannung oder Strom (differential)	Spannung oder Strom

¹ Um das SB 1231 AI 1 x Analogeingabe zu verwenden, benötigen Sie CPU-Firmware ab V2.0.

Tabelle A- 30 Schaltpläne der analogen SBs



① "R" und "0+" für Strom anschließen.

A.5.2 SM 1231 Analogeingabe (AI)

Tabelle A- 31 SM 1231 Analogeingabe (AI)

Technische Daten	SM 1231 AI 4 x 13 Bit	SM 1231 AI 8 x 13 Bit	SM 1231 AI 4 x 16 Bit
Bestellnummer (MLFB)	6ES7 231-4HD32-0XB0	6ES7 231-4HF32-0XB0	6ES7 231-5ND32-0XB0
Anzahl der Eingänge	4 Eingänge (AI)	8 Eingänge (AI)	4 Eingänge
Typ	Spannung oder Strom (differential), wählbar in Gruppen zu je 2	Spannung oder Strom (differential), wählbar in Gruppen zu je 2	Spannung oder Strom (differential)
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	180 Gramm	180 Gramm
Leistungsverlust	1,5 W	1,5 W	1,8 W
Stromaufnahme (SM-Bus)	80 mA	90 mA	80 mA
Stromaufnahme (24 V DC)	45 mA	45 mA	65 mA

A.5.3 SM 1232 Analogausgabe (AO)

Tabelle A- 32 SM 1232 Analogausgabe (AO)

Technische Daten	SM 1232 AO 2 x 14 Bit	SM 1232 AO 4 x 14 Bit
Bestellnummer (MLFB)	6ES7 232-4HB32-0XB0	6ES7 232-4HD32-0XB0
Anzahl und Art der Ausgänge	2 Ausgänge (AO)	4 Ausgänge (AO)
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	180 Gramm
Leistungsverlust	1,5 W	1,5 W
Stromaufnahme (SM-Bus)	80 mA	80 mA
Stromaufnahme (24 V DC)	45 mA (ohne Last)	45 mA (ohne Last)

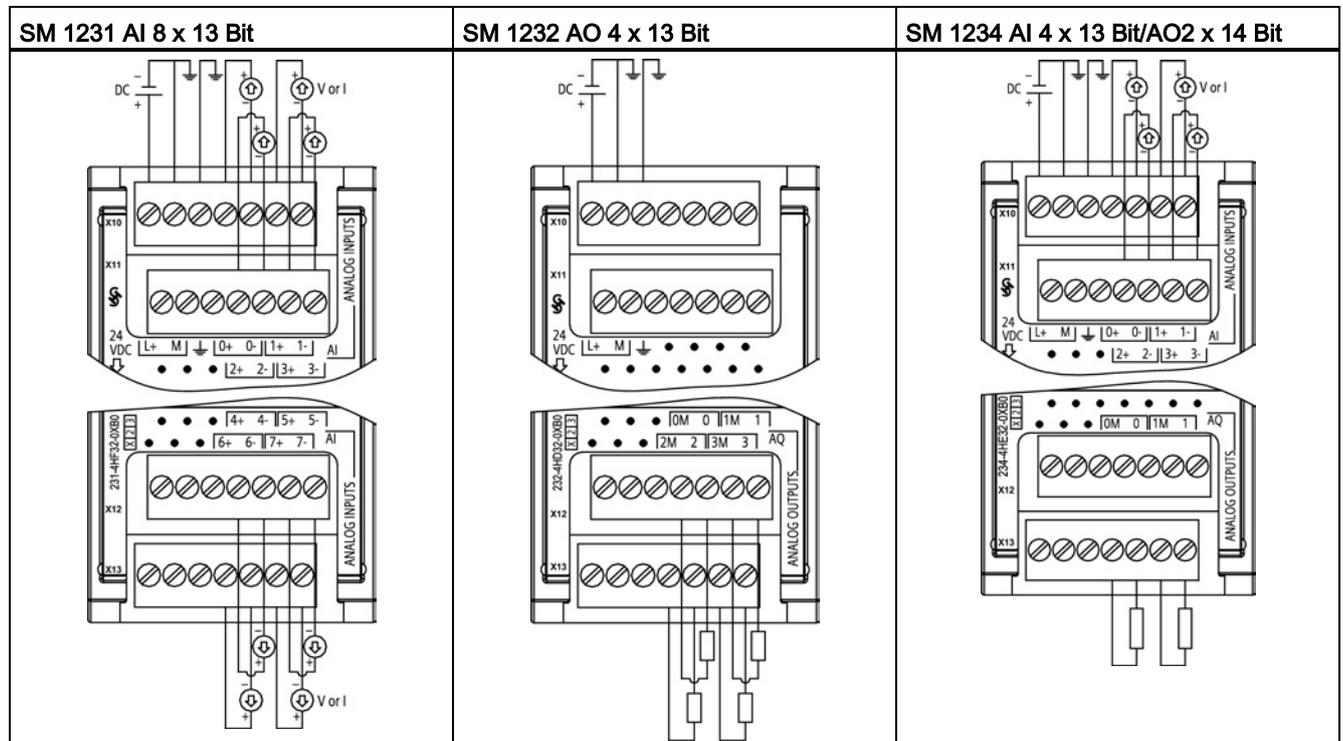
A.5.4 SM 1234 Analogeingabe/-ausgabe (AI/AO)

Tabelle A- 33 SM 1234 Kombinationsmodul Analogeingabe/-ausgabe (AI/AO)

Technische Daten	SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit
Bestellnummer (MLFB)	6ES7 234-4HE32-0XB0
Anzahl der Eingänge	4 Eingänge (AI)
Typ	Spannung oder Strom (differential), wählbar in Gruppen zu je 2
Anzahl der Ausgänge	2 Ausgänge (AO)
Typ	Spannung oder Strom (differential)
Abmessungen B x H x T (mm)	45 x 100 x 75
Gewicht	220 Gramm
Leistungsverlust	2,0 W
Stromaufnahme (SM-Bus)	80 mA
Stromaufnahme (24 V DC)	60 mA (ohne Last)

A.5.5 Schaltpläne SM 1231 (AI), SM 1232 (AO) und SM 1234 (AI/AO)

Tabelle A- 34 Schaltpläne der analogen SMs



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.6 BB 1297 Batterieboard

BB 1297 Batterieboard

Tabelle A- 35 Allgemeine technische Daten

Technische Daten	BB 1297 Batterie
Bestellnummer	6ES7 297-0AX30-0XA0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	28 Gramm
Haltezeit Echtzeituhr	Ca. 1 Jahr
Batterietyp	CR1025 ¹
Wartungs-LED der CPU	Zeigt an, wann die Batterie auszutauschen ist
Anwenderprogramm	Anwendung/System kann den Batteriezustand auswerten

¹ Im S7-1200 Systemhandbuch, Kapitel 2, finden Sie Informationen zum Installieren des BB 1297 oder zum Auswechseln einer Batterie im BB.

Das Batterieboard BB 1297 wird in Anwendungen eingesetzt, in denen die Pufferung der Echtzeituhr über einen Monat hinausgeht. Die Funktionen des Batterieboards BB 1297 werden im Folgenden gezeigt:

- Unterstützung der Echtzeituhr bei ausgeschaltetem PLC-Gerät. Die S7-1200 CPU unterstützt in Verbindung mit dem Batterieboard BB 1297 die Pufferung der Echtzeituhr bei ausgeschalteter Anwendung bis zu einem Jahr.
- Es kann nur ein Batterieboard BB 1297 oder ein anderes SB genutzt werden.
- Das Stecken bzw. Auswechseln im laufenden Betrieb ist nicht zulässig. Das Batterieboard BB 1297 kann nur im ausgeschalteten Zustand der CPU ausgetauscht oder gesteckt werden. Wird das BB 1297 bei ausgeschalteter CPU entfernt, um die eigentliche Batterie auszuwechseln, puffert der interne Hochleistungskondensator die Echtzeituhr, während die Batterie ausgetauscht wird.
- Die Wartungs-LED der CPU zeigt an, wann die Batterie auszutauschen ist.
- Über das Anwenderprogramm können Sie den Zustand der Batterie und des Batterieboards überwachen oder abfragen, und es kann eine Anwendermeldung auf einem HMI-Gerät oder einem Webserver angezeigt werden.

A.7 Technische Daten der analogen E/A

A.7.1 Technische Daten der analogen Eingänge (CPU, SM und SB)

Tabelle A- 36 Technische Daten der Analogeingänge (AI)

Technische Daten	CPU	SB	SM
Typ	Spannung (Eintakteingang)	Spannung oder Strom (differential)	Spannung oder Strom (differential), wählbar in Gruppen zu je 2
Bereich	0 bis 10 V	±10 V, ±5 V, ±2,5, 0 bis 20 mA oder 4 mA bis 20 mA	±10 V, ±5 V, ±2,5 V, 0 bis 20 mA oder 4 mA bis 20 mA
Auflösung	10 Bit	11 Bit + Vorzeichenbit	12 Bit + Vorzeichenbit
Vollausschlag (Datenwort)	0 bis 27648	-27.648 bis 27.648	-27.648 bis 27.648
Genauigkeit (25 °C / -20 bis 60 °C)	3,0 % / 3,5 % des Vollausschlags	±0,3 % / ±0,6 % des Vollausschlags	±0,1% / ±0,2% des Vollausschlags
Überschwing-/Unterschwingbereich (Datenwort) (Siehe Hinweis 1)	Spannung: 27.649 bis 32.511 Strom: -/-	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649 / 0 bis -4864	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649 / 0 bis -4864
Überlauf/Unterlauf (Datenwort) (Siehe Hinweis 1)	Spannung: 32.512 bis 32.767 Strom: -/-	Spannung: 32.767 bis 32.512 / -32.513 bis -32.768 Strom: 32.767 bis 32.512 / -4865 bis -32.768	Spannung: 32.767 bis 32.512 / -32.513 bis -32.768 Strom: 32.767 bis 32.512 / -4865 bis -32.768
Maximale Stehspannung/-strom	35 V DC (Spannung)	±35 V / ±40 mA	±35 V / ±40 mA
Glättung (Siehe Hinweis 2)	Keine, schwach, mittel oder stark	Keine, schwach, mittel oder stark	Keine, schwach, mittel oder stark
Rauschunterdrückung (Siehe Hinweis 2)	10, 50 oder 60 Hz	400, 60, 50 oder 10 Hz	400, 60, 50 oder 10 Hz
Messprinzip	Istwertumwandlung	Istwertumwandlung	Istwertumwandlung
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz	40 dB, Nennwert bei 60 Hz	40 dB, Nennwert bei 60 Hz
Betriebssignalbereich (Signalplus Gleichtaktspannung)	Kleiner als +12 V und größer als 0 V	Kleiner als +35 V und größer als -35 V	Kleiner als +12 V und größer als -12 V
Lastimpedanz	Eintakteingang: ≥100 kΩ	Differenz: 220 kΩ (Spannung), 250 Ω (Strom) Gleichtakt: 55 kΩ (Spannung), 55 kΩ (Strom)	Differenz: 9 MΩ (Spannung), 250 Ω (Strom) Gleichtakt: 4,5 MΩ (Spannung), 4,5 MΩ (Strom)

A.7 Technische Daten der analogen E/A

Technische Daten	CPU	SB	SM
Elektrische Trennung (Feld zu Logik)	Keine	Keine	Keine
Leitungslänge (Meter)	100 m, geschirmtes, verdrehtes Leiterpaar	100 m, verdreht und geschirmt	100 m, verdreht und geschirmt
Diagnose	Überlauf/Unterlauf	Überlauf/Unterlauf	Überlauf/Unterlauf 24-V-DC-Niederspannung

Hinweis 1: Die Bereiche für Überschwngen/Unterschwngen und Überlauf/Unterlauf ermitteln Sie anhand der Messbereiche der analogen Ausgänge für Spannung und Strom (Seite 342).

Hinweis 2: Die Werte für Glättung und Rauschunterdrückung ermitteln Sie anhand der Schrittantwortzeiten (Seite 343).

A.7.2 Eingangsmessbereiche (AI) für Spannung und Strom

Tabelle A- 37 Darstellung Analogeingang für Spannung (SB und SM)

System		Messbereich Spannung				
Dezimal	Hexadezimal	±10 V	±5 V	±2,5 V	±1,25 V	
32767	7FFF	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	

Tabelle A- 38 Darstellung Analogeingang für Strom (SB und SM)

System		Messbereich Strom		
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 mA	
32767	7FFF	23,70 mA	22,96 mA	Überlauf
32512	7F00			
32511	7EFF	23,52 mA	22,81 mA	Überschwingbereich
27649	6C01			
27648	6C00	20 mA	20 mA	Nennbereich
20736	5100	15 mA	16 mA	

System		Messbereich Strom		
Dezimal	Hexadezimal	0 mA bis 20 mA	4 mA bis 20 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	
-4865	ECFF			Unterlauf
-32768	8000			

Tabelle A- 39 Darstellung Analogeingang für Spannung (CPU 1215C und CPU 1217C)

System		Messbereich Spannung	
Dezimal	Hexadezimal	0 bis 10 V	
32767	7FFF	11,851 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.7.3 Schrittantwort der analogen Eingänge (AI)

Die folgende Tabelle zeigt die Schrittantwortzeiten der analogen Eingänge (AI) von CPU, SB und SM.

Tabelle A- 40 Schrittantwort (ms) der analogen Eingänge

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	CPU	-/-	63	65	130
	SB	4.5	18.7	22.0	102
	SM	4	18	22	100
Schwach (4 Zyklen): 4 Abtastwerte	CPU	-/-	84	93	340
	SB	10.6	59.3	70.8	346
	SM	9	52	63	320
Mittel (16 Zyklen): 16 Abtastwerte	CPU	-/-	221	258	1210
	SB	33.0	208	250	1240
	SM	32	203	241	1200

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)
Stark (32 Zyklen): 32 Abtastwerte	CPU	-/-	424	499	2410
	SB	63.0	408	490	2440
	SM	61	400	483	2410
Abtastrate	CPU	-/-	4.17	5	25
	SB	0.156	1.042	1.250	6.250

A.7.4 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A- 41 Abtastzeit und Aktualisierungszeit von SM und CPU

Unterdrückungsfrequenz (Integrationszeit)	Abtastzeit	Aktualisierungszeit für alle Kanäle		
		4-kanaliges SM	8-kanaliges SM	CPU AI
400 Hz (2,5 ms)	0,625 ms ¹	2,5 ms	10 ms	-/- ms
60 Hz (16,6 ms)	4,170 ms	4,17 ms	4,17 ms	4,17 ms
50 Hz (20 ms)	5,000 ms	5 ms	5 ms	5 ms
10 Hz (100 ms)	25,000 ms	25 ms	25 ms	25 ms

¹ Die Abtastrate eines 8-kanaligen SM beträgt 1,250 ms.

Tabelle A- 42 Abtastzeit und Aktualisierungszeit von SBs

Unterdrückungsfrequenz (Integrationszeit)	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.7.5 Technische Daten der analogen Ausgänge

Tabelle A- 43 Technische Daten der analogen Ausgänge (SB und SM)

Technische Daten	SB	SM
Typ	Spannung oder Strom	Spannung oder Strom
Bereich	± 10 V, 0 bis 20 mA oder 4 bis 20 mA	± 10 V, 0 bis 20 mA oder 4 bis 20 mA
Auflösung	Spannung: 12 Bit Strom: 11 Bit	Spannung: 14 Bit Strom: 13 Bit
Vollausschlag (Datenwort) (Siehe Hinweis 1)	Spannung: -27.648 bis 27.648 Strom: 0 bis 27.648	Spannung: -27.648 bis 27.648 Strom: 0 bis 27.648
Genauigkeit (25 °C/-20 bis 60 °C)	$\pm 0,5$ % / ± 1 % des Vollausschlags	$\pm 0,3$ % / $\pm 0,6$ % 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)	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: $\leq 600 \Omega$	Spannung: $\geq 1000 \Omega$ Strom: $\leq 600 \Omega$
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine	Keine
Leitungslänge (Meter)	100 m, verdreht und geschirmt	100 m, verdreht und geschirmt
Diagnose	<ul style="list-style-type: none"> • Überlauf/Unterlauf • Erdschluss (nur Spannungsmodus) • Drahtbruch (nur Strommodus) 	<ul style="list-style-type: none"> • Überlauf/Unterlauf • Erdschluss (nur Spannungsmodus) • Drahtbruch (nur Strommodus) • 24-V-DC-Niederspannung
Hinweis 1: Siehe Abschnitt der Ausgangsbereiche für Spannung und Strom (Seite 346) im Vollausschlagsbereich.		

A.7.6 Ausgangsmessbereiche (AO) für Spannung und Strom

Tabelle A- 44 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		
-32512	8100	-11,76 V	
-32513	80FF	Siehe Hinweis 1	
-32768	8000	Siehe Hinweis 1	

¹ Bei Überlauf oder Unterlauf nehmen die Analogausgänge den Ersatzwert für den Betriebszustand STOP an.

Tabelle A- 45 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	
-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.

Tabelle A- 46 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	
32511	7EFF	23,52 mA	Überschwingbereich
27649	6C01		
27648	6C00	20 mA	Bemessungsbereich
20736	5100	15 mA	
34	22	12 mV	
0	0	0 mA	
Negative Werte		Negative Werte werden nicht unterstützt	

¹ Bei Überlauf verhalten sich die Analogausgänge entsprechend den eingestellten Eigenschaften der Gerätekonfiguration. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: "Ersatzwert aufschalten" oder "Letzten Wert halten".

A.8 RTD- und Thermoelementmodule

Die Thermoelementmodule (TC) (SB 1231 TC und SM 1231 TC) messen den Wert der an die Analogeingänge angeschlossenen Spannung. Dies kann entweder ein Temperaturwert eines Thermoelements oder ein Spannungswert sein.

- Bei Spannungsmessung beträgt der Messbereichsendwert im Nennbereich 27648 dezimal.
- Bei Temperaturmessung wird der Messwert in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).

Die RTD-Module (SB 1231 RTD und SM 1231 RTD) messen den Wert des an die Analogeingänge angeschlossenen Widerstands. Dies kann entweder ein Temperatur- oder ein Widerstandswert sein.

- Bei Widerstandsmessung beträgt der Messbereichsendwert im Nennbereich 27648 dezimal.
- Bei Temperaturmessung wird der Messwert in Grad, multipliziert mit zehn, ausgegeben (Beispiel: 25,3 Grad werden als Dezimalwert 253 dargestellt).

Die RTD-Module unterstützen Messungen über 2-Leiter-, 3-Leiter- und 4-Leiter-Anschlüsse zum Geberwiderstand.

Hinweis

Für alle aktivierten Kanäle ohne angeschlossenen Geber melden die RTD- und TC-Module 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.

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.

A.8.1 Technische Daten SB 1231 RTD und SB 1231 TC

Hinweis

Um die TC- und RTD-SBs nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 47 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 x 16 Bit TC	SB 1231 AI 1 x 16 Bit RTD
Bestellnummer	6ES7 231-5QA30-0XB0	6ES7 231-5PA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
Gewicht	35 Gramm	35 Gramm
Leistungsverlust	0,5 W	0,7 W
Stromaufnahme (SM-Bus)	5 mA	5 mA
Stromaufnahme (24 V DC)	20 mA	25 mA

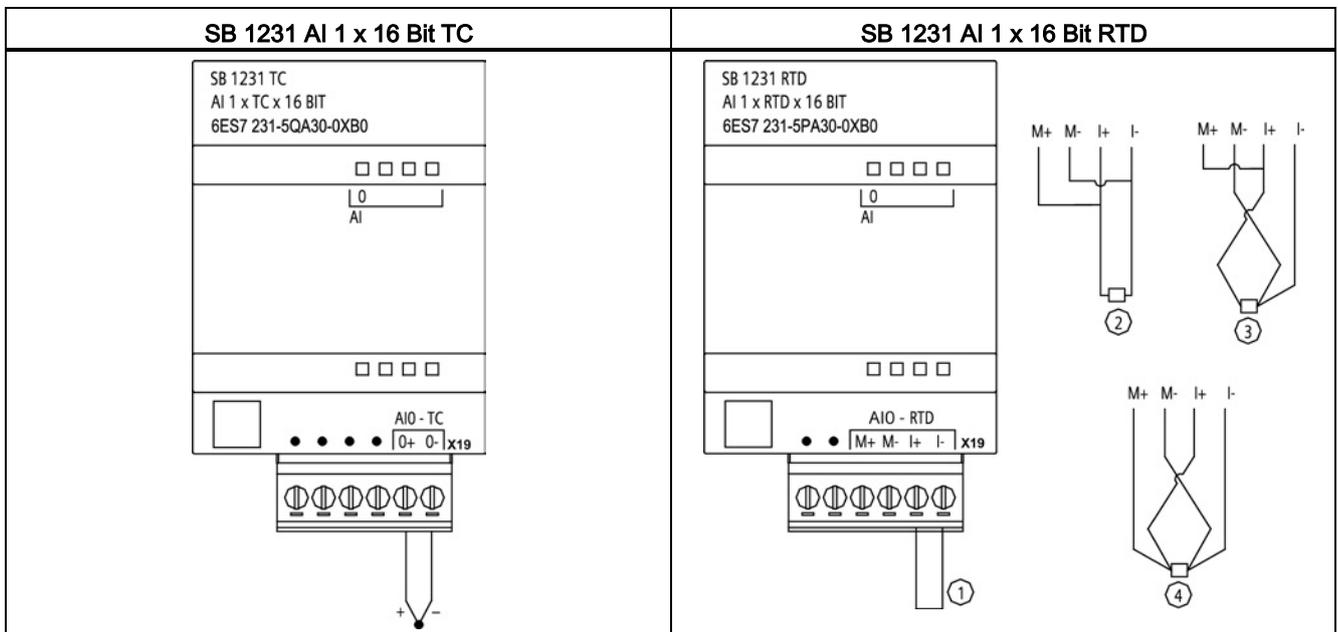
Technische Daten	SB 1231 AI 1 x 16 Bit TC	SB 1231 AI 1 x 16 Bit RTD
Anzahl der Eingänge (Seite 353)	1	1
Typ	Potentialfrei, TC und mV	Modulreferenz RTD und Ω
Diagnose	<ul style="list-style-type: none"> Überlauf/Unterlauf^{1, 2} Drahtbruch³ 	<ul style="list-style-type: none"> Überlauf/Unterlauf^{1, 2} Drahtbruch³

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 RTD: 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.

Tabelle A- 48 Schaltpläne SB 1231 TC und RTD



- ① Nicht belegten RTD-Eingang zurückschleifen
- ② 2-Draht-RTD ③ 3-Draht-RTD ④ 4-Draht-RTD

A.8.2 Technische Daten SM 1231 RTD

Tabelle A- 49 Allgemeine technische Daten

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x 16 Bit
Bestellnummer	6ES7 231-5PD32-0XB0	6ES7 231-5PF32-0XB0
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	1,5 W
Stromaufnahme (SM-Bus)	80 mA	90 mA
Stromaufnahme ¹ (24 V DC)	40 mA	40 mA
Anzahl der Eingänge (Seite 353) Typ	4 Modulreferenz RTD und Ω	8 Modulreferenz RTD und Ω
Diagnose	<ul style="list-style-type: none"> • Überlauf/Unterlauf ^{2,3} • 24-V-DC-Niederspannung² • Drahtbruch (nur Strommodus) ⁴ 	<ul style="list-style-type: none"> • Überlauf/Unterlauf ^{2,3} • 24-V-DC-Niederspannung² • Drahtbruch (nur Strommodus) ⁴

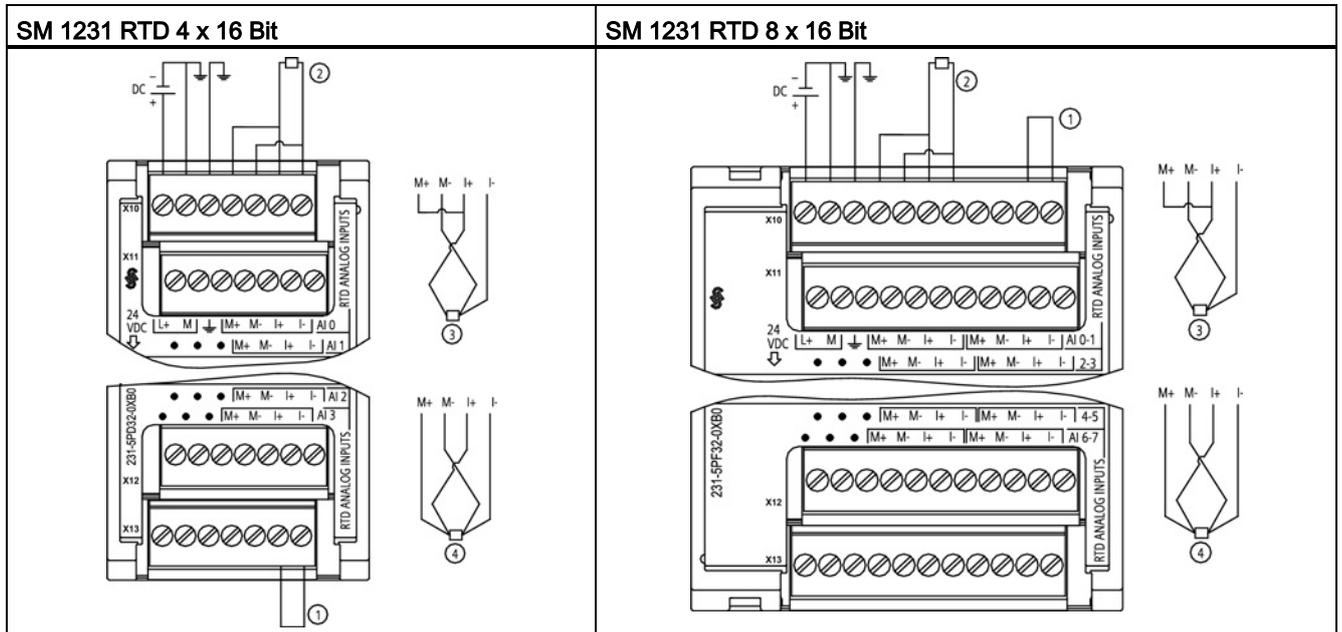
¹ 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus CPU)

² Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

³ Für Widerstandsmessbereiche ist die Prüfung auf Unterlauf grundsätzlich nicht aktiviert.

⁴ Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Tabelle A- 50 Schaltpläne der RTD-SMs



- ① Nicht belegte RTD-Eingänge zurückschleifen
- ② 2Draht-RTD
- ③ 3Draht-RTD
- ④ 4Draht-RTD

Hinweis: Die Steckverbinder müssen vergoldet sein. In Anhang C des Systemhandbuchs S7-1200 Automatisierungssystem finden Sie die Bestellnummer.

A.8.3 Technische Daten SM 1231 TC

Tabelle A- 51 Allgemeine technische Daten

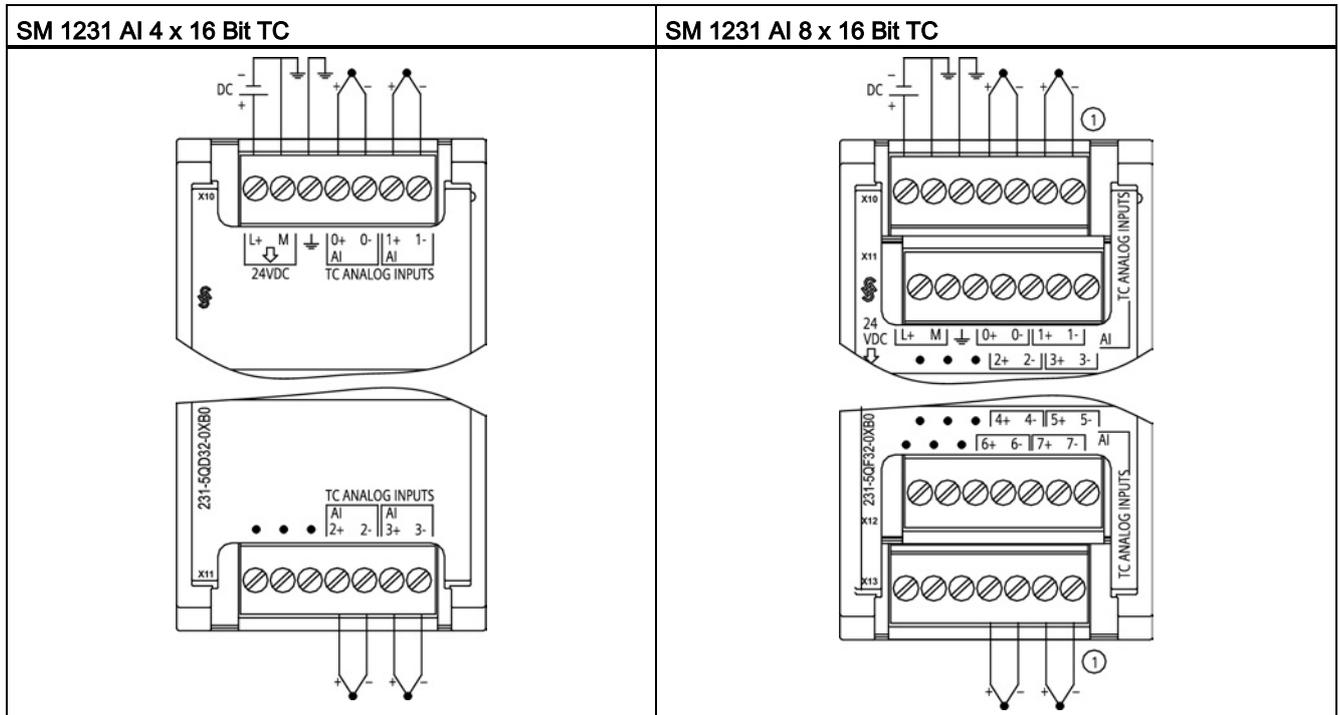
Modell	SM 1231 AI 4 x 16 Bit TC	SM 1231 AI 8 x 16 Bit TC
Bestellnummer	6ES7 231-5QD32-0XB0	6ES7 231-5QF32-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	xxx Gramm
Leistungsverlust	1,5 W	1,5 W
Stromaufnahme (SM-Bus)	80 mA	80 mA
Stromaufnahme ¹ (24 V DC)	40 mA	40 mA
Anzahl der Eingänge (Seite 353)	4	8
Typ	Potentialfrei, TC und mV	Potentialfrei, TC und mV
Diagnose	<ul style="list-style-type: none"> • Überlauf/Unterlauf ² • 24-V-DC-Niederspannung² • Drahtbruch (nur Strommodus) ³ 	<ul style="list-style-type: none"> • Überlauf/Unterlauf ² • 24-V-DC-Niederspannung² • Drahtbruch (nur Strommodus) ³

¹ 20,4 bis 28,8 V DC (Klasse 2, leistungsbegrenzt oder Geberspannung aus CPU)

² Die Informationen der Diagnosealarme "Überlauf", "Unterlauf" und "Niederspannung" werden auch dann mit den Analogdatenwerten gemeldet, wenn die entsprechenden Alarme bei der Modulprojektierung deaktiviert werden.

³ Wenn der Drahtbruchalarm deaktiviert ist und in der Geberverdrahtung ein frei liegender Draht vorliegt, kann das Modul zufällige Werte melden.

Tabelle A- 52 Schaltpläne der TC-SMs



① SM 1231 AI 8 TC: Die Anschlüsse von TC 2, 3, 4 und 5 werden zur besseren Übersichtlichkeit weggelassen.

A.8.4 Technische Daten der analogen Eingänge für RTD und TC (SM und SB)

Tabelle A- 53 Analoge Eingänge der RTD- und TC-Module (SB und SM)

Technische Daten	RTD und Thermoelement (TC)
Anzahl der Eingänge	1 (SB), 4 oder 8 (SM)
Typ	<ul style="list-style-type: none"> • RTD: Modulreferenz RTD und Ω • TC: Potentialfrei, TC und mV
Bereich	Siehe RTD/TC-Typtabellen:
<ul style="list-style-type: none"> • Nennbereich (Datenwort) • Überschwing-/Unterschwingbereich (Datenwort) • Überlauf/Unterlauf (Datenwort) 	<ul style="list-style-type: none"> • RTD (Seite 356) • TC (Seite 354)
Auflösung	0,1 °C / 0,1 °F
Temperaturbereich	15 Bit plus Vorzeichen
Widerstand/Spannung	±35 V
Max. Stehspannung	85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)
Rauschunterdrückung	> 120 dB bei 120 V AC
Gleichtaktunterdrückung	≥ 10 M Ω
Impedanz	

Technische Daten		RTD und Thermoelement (TC)
Elektrische Trennung	Feldseite zu Logik	500 V AC
	Feld zu 24 V DC	RTD- und TC-SMs: 500 V AC (gilt nicht bei RTD- und TC-SBs)
	24 V DC zu Logik	RTD- und TC-SMs: 500 V AC (gilt nicht bei RTD- und TC-SBs)
Trennung Kanäle untereinander		<ul style="list-style-type: none"> • RTD-SM: Ohne (gilt nicht bei RTD-SBs) • TC-SM: 120 V AC (gilt nicht bei TC-SBs)
Genauigkeit (25 °C/-20 bis 60 °C)		Siehe RTD/TC-Typtabellen: <ul style="list-style-type: none"> • RTD (Seite 356) • TC (Seite 354)
Wiederholgenauigkeit		±0,05 % Vollausschlag
Maximale Verlustleistung Geber		<ul style="list-style-type: none"> • RTD: 0,5 mW • TC: -/-
Messprinzip		Integrierend
Aktualisierungszeit Modul		Siehe RTD/TC-Filterauswahltabellen: <ul style="list-style-type: none"> • RTD (Seite 357) • TC (Seite 355)
Fehler kalte Verbindungsstelle		<ul style="list-style-type: none"> • RTD: -/- • TC: ±1,5 °C
Leitungslänge (Meter)		Max. 100 m zum Geber
Leitungswiderstand		<ul style="list-style-type: none"> • RTD: 20 Ω, 2,7 Ω für max. 10 Ω RTD • TC: max. 100 Ω

A.8.5 Thermoelementtyp

Tabelle A- 54 Thermoelementtyp (Bereiche und Genauigkeit)

Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Normalbereich ³ , ⁴ Genauigkeit bei 25 °C	Normalbereich ^{3,4} 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
K	-270,0 °C	-200,0 °C	1.372,0 °C	1.622,0 °C	±0,4 °C	±1,0 °C
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	±0,5 °C	±1,0 °C
E	-270,0 °C	-200,0 °C	1.000,0 °C	1.200,0 °C	±0,3 °C	±0,6 °C
R & S	-50,0 °C	100,0 °C	1.768,0 °C	2.019,0 °C	±1,0 °C	±2,5 °C
B	0,0 °C	200,0 °C	800,0 °C	--	±2,0 °C	±2,5 °C
	--	800,0 °C	1.820,0 °C	1.820,0 °C	±1,0 °C	±2,3 °C
N	-270,0 °C	-200,0 °C	1.300,0 °C	1.550,0 °C	±1,0 °C	±1,6 °C

Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Normalbereich ³ , ⁴ Genauigkeit bei 25 °C	Normalbereich ^{3,4} Genauigkeit -20 °C bis 60 °C
C	0,0 °C	100,0 °C	2.315,0 °C	2.500,0 °C	±0,7 °C	±2,7 °C
TXK/XK (L)	-200,0 °C	-150,0 °C	800,0 °C	1.050,0 °C	±0,6 °C	±1,2 °C
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 Minimums 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 die hier genannten Spezifikationen erfüllt werden.
- ⁴ Nur beim 4-kanaligen TC-SM: Bei Vorhandensein von abgestrahlter Funkfrequenz zwischen 970 MHz und 990 MHz kann sich die Genauigkeit verschlechtern.

Hinweis

Thermoelementkanal

Für jeden Kanal am Thermoelement-Signalmodul kann ein unterschiedlicher Thermoelementtyp konfiguriert werden (bei der Modulkonfiguration in der Software wählbar).

A.8.6 Filterauswahl und Aktualisierungszeiten beim Thermoelement

Für die Messung von Thermoelementen wird eine Integrationszeit von 100 ms empfohlen. Niedriger eingestellte Integrationszeiten führen zu einem höheren Wiederholgenauigkeitsfehler der Temperaturmessungen.

Tabelle A- 55 Filterauswahl und Aktualisierungszeiten beim Thermoelement

Unterdrückungsfrequenz (Hz)	Integrationszeit (ms)	Aktualisierungszeit (Sekunden)		
		1-kanaliges SB	4-kanaliges SM	8-kanaliges SM
10	100	0.301	1.225	2.450
50	20	0.061	0.263	0.525
60	16.67	0.051	0.223	0.445
400 ¹	10	0.031	0.143	0.285

- ¹ Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl der 400-Hz-Unterdrückung aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

A.8.7 Auswahltabelle RTD-Gebertyp

Tabelle A- 56 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 klimatisch	-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 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 +32767 ausgegeben.

Tabelle A- 57 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.

A.8.8 Filterauswahl und Aktualisierungszeiten beim RTD

Tabelle A- 58 Filterauswahl und Aktualisierungszeiten

Rauschunterdrückungsfrequenz (Hz)	Integrationszeit (ms)	Aktualisierungszeit (Sekunden)		
		1-kanaliges SB	4-kanaliges SM	8-kanaliges SM
10	100	4-/2-Draht: 0.301 3-Draht: 0.601	4-/2-Draht: 1.222 3-Draht: 2.445	4-/2-Draht: 2.445 3-Draht: 4.845
50	20	4-/2-Draht: 0.061 3-Draht: 0.121	4-/2-Draht: 0.262 3-Draht: .505	4-/2-Draht: 0.525 3-Draht: 1.015
60	16.67	4-/2-Draht: 0.051 3-Draht: 0.101	4-/2-Draht: 0.222 3-Draht: 0.424	4-/2-Draht: 0.445 3-Draht: 0.845
400 ¹	10	4-/2-Draht: 0.031 3-Draht: 0.061	4-/2-Draht: 0.142 3-Draht: 0.264	4-/2-Draht: 0.285 3-Draht: 0.525

¹ Um die Auflösung und Messgenauigkeit des Moduls bei Auswahl des 400-Hz-Filters aufrecht zu erhalten, beträgt die Integrationszeit 10 ms. Mit dieser Auswahl erfolgt auch die Rauschunterdrückung bei 100 Hz und 200 Hz.

Hinweis

Für alle aktivierten Kanäle ohne angeschlossenen Geber meldet das Modul 32767. Wenn außerdem die Prüfung auf offene Leitungen aktiviert ist, blinken am Modul die entsprechenden roten LEDs.

Optimale Genauigkeit für die 10 Ω-RTD-Bereiche ermöglichen 4-Draht-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Draht-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

A.9 Kommunikationsschnittstellen

Eine vollständige Liste der Module für die S7-1200 finden Sie im S7-1200 Systemhandbuch oder auf der Kundensupport-Website (<http://www.siemens.com/automation/>).

A.9.1 PROFIBUS-Master/Slave

A.9.1.1 CM 1242-5 PROFIBUS-Slave

Tabelle A- 59 Technische Daten des CM 1242-5

Technische Daten	
Bestellnummer	6GK7 242-5DX30-0XE0
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 die Bus-Terminierung) *)
Zulässige Umgebungsbedingungen	
Umgebungstemperatur	
<ul style="list-style-type: none"> • während Lagerung • während Transport • während Betriebsphase bei senkrechter Installation (Hutschiene horizontal) • während Betriebsphase bei waagerechter Installation (Hutschiene vertikal) 	<ul style="list-style-type: none"> • -40 °C ... 70 °C • -40 °C ... 70 °C • 0 °C ... 55 °C • 0 °C ... 45 °C
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme, Verlustleistung	
Art der Spannungsversorgung	DC
Spannungsversorgung aus dem Rückwandbus	5 V
Aufgenommener Strom (typisch)	150 mA
Verlustwirkleistung (typisch)	0,75 W
Maße und Gewichte	
<ul style="list-style-type: none"> • Breite • Höhe • Tiefe 	<ul style="list-style-type: none"> • 30 mm • 100 mm • 75 mm

Technische Daten

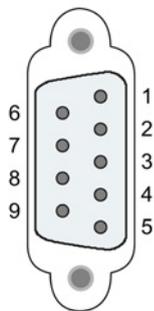
Gewicht

- | | |
|--------------------------------|---------|
| • Nettogewicht | • 115 g |
| • Gewicht inklusive Verpackung | • 152 g |

*) Die Strombelastung durch einen externen Verbraucher, der zwischen VP (Pin 6) und DGND (Pin 5) angeschlossen wird, darf für die Bus-Terminierung maximal 15 mA betragen (kurzschlussfest).

PROFIBUS-Schnittstelle

Tabelle A- 60 Kontaktbelegung 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: Datenader B	8	RxD/TxD-N: Datenader A
4	RTS	9	- nicht belegt -
5	M5V2: Datenbezugspotenzial (Masse DGND)	Gehäuse	Erdungsanschluss

A.9.1.2 CM 1243-5 PROFIBUS-Master

Tabelle A- 61 Technische Daten des CM 1243-5

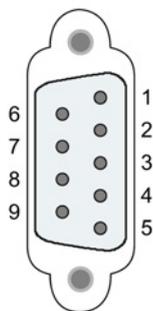
Technische Daten	
Bestellnummer	6GK7 243-5DX30-0XE0
Schnittstellen	
Anschluss an PROFIBUS	9-polige Sub-D-Buchse
Maximale Stromaufnahme an der PROFIBUS-Schnittstelle beim Anschluss von Netzkomponenten (beispielsweise optische Netzkomponenten)	15 mA bei 5 V (nur für die Bus-Terminierung) *)
Zulässige Umgebungsbedingungen	
Umgebungstemperatur	
• während Lagerung	• -40 °C ... 70 °C
• während Transport	• -40 °C ... 70 °C
• während Betriebsphase bei senkrechter Installation (Hutschiene horizontal)	• 0 °C ... 55 °C
• während Betriebsphase bei waagerechter Installation (Hutschiene vertikal)	• 0 °C ... 45 °C
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme, Verlustleistung	
Art der Spannungsversorgung	DC

Technische Daten	
Versorgungsspannung / extern	24 V
<ul style="list-style-type: none"> minimal maximal 	<ul style="list-style-type: none"> 19,2 V 28,8 V
Aufgenommener Strom (typisch)	
<ul style="list-style-type: none"> aus DC 24 V aus dem S7-1200-Rückwandbus 	<ul style="list-style-type: none"> 100 mA 0 mA
Verlustwirkleistung (typisch)	
<ul style="list-style-type: none"> aus DC 24 V aus dem S7-1200-Rückwandbus 	<ul style="list-style-type: none"> 2,4 W 0 W
Spannungsversorgung DC 24 V / extern	
<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.)
Maße 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 inklusive 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 die Bus-Terminierung maximal 15 mA betragen (kurzschlussfest).

PROFIBUS-Schnittstelle

Tabelle A- 62 Kontaktbelegung der Sub-D-Buchse



Pin	Beschreibung	Pin	Beschreibung
1	- nicht belegt -	6	VP: Spannungsversorgung +5V nur für Busabschlusswiderstände (Terminierung); nicht für Versorgung externer Geräte
2	- nicht belegt -	7	- nicht belegt -
3	RxD/TxD-P: Datenader B	8	RxD/TxD-N: Datenader A
4	CNTR-P: RTS	9	- nicht belegt -
5	DGND: Masse für Datensignale und VP	Gehäuse	Erdungsanschluss

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 Funktionserdung.

A.9.2 GPRS CP

Hinweis

Der GPRS CP hat keine Zulassung für Anwendungen für das Seewesen

Das folgende Modul hat keine Zulassung für das Seewesen:

- CP 1242-7 GPRS-Modul

Hinweis

Um diese Module nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

A.9.2.1 Technische Daten des CP 1242-7 GPRS

Tabelle A- 63 Technische Daten des CP 1242-7

Technische Daten	
Bestellnummer	6GK7 242-7KX30-0XE0
Funk-Schnittstelle	
Antennenanschluss	SMA-Buchse
Impedanz nominal	50 Ohm
Funkverbindung	
Maximale Sendeleistung	<ul style="list-style-type: none">• GSM 850, Class 4: +33 dBm \pm2dBm• GSM 900, Class 4: +33 dBm \pm2dBm• GSM 1800, Class 1: +30 dBm \pm2dBm• GSM 1900, Class 1: +30 dBm \pm2dBm
GPRS	Multislot-Klasse 10 Endgerätekategorie B Kodierungsschema 1...4 (GMSK)
SMS	Betriebsmodus abgehend: MO Dienst: Punkt zu Punkt
Zulässige Umgebungsbedingungen	

Technische Daten	
Umgebungstemperatur	
• während Lagerung	• -40 °C ... 70 °C
• während Transport	• -40 °C ... 70 °C
• während Betriebsphase bei senkrechter Installation (Hutschiene horizontal)	• 0 °C ... 55 °C
• während Betriebsphase bei waagerechter Installation (Hutschiene vertikal)	• 0 °C ... 45 °C
Relative Luftfeuchte bei 25 °C während der Betriebsphase, ohne Kondensation, maximal	95 %
Schutzart	IP20
Spannungsversorgung, Stromaufnahme, Verlustleistung	
Art der Spannungsversorgung	DC
Versorgungsspannung / extern	24 V
• minimal	• 19,2 V
• maximal	• 28,8 V
Aufgenommener Strom (typisch)	
• aus DC 24 V	• 100 mA
• aus dem S7-1200-Rückwandbus	• 0 mA
Verlustwirkleistung (typisch)	
• aus DC 24 V	• 2,4 W
• aus dem S7-1200-Rückwandbus	• 0 W
Spannungsversorgung DC 24 V	
• Min. Leitungsquerschnitt	• min.: 0,14 mm ² (AWG 25)
• Max. Leitungsquerschnitt	• max.: 1,5 mm ² (AWG 15)
• Anzugsmoment der Schraubklemmen	• 0,45 Nm (4 lb.in.)
Potenzialtrennung Netzteil gegen interne Schaltung	710 V DC für 1 Minute
Maße und Gewichte	
• Breite	• 30 mm
• Höhe	• 100 mm
• Tiefe	• 75 mm
Gewicht	
• Nettogewicht	• 133 g
• Gewicht inklusive Verpackung	• 170 g

Technische Daten der GSM/GPRS-Antenne ANT794-4MR

ANT794-4MR	
Bestellnummer	6NH9860-1AA00
Mobilfunknetze	GSM / GPRS
Frequenzbereiche	<ul style="list-style-type: none">• 824...960 MHz (GSM 850, 900)• 1 710...1 880 MHz (GSM 1 800)• 1 900...2 200 MHz (GSM / UMTS)
Charakteristik	omnidirektional
Antennengewinn	0 dB
Impedanz	50 Ohm
Stehwellenverhältnis (SWR)	< 2,0
Max. Leistung	20 W
Polarität	linear vertikal
Stecker	SMA
Länge Antennenkabel	5 m
Außenmaterial	Hart-PVC, UV-beständig
Schutzart	IP20
Zulässige Umgebungsbedingungen	<ul style="list-style-type: none">• Betriebstemperatur • -40 °C bis +70 °C• Transport-/Lagertemperatur • -40 °C bis +70 °C• Relative Feuchte • 100 %
Außenmaterial	Hart-PVC, UV-beständig
Konstruktiver Aufbau	Antenne mit 5 m fest verbundenem Kabel und SMA-Stecker
Maße (D x H) in mm	25 x 193
Gewicht	<ul style="list-style-type: none">• Antenne inkl. Kabel • 310 g• Montageteile • 54 g
Montage	Über mitgelieferten Winkel

Technische Daten der Flachantenne ANT794-3M

Bestellnummer	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	IP 64	
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.9.3 Teleservice (TS)

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.

A.9.4 RS485-, RS232- und RS422-Kommunikation

A.9.4.1 Technische Daten des CM 1241 RS485

Hinweis

Um dieses CB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 64 Allgemeine technische Daten

Technische Daten	CB 1241 RS485
Bestellnummer	6ES7 241-1CH30-1XB0
Abmessungen B x H x T (mm)	38 x 62 x 21
Gewicht	40 Gramm

Tabelle A- 65 Sender und Empfänger

Technische Daten	CB 1241 RS485
Typ	RS485 (2-Leiter-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	Kurzschluss Pin TB mit Pin T/RB, effektive Abschlussimpedanz beträgt 127Ω , Anschluss an RS485 Pin 3 Kurzschluss Pin TA mit Pin T/RA, effektive Abschlussimpedanz beträgt 127Ω , Anschluss an RS485 Pin 4
Eingangsimpedanz Empfänger	min. $5,4 \text{ k}\Omega$ einschließlich Abschluss
Ansprechgrenze/Sensibilität Empfänger	min. +/- 0,2 V, 60 mV typ. Hysterese
Potentialtrennung RS485-Signal zu Masse RS485-Signal zu CPU-Logik	500 V AC für 1 Minute
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- 66 Stromversorgung

Technische Daten	CB 1241 RS485
Verlustleistung	1,5 W
Max. Stromaufnahme (SM-Bus)	50 mA
Max. Stromaufnahme (24 V DC)	80 mA

<p>CB 1241 RS485 (6ES7 241-1CH30-1XB0)</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. Siehe Abschnitt "Abschließen eines RS485-Busanschlusssteckers" im S7-1200 Systemhandbuch.

Tabelle A- 67 Anschlussbelegungen für das CB 1241 RS485 (6ES7 241-1CH30-1XB0)

Pin	9-poliger Steckverbinder	X20
1	RS485 / logische Erde	--
2	RS485 / frei	--
3	RS485 / TxD+	3 - T/RB
4	RS485 / RTS	1 - RTS
5	RS485 / logische Erde	--
6	RS485 / 5-V-Leistung	--
7	RS485 / frei	--
8	RS485 / TxD-	4 - T/RA

Pin	9-poliger Steckverbinder	X20
9	RS485 / frei	--
Hülse		7 - M

A.9.4.2 Technische Daten des CM 1241 RS422/485

Technische Daten des CM 1241 RS422/485

Tabelle A- 68 Allgemeine technische Daten

Technische Daten	CM 1241 RS422/485
Bestellnummer	6ES7 241-1CH32-0XB0
Abmessungen B x H x T (mm)	30 x 100 x 75
Gewicht	155 Gramm

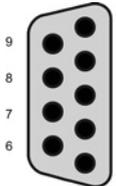
Tabelle A- 69 Sender und Empfänger

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-Netzwerksteckers und Konfigurieren von RS422 und RS485 im S7-1200 Systemhandbuch.
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	500 V AC für 1 Minute
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)
Anzahl Stoppbits	1 (Standard), 2
Flusskontrolle	XON/XOFF wird im RS422-Modus unterstützt
Wartezeit	0 bis 65535 ms

Tabelle A- 70 Stromversorgung

Technische Daten	CM 1241 RS422/485
Verlustleistung	1,1 W
aus +5 V DC	220 mA

Tabelle A- 71 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+	Signal B (Rx/D/TxD+): Eingang/Ausgang		8 TXD-	Signal A (Rx/D/TxD-): Eingang/Ausgang
4 RTS ²	Ausgang Sendeanforderung/RTS (TTL-Pegel)		9 TXD- ¹	Angeschlossen für RS422 Nicht belegt für RS485: Ausgang
5 GND	Logik- oder Kommunikationsmasse		SHELL	Erdungsanschluss

- ¹ Die Pins 2 und 9 werden lediglich als Sendesignale bei RS422 verwendet.
- ² RTS ist ein Signal auf TTL-Ebene und kann dazu genutzt werden, ein weiteres Halbduplex-Gerät basierend auf diesem Signal zu steuern. Es ist beim Senden aktiv und ansonsten inaktiv.

A.9.4.3 Technische Daten des CM 1241 RS232

Tabelle A- 72 Allgemeine technische Daten

Technische Daten	CM 1241 RS232
Bestellnummer	6ES7 241-1AH32-0XB0
Abmessungen (mm)	30 x 100 x 75
Gewicht	150 Gramm

Tabelle A- 73 Sender und Empfänger

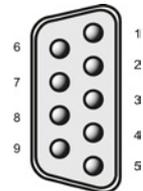
Technische Daten	CM 1241 RS232
Typ	RS232 (Vollduplex)
Ausgangsspannung Sender	min. +/- 5 V bei R _L = 3 kΩ
Ausgangsspannung Empfänger	max. +/- 15 V DC
Eingangsimpedanz Empfänger	min. 3 kΩ
Ansprechgrenze/Sensibilität Empfänger	min. 0,8 V Low, max. 2,4 V High, typ. Hysterese 0,5 V
Empfänger-Eingangsspannung	max. +/- 30 V DC

Technische Daten	CM 1241 RS232
Potentialtrennung RS-232-Signal zu Masse RS-232-Signal zu CPU-Logik	500 V AC für 1 Minute
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- 74 Stromversorgung

Technische Daten	CM 1241 RS232
Verlustleistung	1 W
aus +5 V DC	200 mA

Tabelle A- 75 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.10 Technologiemodule

A.10.1 SM 1278 4xIO-Link-Master SM

A.10.1.1 Technische Daten des Signalmoduls SM 1278 4xIO-Link-Master

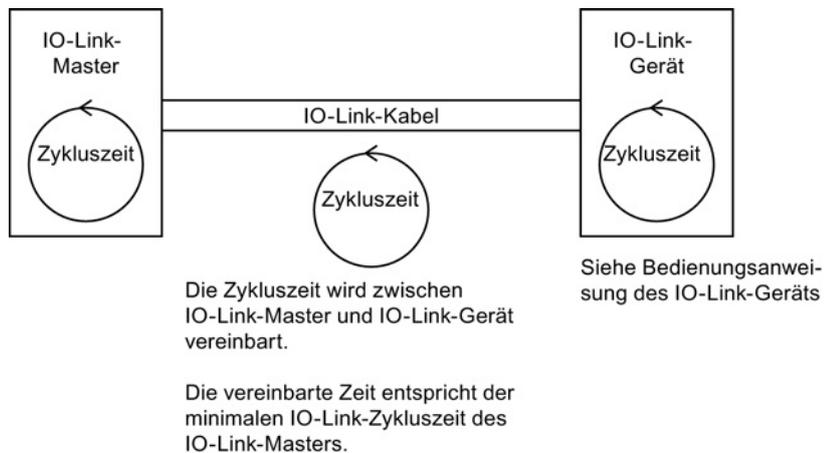
Tabelle A- 76 Allgemeine technische Daten

Technische Daten	Signalmodul SM 1278 4xIO-Link-Master
Bestellnummer	6ES7 278-4BD32-0XB0
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
Verlustleistung	
Verlustleistung, typ.	1 W, ohne Portlast
Digitaleingänge/-ausgänge	

Technische Daten		Signalmodul SM 1278 4xIO-Link-Master
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		
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 Byte
Größe der Prozessdaten, Ausgang je Modul		32 Byte
Speichergröße 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		

Technische Daten	Signalmodul SM 1278 4xIO-Link-Master
Potentialtrennung Kanäle	
Zwischen den Kanälen	Nein
Zwischen den Kanälen und dem Rückwandbus	Ja
Zulässige Potentialdifferenz	
Zwischen den verschiedenen Stromkreisen	75 V DC / 60 V AC (Basisisolierung)
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



A.10.1.2 SM 1278 4xIO-Link-Master SM, Schaltpläne

Tabelle A- 77 Schaltplan für das SM 1278 IO-Link-Master

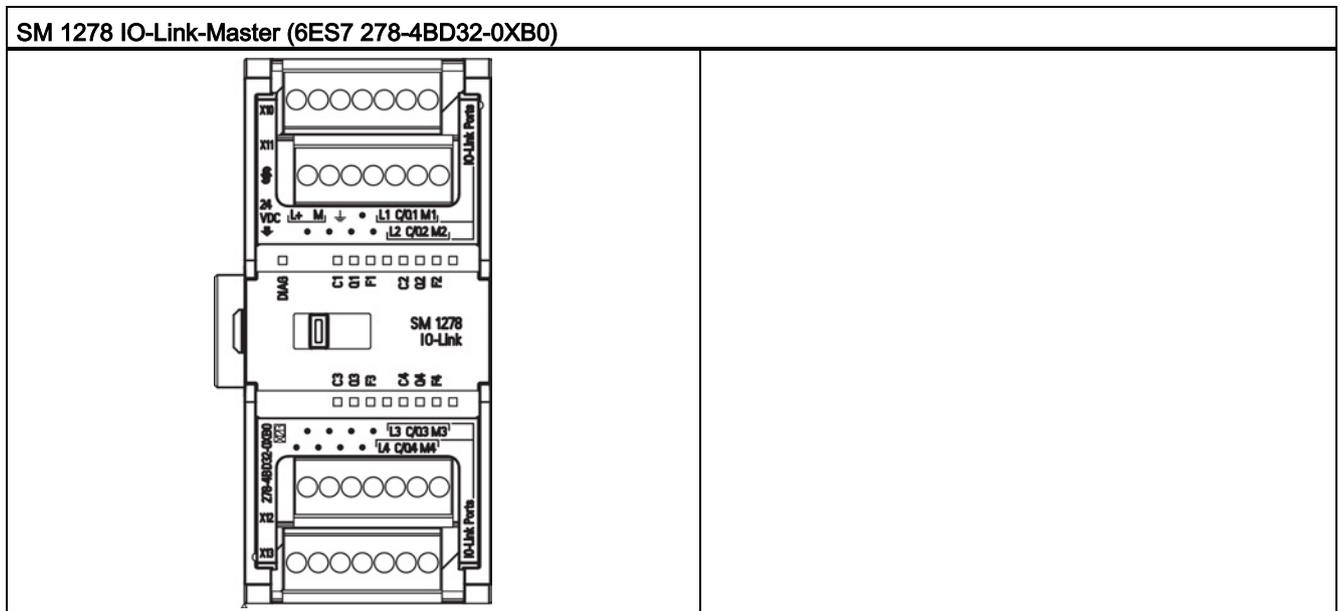


Tabelle A- 78 Anschlussbelegung für das SM 1278 IO-Link-Master (6ES7 278-4BD32-0XB0)

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/OL ₂	C/O ₃	C/OL ₄
7	ML ₁	ML ₂	M ₃	ML ₄

A.11 Zugehörige Produkte

A.11.1 PM 1207 Stromversorgungsmodul

Das PM 1207 ist ein Stromversorgungsmodul für die SIMATIC S7-1200. Es bietet die folgenden Leistungsmerkmale:

- 120/230-V-AC-Eingang, 24-V-DC/2,5-A-Ausgang
- Bestellnummer 6ESP 332-1SH71-4AA0

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Website mit dem Produktkatalog zum PM 1207.

Siehe auch

Kunden-Support (<http://www.siemens.com/automation/>)

A.11.2 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
- Bestellnummer 6GK7 277-1AA00-0AA0

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Website mit dem Produktkatalog zum CSM 1277.

Siehe auch

Kunden-Support (<http://www.siemens.com/automation/>)

A.11.3 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 (Bestellnummer 021620-B) und das für höhere Belastung ausgelegte CANopen (Ruggedized) -Modul (Bestellnummer 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.

Austauschen einer V3.0-CPU durch eine V4.0-CPU

B.1 Austauschen einer V3.0-CPU durch eine V4.0-CPU

Sie können Ihre V3.0-CPU durch eine V4.0-CPU ersetzen (Seite 82) und Ihr bestehendes, für die V3.0-CPU entwickeltes STEP 7-Projekt verwenden. Außerdem sollten Sie auf Firmware-Updates (Seite 298) für Ihre angeschlossenen Module prüfen und diese installieren.

Hinweis

Kein Gerätetausch möglich von V4.0 nach V3.0

Sie können eine V3.0-CPU durch eine V4.0-CPU ersetzen, doch Sie können nach dem Laden der Konfiguration eine V4.0-CPU nicht durch eine V3.0-CPU 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 jedoch können Sie den Austausch von V3.0 durch V4.0 nicht mehr rückgängig machen.

Beachten Sie dabei einige Konfigurations- und Betriebsänderungen zwischen den beiden CPU-Versionen:

Organisationsbausteine

Bei Version 4.0 können Sie konfigurieren, ob die OB-Ausführung unterbrechbar oder nicht unterbrechbar (Seite 63) sein soll. In Projekten von früheren V3.0-CPU legt STEP 7 für alle OBs standardmäßig die nicht unterbrechbare Ausführung fest.

STEP 7 legt für alle OB-Prioritäten (Seite 63) 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 beziehen sich auf das Submodul als Ganzes, wenn kein Diagnoseereignis ansteht.

CPU-Passwortschutz

STEP 7 legt für die V4.0-CPU als Passwortschutzstufe (Seite 91) die gleiche Stufe fest, die in der V3.0-CPU eingestellt war, und weist das Passwort der Version 3.0 dem Passwort für "Vollzugriff (kein Schutz)" der V4.0-CPU zu:

Schutzstufe in der Version 3.0	Zugriffsstufe in der Version V4.0
Kein Schutz	Vollzugriff (kein Schutz)
Schreibschutz	Lesezugriff
Schreib-/Lesechutz	HMI-Zugriff

Beachten Sie, dass die V4.0-Zugriffsstufe "Kein Zugriff (kompletter Schutz)" in der Version 3.0 nicht vorhanden war.

Webserver

Wenn Sie eine V3.0-CPU durch eine V4.0-CPU austauschen, sind Ihre Webserver-Projekteinstellungen zum Aktivieren des Webserver und ob HTTPS-Zugriff erforderlich ist die gleichen Einstellungen wie in der Version 3.0. Sie können dann Benutzer, Rechte, Passwörter (Seite 209) und Sprachen für die Nutzung des Webserver nach Bedarf konfigurieren. Wenn Sie keine Benutzer mit zusätzlichen Rechten konfigurieren, sind Sie auf das begrenzt, was auf den Standard-Webseiten (Seite 210) angezeigt wird. Die S7-1200-CPU V4.0 unterstützt den bisher vorkonfigurierten Benutzer "admin" und das zugehörige Passwort nicht.

Bisher bot die Webserverseite für das Datenprotokoll die Funktion "Herunterladen und Löschen". Die Dateibrowser-Seite (Seite 210) des Webserver der Version 4.0, über die Sie auf Datenprotokolle zugreifen, bietet diese Funktion nicht. Stattdessen bietet der Webserver die Möglichkeit zum Herunterladen, Umbenennen und Löschen von Datenprotokolldateien.

Inkompatibilität der Übertragungskarte

Sie können eine Übertragungskarte (Seite 67) der Version 3.0 nicht dazu verwenden, ein Programm der Version 3.0 in eine CPU der Version 4.0 zu übertragen. Sie müssen das V3.0-Projekt in STEP 7 öffnen, das Gerät in eine V4.0-CPU ändern (Seite 82) und das STEP 7-Projekt in Ihre V4.0-CPU laden. Nachdem Sie Ihr Projekt in ein Projekt der Version 4.0 geändert haben, können Sie eine V4.0-Übertragungskarte für nachfolgende Programmübertragungen einrichten.

GET/PUT-Kommunikation

In Version 3.0 war die GET/PUT-Kommunikation standardmäßig aktiviert. Wenn Sie Ihre V3.0-CPU durch eine V4.0-CPU ersetzen (Seite 82), wird im Informationsbereich zur Kompatibilität eine Meldung angezeigt, dass GET/PUT aktiviert ist.

Unterstützung der Bewegungssteuerung

S7-1200 V4.0-CPU's 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 (Seite 237) der Version 3.0 ersetzt.

Wenn Sie für ein STEP 7-Projekt mit zwei verschiedenen Versionen von Bewegungssteuerungsanweisungen (V3.0 und V4.0) einen Gerätetausch durchführen, werden durch den Gerätetausch zum Zeitpunkt des Übersetzens kompatible Bewegungssteuerungsanweisungen (Seite 237) der Version 4.0 ersetzt.

Bei einem Gerätetausch von V3.0 nach V4.0 wird die Version des Technologieobjekts (TO) für die Bewegungssteuerung automatisch zur Version 4.0. Die TO-Struktur unterscheidet sich zwischen V3.0 und V4.0. Alle zugehörigen Bausteine werden ebenfalls verändert. Bausteinschnittstellen, Beobachtungstabellen und die Trace-Funktion werden auf die neue Struktur von V4.0 aktualisiert. Die Unterschiede in den Achsenparametern der Bewegungssteuerung zwischen der Version 3.0 und der Version 4.0 werden in den folgenden zwei Tabellen aufgeführt:

V3.0	V4.0
Config.General.LengthUnit	Units.LengthUnit
Config.Mechanics.PulsesPerDriveRevolution	Actor.DriveParameter.PulsesPerDriveRevolution
Config.Mechanics.LeadScrew	Mechanics.LeadScrew
Config.Mechanics.InverseDirection	Actor.InverseDirection
Config.DynamicLimits.MinVelocity	DynamicLimits.MinVelocity
Config.DynamicLimits.MaxVelocity	DynamicLimits.MaxVelocity
Config.DynamicDefaults.Acceleration	DynamicDefaults.Acceleration
Config.DynamicDefaults.Deceleration	DynamicDefaults.Deceleration
Config.DynamicDefaults.EmergencyDeceleration	DynamicDefaults.EmergencyDeceleration
Config.DynamicDefaults.Jerk	DynamicDefaults.Jerk
Config.PositionLimits_SW.Active	PositionLimitsSW.Active
Config.PositionLimits_SW.MinPosition	PositionLimitsSW.MinPosition
Config.PositionLimits_SW.MaxPosition	PositionLimitsSW.MaxPosition
Config.PositionLimits_HW.Active	PositionLimitsHW.Active
Config.PositionLimits_HW.MinSwitchedLevel	PositionLimitsHW.MinSwitchLevel
Config.PositionLimits_HW.MaxSwitchedLevel	PositionLimitsHW.MaxSwitchLevel
Config.Homing.AutoReversal	Homing.AutoReversal
Config.Homing.Direction	Homing.ApproachDirection
Config.Homing.SideActiveHoming	Sensor[1].ActiveHoming.SideInput
Config.Homing.SidePassiveHoming	Sensor[1].PassiveHoming.SideInput
Config.Homing.Offset	Sensor[1].ActiveHoming.HomePositionOffset
Config.Homing.FastVelocity	Homing.ApproachVelocity
Config.Homing.SlowVelocity	Homing.ReferencingVelocity
MotionStatus.Position	Position
MotionStatus.Velocity	Velocity

V3.0	V4.0
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	V4.0
Config.Command[]	Command[]

Hinweis: Das Array "Command[]" ist in Version 3.0 ein UDT vom Typ "TO_CmdTab_Config_Command" und in Version 4.0 ein UDT vom Typ "TO_Struct_Command".

Geänderte Anweisungen

Bei den folgenden Anweisungen ändern sich Parameter oder das Verhalten:

- RDREC und WRREC (Seite 147)
- CONV (Seite 114)

HMI-Panel-Kommunikation

Wenn Sie eines oder mehrere HMI-Panels (Seite 23) an Ihre S7-1200 CPU der Version 3.0 angeschlossen hatten, hängt die Kommunikation mit der S7-1200 CPU der Version 4.0 vom verwendeten Kommunikationstyp und 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 Austauschen einer V3.0-CPU durch eine V4.0-CPU müssen Sie alle Programmbausteine neu übersetzen, damit Sie sie in die CPU der Version 4.0 laden können. Außerdem müssen Sie bei Bausteinen mit Knowhow-Schutz (Seite 94) oder an eine PLC-Seriennummer gebundenem Kopierschutz (Seite 95) den Schutz aufheben, damit Sie die Bausteine übersetzen und laden können. (An eine Memory Card gebundenen Kopierschutz brauchen Sie jedoch nicht aufzuheben.) Nach dem Laden 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 4.0 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.

Ausführliche Informationen zu den V4.0-Funktionen finden Sie im S7-1200 Systemhandbuch.

Siehe auch

Ändern eines Geräts (Seite 82)

Austauschen einer V3.0-CPU durch eine V4.0-CPU

B.1 Austauschen einer V3.0-CPU durch eine V4.0-CPU

Index

"

"Box"-Anweisung
Erste Schritte, 51

A

Abstand, Luftströmung und Kühlung, 27

Achssteuerungsbefehle als Bewegungsfolge ausführen
(MC_CommandTable), 257

Ad-hoc-Modus, TCP und ISO-on-TCP, 150

Adressierung

Boolesche Werte oder Bitwerte, 72

Datenbaustein, 71

Einzelne Eingänge (E) oder Ausgänge (A), 72

Globaler Speicher, 71

Prozessabbild, 71

Speicherbereiche, 71

Temporärer Speicher, 71

Aktive/passive Kommunikation

Parameter, 153

Partner konfigurieren, 156, 171

Verbindungs-IDs, 150

Alarmer

Alarmlatenz, 63

Übersicht, 63

Analoge E/A

Darstellung Ausgang (Spannung), 346

Darstellung Ausgang (Strom), 346

Darstellung Eingang (Spannung), 342

Darstellung Eingang (Strom), 342

Schrittantwortzeiten der Eingänge, 343

Umwandlung in physikalische Einheiten, 53

Analoge Eingänge

Technische Daten für CPUs, SBs und SMs, 341

Technische Daten für RTD/TC-SBs und -SMs, 353

Analoge Signalboards

SB 1231, 336

SB 1232, 336

Analoge Signalmodule

SM 1231, 337

SM 1231 RTD, 350

SM 1231 Thermoelement, 352

SM 1232, 338

SM 1234, 338

Analogwerte skalieren, 53

Ändern

Zustand im Programmiereditor, 288

Anfangswerte

Startwerte eines DBs erfassen und
zurücksetzen, 293

Anlaufparameter, 84

Anlegen einer HMI-Verbindung, 55

Anschlüsse

Anzahl der Verbindungen (PROFINET), 147

Ethernet-Protokolle, 170

HMI-Verbindung, 55

Kommunikationsarten, 145

Konfiguration, 153

Netzwerkverbindung, 55

Partner, 156

S7-Verbindung, 170

Typen, Verbindungen mit mehreren

Teilnehmern, 170

Verbindungs-IDs, 150

Webserver, 213

Anweisung Bereich kopieren (MOVE_BLK), 113

Anweisung Bereich ununterbrechbar kopieren
(UMOVE_BLK), 113

Anweisung CTRL_PWM, 125

Anweisung TRCV_C, 147

Anweisung TSEND_C, 147

Anweisung Verschieben, 113

Anweisungen

Analogwerte skalieren, 53

Beobachten, 287, 288

Bereich kopieren (MOVE_BLK), 113

Bereich ununterbrechbar kopieren

(UMOVE_BLK), 113

Bewegungssteuerung, 237

Bitverknüpfung, 110

CALCULATE, 52

CALCULATE (Berechnen), 116

CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl
erzeugen), 115

CONV (Umwandeln), 114

CTRL_PWM, 125

DeviceStates, 132

Drag & Drop, 34

Drag & Drop zwischen Editoren, 40

Einfügen, 34

Ergänzen von Eingängen oder Ausgängen in KOP-
und FUP-Anweisungen, 35

Erste Schritte, 50, 51

- Erweiterbare Anweisungen, 36
- Favoriten, 35
- FLOOR, 115
- Forcefunktion, 291
- Forcen, 290
- GET, 169
- GET_DIAG, 132
- Hinzufügen eines Parameters, 51
- HSC (schneller Zähler), 133, 134
- LED-Zustand, 131
- MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern), 260
- MC_CommandTable, 257
- MC_Halt (Achse pausieren), 245
- MC_Home (Referenzpunktfahrt der Achse durchführen), 242
- MC_MoveAbsolute (Achse absolut positionieren), 247
- MC_MoveJog (Achse im Tipbetrieb bewegen), 254
- MC_MoveRelative (Achse relativ positionieren), 249
- MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 251
- MC_Power (Achse freigeben/sperrern), 238
- MC_ReadParam (Parameter eines Technologieobjekts lesen), 264
- MC_Reset (Fehler bestätigen), 241
- MC_WriteParam (Parameter des Technologieobjekts schreiben), 262
- ModuleStates, 132
- NORM_X (Normalisieren), 116
- PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 189
- PUT, 169
- ROUND, 115
- SCALE_X (Skalieren), 116
- Spalten und Überschriften, 39
- TRCV_C, 147
- TRUNC (Ganzzahl erzeugen), 115
- TSEND_C, 147
- Vergleichen, 113
- Verschieben, 113
- Versionen von Anweisungen, 39
- WWW, 219
- Zähler, 122
- Zustand, 287, 288
- Anwenderprogramm
 - Bausteine aus einer Online-CPU kopieren, 294
 - Drag & Drop zwischen Editoren, 40
 - Einfügen von Anweisungen, 34
 - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 35
 - Erweiterbare Anweisungen, 36
 - Favoriten, 35
 - Mit CPU oder Memory Card verknüpfen, 95
 - Organisationsbausteine (OBs), 99
 - Passwortschutz, 94
- Anzeige erster Zyklus, 87
- Arbeitsspeicher, 16, 67
- Arithmetik, 52, 116
- ASi
 - Adresse,
 - AS-i-Master CM 1243-2,
 - AS-i-Mastermodul CM1243-2 hinzufügen,
 - AS-i-Slave hinzufügen,
- ATEX-Zulassung, 312
- Auf Werkseinstellungen zurücksetzen, 297
- Aufrufstruktur, 130
- Aus CPU laden
 - Erkennen, 299
- Ausgangsparameter, 101
- B**
- Batterieboard, BB 1297, 340
- Bausteinanruf
 - Grundwissen, 62
- Bausteine
 - Alarmer, 17, 63
 - Anlauf-OBs, 63
 - Anzahl der Codebausteine, 17
 - Anzahl der OBs, 17, 63
 - Arten, 62
 - Aufrufen eines anderen Codebausteins, 105
 - Bausteine aus einer Online-CPU kopieren, 294
 - Beobachten, 17
 - Datenbausteine (DBs), 62
 - Ereignisse, 63
 - Erste Schritte, 103
 - Funktionen (FCs), 62
 - Funktionsbausteine (FBs), 62
 - Größe des Anwenderprogramms, 17
 - Konsistenzprüfung, 130
 - Organisationsbausteine (OBs), 17, 62, 63
 - Passwortschutz, 94
 - Schachtelungstiefe, 17
 - Zähler (Anzahl und Speicherbedarf), 18
 - Zeiten (Anzahl und Speicherbedarf), 17
- Bausteine aus einer Online-CPU kopieren, 294
- Bearbeiten im Betriebszustand RUN, (Laden im Betriebszustand RUN)
- Bedienoberfläche
 - STEP 7 Projekt- und Portalansicht, 33
- Bedienpanel, 23, 36, 61, 286
- Bemessungsspannungen, 316, 316

- Benutzerdefinierte Webseiten, 209, 216
 - Aktivieren mit WWW-Anweisung, 219
 - Aktualisieren, 217
 - Anlegen mit HTML-Editor, 216
 - Einschränkungen Ladespeicher, 217
 - Konfigurieren, 218
 - Programmbausteine generieren, 218
 - Programmieren in STEP 7, 219
 - Benutzerdefinierte Webseiten aktualisieren, 217
 - Benutzerdefinierte Webseiten anlegen, 216
 - Beobachten
 - Beobachtungstabelle, 287
 - Forcefunktion, 291
 - Forcetabelle, 290
 - KOP-Zustand, 287, 288
 - Startwerte eines DBs zurücksetzen, 293
 - Werte eines DBs erfassen, 293
 - Beobachtungstabelle
 - Beobachten, 287
 - Forcen, 128
 - Beobachtungstabellen, 288
 - Betriebszustand, 36, 36, 61, 286
 - Betriebszustand RUN, 60, 62
 - Bedienpanel, 36, 61, 286
 - Forcefunktion, 291
 - Schaltflächen in der Funktionsleiste, 36
 - Betriebszustand STOP, 60
 - Bedienpanel, 61, 286
 - Forcefunktion, 291
 - Schaltflächen in der Funktionsleiste, 36
 - Bewegungsfolge (MC_CommandTable), 257
 - Bewegungssteuerung
 - Achse konfigurieren, 230
 - Hardware- und Software-Endschalter, 268
 - Konfigurationsparameter für die Referenzpunktfahrt, 273
 - MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern), 260
 - MC_CommandTable, 257
 - MC_Halt (Achse pausieren), 245
 - MC_Home (Referenzpunktfahrt der Achse durchführen), 242
 - MC_MoveAbsolute (Achse absolut positionieren), 247
 - MC_MoveJog (Achse im Tippbetrieb bewegen), 254
 - MC_MoveRelative (Achse relativ positionieren), 249
 - MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 251
 - MC_Power (Achse freigeben/sperrn), 238
 - MC_ReadParam (Parameter eines Technologieobjekts lesen), 264
 - MC_Reset (Fehler bestätigen), 241
 - MC_WriteParam (Parameter des Technologieobjekts schreiben), 262
 - Referenzpunktfahrt (Abfolge bei der aktiven Referenzpunktfahrt), 275
 - Referenzpunktfahrt der Achse, 271
 - Übersicht, 221
 - Bewegungssteuerungsanweisungen, 237
 - Bibliothek für das USS-Protokoll, 181
 - Bitverknüpfung, 110
 - Boolesche Werte oder Bitwerte, 72
 - Busstecker, 21
- ## C
- CALCULATE (Berechnen), 116
 - Analogwerte skalieren, 53
 - Für komplexe Gleichungen verwenden, 52
 - CANopen-Module
 - 021620-B, 021630-B, 375
 - CB 1241 RS485, 366
 - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 115
 - CE-Zulassung, 311
 - Codebaustein, 97
 - Alarmer, 17
 - Anzahl der Codebausteine, 17
 - Anzahl der OBs, 17
 - Baustein aufrufen, 105
 - DB (Datenbaustein), 102
 - FB (Funktionsbaustein), 101
 - FC (Funktion), 101
 - Größe des Anwenderprogramms, 17
 - Knowhow-Schutz, 94
 - Kopierschutz, 95
 - Mit CPU oder Memory Card verknüpfen, 95
 - Organisationsbausteine (OBs), 17, 99
 - Schachtelungstiefe, 17
 - Überwachung, 17
 - Zähler (Anzahl und Speicherbedarf), 18
 - Zeiten (Anzahl und Speicherbedarf), 17
 - Codebausteine vergleichen, 295
 - Compact Switch Module
 - CSM 1277, 374
 - CONV (Umwandeln), 114
 - Cookie-Einschränkungen, Standard-Webseiten, 215
 - CPU
 - Anlaufparameter, 84
 - Anlaufverarbeitung, 84
 - ASi-Adresse,
 - Auf Werkseinstellungen zurücksetzen, 297
 - Baustein aufrufen, 105
 - Bausteine aus einer Online-CPU kopieren, 294

Bausteine vergleichen und synchronisieren, 295
 Bedienpanel, 36, 61, 286
 Beobachten, 287
 Beobachtungstabellen, 288
 Betriebszustände, 60
 Diagnosepuffer, 296
 Ethernet-Port, 89
 Forcen, 290, 291
 Gerätekonfiguration, 79
 HSC-Konfiguration, 142
 IP-Adresse, 89
 Knowhow-Schutz, 94
 Kommunikation mit HMI konfigurieren, 143
 Kommunikationsarten, 145
 Konfigurieren von Parametern, 84, 88
 Netzwerkverbindung, 144
 Neues Gerät hinzufügen, 81
 Nicht spezifizierte CPU, 80, 299
 OBs verarbeiten, 99
 Online, 287, 296
 Online gehen, 285
 Passwortschutz, 91
 PROFIBUS, 162
 PROFIBUS-Adresse, 163
 PROFINET, 89
 Programmausführung, 59
 Schaltflächen RUN/STOP, 36
 Schrittwortzeiten der analogen Eingänge, 343
 Schutzstufen, 91
 Startwerte eines DBs zurücksetzen, 293
 Thermischer Bereich, 25, 27
 Werte eines DBs erfassen, 293
 Zugriffsschutz, 91
 CPU-Eigenschaften, benutzerdefinierte
 Webseiten, 218
 CPUs
 Anzahl der Kommunikationsverbindungen, 147
 CPU 1211C, 319
 CPU 1212C, 319
 CPU 1214C, 319
 CPU 1215C, 319
 CPU 1217C, 319
 Kommunikationsboards (CB), 21
 Module hinzufügen, 82
 Signalboards (SB), 21
 Thermischer Bereich, 24
 Übersicht, 15
 Vergleichstabelle, 16
 CSM 1277 Compact Switch Module, 374
 C-Tick-Zulassung, 313
 cULus-Zulassung, 312

D

Datenbaustein
 Globaler Datenbaustein, 71, 102
 Instanz-Datenbaustein, 71
 Organisationsbausteine (OBs), 99
 Startwerte zurücksetzen, 293
 Werte erfassen, 293
 Datenhantierungsbausteine (DHBs), 102
 Datenprotokoll
 Überblick über Datenprotokolle, 126
 Datentyp DTL, 69
 Datentyp DTL (Date and Time Long), 69
 Datentypen, 68
 DTL, 69
 DB (Datenbaustein), 102
 DB-Fragmente (benutzerdefinierte Webseiten)
 Generieren, 218
 DBs für benutzerdefinierte Webseiten erstellen, 218
 DBs für benutzerdefinierte Webseiten generieren, 218
 DC
 Ausgänge, 317
 DeviceStates, 132
 Diagnose
 Anweisung LED, 131
 DeviceStates, 132
 GET_DIAG, 132
 ModuleStates, 132
 Puffer, 296
 Statusanzeige, 87
 Digitale Ausgänge
 Technische Daten für Relais, 24-V-DC-CPU, SMs
 und SBs, 334
 Digitale Eingänge
 Technische Daten AC für SMs, 333
 Technische Daten V DC für CPUs, SMs und
 SBs, 331
 Digitale Signalboards
 SB 1221, 323
 SB 1222, 323
 SB 1223, 324
 Digitale Signalmodule
 SM 1221, 326
 SM 1222, 327
 SM 1223, 329, 330
 Dokumentation, 5
 Drag & Drop zwischen Editoren, 40

E

E/A
 Adressierung, 73

- Beobachtungstabellen, 288
 - Darstellung Analogausgang (Spannung), 346
 - Darstellung Analogausgang (Strom), 346
 - Darstellung Analogeingang (Spannung), 342
 - Darstellung Analogeingang (Strom), 342
 - Forcefunktion, 291
 - Schrittantwortzeiten der analogen Eingänge, 343
 - Zustand in KOP beobachten, 288
 - Editoren teilen
 - Erste Schritte, 46, 50
 - Ein- und Ausgänge
 - Beobachten, 287
 - Einbau
 - Abstand, 27
 - Kühlung, 27
 - Luftströmung, 27
 - Montageabmessungen, 24
 - Richtlinien, 27
 - Signalmodule (SM), 21
 - Thermischer Bereich, 24, 27
 - Übersicht, 27
 - Einfügen eines Geräts
 - Nicht spezifizierte CPU, 80, 299
 - Einfügen von Anweisungen
 - Drag & Drop, 34
 - Drag & Drop zwischen Editoren, 40
 - Favoriten, 35
 - Einschränkungen
 - Benutzerdefinierte Webseiten, 217
 - Webserver, 213
 - Einstellungen, 38
 - Einstellungen von STEP 7 ändern, 38
 - Elektromagnetische Verträglichkeit (EMV), 314, 315
 - Entwerfen einer Automatisierungslösung mit einem PLC-Gerät, 62, 97
 - Ereignisausführung, 63
 - Ereignisse, 296
 - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 35
 - Erkennen, 299
 - Erkennung zum Laden einer Online-CPU, 80
 - Ersetzen von Modulen, 43
 - Erste Schritte
 - "Box"-Anweisung, 51
 - Adressierung, 50
 - Anweisungen, 50
 - Arithmetische Anweisung, 51
 - Codebaustein, 103
 - CPU, 45
 - Editoren teilen, 46, 50
 - HMI, 54, 56
 - HMI-Verbindung, 55
 - Kontakte, 48
 - KOP-Programm, 48, 51
 - Netzwerk, 48
 - Netzwerkverbindung, 55
 - neue SPS, 45
 - PLC-Variablen, 46, 50
 - Programmbaustein, 103
 - Projekt, 45
 - Variablen, 46, 50
 - Erweiterbare Anweisungen, 36
 - Erweiterung der Leistungsmerkmale der S7-1200, 18
 - Ethernet
 - Ad-hoc-Modus, 150
 - Anzahl der Kommunikationsverbindungen, 147
 - CSM 1277 Compact Switch Module, 374
 - GET, 169
 - IP-Adresse, 89
 - Kommunikation, 143
 - Kommunikationsarten, 145
 - Netzwerkverbindung, 144
 - PUT, 169
 - Übersicht, 149
 - Verbindungs-IDs, 150
 - Ethernet-Anweisungen
 - TRCV_C, 147
 - TSEND_C, 147
 - Ethernet-Protokolle, 149
 - Verbindungen mit mehreren Teilnehmern, 170
- ## F
- FAQs, 5
 - Favoriten, Funktionsleiste, 35
 - FB (Funktionsbaustein), 101
 - FC (Funktion), 101
 - Firmware aktualisieren
 - aus STEP 7, 298
 - Firmware-Update
 - aus STEP 7, 298
 - FLOOR, 115
 - FM-Zertifizierung, 312
 - Forcen, 290, 291
 - Eingänge der Peripherie, 290, 291
 - Eingänge und Ausgänge, 291
 - Speicherbereich E, 290, 291
 - Zyklus, 291
 - Forcetabelle
 - Eingänge der Peripherie ansprechen, 290
 - Forcefunktion, 291
 - Forcen, 290
 - Frequenz, Taktbits, 87
 - Funktion (FC), 101

- Knowhow-Schutz, 94
- Funktionsbaustein (FB)
 - Anfangswert, 101
 - Ausgangsparameter, 101
 - Instanz-Datenbaustein, 101
 - Knowhow-Schutz, 94
- FUP (Funktionsplan), 107

G

- Gerät ändern, 82
- Geräteaustausch, 82
- Gerätekonfiguration, 79
 - Ändern eines Gerätetyps, 82
 - ASi,
 - AS-i-Port, 166
 - Erkennen, 80, 299
 - Ethernet-Port, 89
 - Konfigurieren der CPU, 84, 88
 - Konfigurieren der Module, 84, 88
 - Module hinzufügen, 82
 - Netzwerkverbindung, 144
 - Neues Gerät hinzufügen, 81
 - Nicht gesteckte Module, 43
 - PROFIBUS, 162
 - PROFINET, 89
- GET, 169
 - Verbindung konfigurieren, 157
- GET_DIAG, 132
- Globale Bibliothek
 - USS, 181
- Globaler Datenbaustein, 71, 102
- Globaler Speicher, 71
- Grundlegende Panels (HMI), 23

H

- Handbücher, 5
- Hardwarekonfiguration, 79
 - ASi,
 - AS-i-Port, 166
 - Erkennen, 80, 299
 - Ethernet-Port, 89
 - Konfigurieren der CPU, 84, 88
 - Konfigurieren der Module, 84, 88
 - Module hinzufügen, 82
 - Netzwerkverbindung, 144
 - Neues Gerät hinzufügen, 81
 - PROFIBUS, 162
 - PROFINET, 89
- Haupteinspeisung, 327

- HMI
 - Bild, 56
 - Erste Schritte, 54, 56
 - HMI-Verbindung, 55
 - Netzwerkverbindung, 55
 - PROFINET-Kommunikation konfigurieren, 143
- HMI-Geräte
 - Netzwerkverbindung, 144
 - Übersicht, 23
- HMI-Verbindung, 55
- HochspannungsIsolationsprüfung,
- Hotline, 6
- HSC (schneller Zähler)
 - Funktionsweise, 133, 134
 - Konfiguration, 142
- HSC-Eingangstaktfrequenzen, 333
- HTML-Seiten, benutzerdefiniert, 216
 - Aktualisieren, 217
 - Entwickeln, 216
 - Seitenanordnung, 218
- HTTP-Verbindungen, Webserver, 213

I

- Impulsfolge (PTO), 125
- Industrieumgebungen
 - Zulassungen, 313
- Informationsquellen, 5
- Instanz-Datenbaustein, 71
- IO-Link
 - Anlauf, 305
 - Daten, 306
 - Geräteprofil, 306
 - Komponenten, 305
 - Konfigurieren, 306
 - Technischer Überblick, 305
- IO-Link-Master
 - Abbildung, 307
 - Beispiele für Anschlüsse, 309
 - Schaltbild, 308
- IO-Link-Master-Signalmodul, 370
- IP-Adresse, 89
 - CPU online konfigurieren, 296
- IP-Adresse des Routers, 89
- IP-Router, 89
- ISO on TCP
 - Ad-hoc-Modus, 150
- ISO-on-TCP
 - Parameter, 153
 - Verbindungs-IDs, 150
 - Verbindungskonfiguration, 156
- ISO-on-TCP-Protokoll, 149

J

JavaScript-Einschränkungen, Standard-Webseiten, 214

K

Knowhow-Schutz

 Passwortschutz, 94

Kommunikation

 aktiv/passiv, 171

 Aktiv/passiv, 153, 156

 Anzahl der Verbindungen (PROFINET), 147

 ASi-Adresse,

 IP-Adresse, 89

 Konfiguration, 153, 156, 171

 Netzwerk, 143

 Netzwerkverbindung, 144

 Parameter, 153

 PROFIBUS-Adresse, 163

 PROFINET und PROFIBUS, 145

 TCON_Param, 153

 Verbindungs-IDs, 150

Kommunikationsboard (CB)

 CB 1241 RS485, 366

 LED-Anzeigen, 180

 Module hinzufügen, 82

 RS485, 180

 Übersicht, 21

 Vergleichstabelle, 18

Kommunikationsmodul

 Neues Gerät hinzufügen, 81

Kommunikationsmodul (CM)

 AS-i-Mastermodul CM1243-2 hinzufügen, 165

 CM 1241 RS232, 368

 CM 1241 RS422/RS485, 367

 CM 1243-5 (DP-Master) hinzufügen, 161

 LED-Anzeigen, 180

 Module hinzufügen, 82

 RS232 und RS485, 180

 Übersicht, 22

 Vergleichstabelle, 18

Kommunikationsmodule (CM)

 Gerätekonfiguration, 79

Kommunikationsmodule (CM), USS-Bibliothek, 181

Kommunikationsprozessor (CP)

 Module hinzufügen, 82

 Übersicht, 22

 Vergleichstabelle, 18

Kommunikationsschnittstellen

 CB 1241 RS485, 366

 CM 1241 RS232, 368

 Module hinzufügen, 82

 RS232 und RS485, 180

 Vergleichstabelle der Module, 18

Konfiguration

 Anlaufparameter, 84

 ASi,

 Benutzerdefinierte Webseiten, 218

 Erkennen, 80, 299

 HSC (schneller Zähler), 142

 Industrial Ethernet-Port, 89

 IP-Adresse, 89

 Module hinzufügen, 82

 Netzwerkverbindung, 144

 PROFIBUS, 162

 PROFIBUS-Adresse, 163

 PROFINET, 89

Konfiguration Sendeparameter, 156, 171

Konfigurieren von Parametern

 CPU, 84, 88

 Module, 88

Konsistenzprüfung, 130

Kontaktdaten, 6, 82

Kontakte

 Programmierung, 48

KOP (Kontaktplan)

 Beobachten, 287, 288

 Programmiereditor, 288

 Übersicht, 106

 Zustand, 287, 288, 290

Kopierschutz

 Mit CPU oder Memory Card verknüpfen, 95

Kühlen, 27

Kunden-Support, 6

L

Laden aus CPU

 Anwenderprogramm, 294

 Bausteine aus einer Online-CPU kopieren, 294

Laden im Betriebszustand RUN

 aus STEP 7, 303

 Übersicht, 301

Ladespeicher, 16, 67

 Benutzerdefinierte Webseiten, 217

Latenz, 63

Lebensdauer eines Relais, 317

LED (LED-Status lesen), 131

LED-Anzeigen

 Anweisung LED, 131

 Kommunikationsschnittstelle, 180

Lineare Programmierung, 97

Logic Analyzer, 304

Luftströmung, 27

M

MAC-Adresse, 89
 Maximale Anzahl Webserver-Verbindungen, 213
 MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern), 260
 MC_CommandTable, 257
 MC_Halt (Achse pausieren), 245
 MC_Home (Referenzpunktfahrt der Achse durchführen), 242
 MC_MoveAbsolute (Achse absolut positionieren), 247
 MC_MoveJog (Achse im Tippbetrieb bewegen), 254
 MC_MoveRelative (Achse relativ positionieren), 249
 MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 251
 MC_Power (Achse freigeben/sperrern), 238
 MC_ReadParam (Parameter eines Technologieobjekts lesen), 264
 MC_Reset (Fehler bestätigen), 241
 MC_WriteParam (Parameter des Technologieobjekts schreiben), 262
 Memory Card
 Ladespeicher, 67
 Mit CPU oder Memory Card verknüpfen, 95
 Modbus
 Versionen, 39
 Module
 Kommunikationsboards (CB), 21
 Kommunikationsmodul (CM), 22
 Kommunikationsprozessor (CP), 22
 Konfigurieren von Parametern, 84, 88
 Signalboard (SB), 21
 Signalmodule (SM), 21
 Thermischer Bereich, 24, 27
 Vergleichstabelle, 18
 ModuleStates, 132
 Montage
 Abmessungen, 24
 Abstand, 27
 Kühlung, 27
 Luftströmung, 27
 Richtlinien, 27
 Thermischer Bereich, 24, 27
 MRES
 Bedienpanel, 36, 61, 286
 My Documentation Manager, 5

N

Netzwerk
 Erste Schritte, 48, 51
 Netzwerkverbindung, 55
 Netzwerkkommunikation, 143
 Netzwerkverbindung
 Konfiguration, 144
 Netzwerkverbindung erstellen
 zwischen PLC- und HMI-Geräten, 55
 Zwischen PLC-Geräten, 144
 Neues Gerät hinzufügen
 Nicht spezifizierte CPU, 80, 299
 Vorhandene Hardware erkennen, 80
 Neues Projekt
 Erste Schritte, 45
 Hinzufügen eines HMI-Geräts, 54
 HMI-Bild, 56
 HMI-Verbindung, 55
 Netzwerkverbindung, 55
 Nicht gesteckte Module, 43
 Nicht spezifizierte CPU, 80, 299
 NORM_X (Normalisieren), 116

O

OB, (Organisationsbaustein)
 Online
 Bedienpanel, 36, 61, 286
 Beobachten, 287
 Beobachtungstabelle, 287, 288
 Erkennen, 299
 Forcefunktion, 291
 Forcen, 290
 IP-Adresse, 296
 Online gehen, 285
 Schaltflächen RUN/STOP, 36
 Startwerte eines DBs zurücksetzen, 293
 Tageszeit, 296
 Überwachung der Speicherauslastung, 286
 Vergleichen und synchronisieren, 295
 Werte eines DBs erfassen, 293
 Zustand, 287, 288
 Zykluszeitüberwachung, 286
 Online- und Diagnose-Tools
 Laden im Betriebszustand RUN, 301
 Online- und Offline-CPU's vergleichen und synchronisieren, 295
 OPC, 174
 Organisationsbaustein
 Aufrufen, 63
 Bearbeiten, 99

Erstellen, 100
 Funktion, 63
 Funktionsweise konfigurieren, 100
 Knowhow-Schutz, 94
 Mehrere Zyklus-OBs, 100
 Prioritätsklassen, 63

P

Panels (HMI), 23
 Parameter konfigurieren
 Ethernet-Port, 89
 Module, 84
 PROFINET, 89
 Parametrieren, 101
 Passive/aktive Kommunikation
 Parameter, 153
 Partner konfigurieren, 156, 171
 Verbindungs-IDs, 150
 Passwortschutz
 Codebaustein, 94
 CPU, 91
 Kopierschutz, 95
 Mit CPU oder Memory Card verknüpfen, 95
 Zugriff auf die CPU, 91
 PID
 Algorithmus PID_3Step, 186
 Algorithmus PID_Compact, 186
 PID_3STEP (PID-Regler mit Einstellung für Ventile), 195
 PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 189
 Übersicht, 186
 PID-Ventileinstellung, 195
 PLC
 Anweisungen, 50
 Baustein aufrufen, 105
 Bausteine aus einer Online-CPU kopieren, 294
 Bausteine verwenden, 62, 97
 Beobachten, 287
 Erste Schritte, 45
 Forcefunktion, 291
 Forcen, 290
 HSC-Konfiguration, 142
 Knowhow-Schutz, 94
 Module hinzufügen, 82
 Übersicht über die CPU, 15
 Variablen, 46, 50
 Vergleichen und synchronisieren, 295
 PLC-Variablen
 Erste Schritte, 46, 50
 PM 1207 Stromversorgungsmodul, 374

Podcasts, 5
 Portalansicht, 33
 Ethernet-Port konfigurieren, 89
 Konfigurieren der CPU, 84, 88
 Konfigurieren der Module, 84, 88
 Neues Gerät hinzufügen, 81
 PROFINET, 89
 Portnummer, 149
 Priorität
 Priorität bei der Verarbeitung, 63
 Prioritätsklassen, 63
 PROFIBUS
 Adresse, 163
 Adresse konfigurieren, 163
 CM 1242-5 (DP-Slave), 159
 CM 1243-5 (DP-Master), 159
 CM 1243-5 (DP-Master) hinzufügen, 161
 DP-Slave hinzufügen, 161
 Eigenschaften PROFIBUS-Adresse, 163
 GET, 169
 Master, 158
 Netzwerkverbindung, 144
 PROFIBUS-Adresse, 163
 PUT, 169
 S7-Verbindung, 170
 Slave, 158
 PROFIBUS-Adresse, 163
 PROFINET, 143
 Ad-hoc-Modus, 150
 Anzahl der Kommunikationsverbindungen, 147
 GET, 169
 IP-Adresse, 89
 Kommunikationsarten, 145
 Netzwerkverbindung, 144
 PUT, 169
 S7-Verbindung, 170
 Testen eines Netzwerks, 90
 Übersicht, 149
 Verbindungs-IDs, 150
 PROFINET RT, 149
 PROFINET-Schnittstelle
 Eigenschaften der Ethernet-Adresse, 89
 Programm
 Arithmetische Anweisung, 51
 Baustein aufrufen, 105
 Bausteine aus einer Online-CPU kopieren, 294
 Beispielnetzwerk, 48, 51
 Erste Schritte, 48, 51
 Mit CPU oder Memory Card verknüpfen, 95
 Organisationsbausteine (OBs), 99
 Passwortschutz, 94
 Prioritätsklassen, 63

- Startwerte eines DBs zurücksetzen, 293
 - Werte eines DBs erfassen, 293
 - Programm beobachten, 128
 - Programm testen, 128
 - Programmausführung
 - Bausteinstruktur, 62
 - Überblick, 59
 - Programmbaustein
 - Anlegen, 103
 - Erste Schritte, 45
 - Programmiereditor
 - Beobachten, 288
 - Startwerte eines DBs zurücksetzen, 293
 - Werte eines DBs erfassen, 293
 - Zustand, 288
 - Programmierung
 - Algorithmus PID_3Step, 186
 - Algorithmus PID_Compact, 186
 - Codebausteine vergleichen und synchronisieren, 295
 - Drag & Drop zwischen Editoren, 40
 - Einfügen von Anweisungen, 34
 - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 35
 - Erste Schritte, 50
 - Erweiterbare Anweisungen, 36
 - Favoriten, 35
 - FUP (Funktionsplan), 107
 - KOP (Kontaktplan), 106
 - Linear, 97
 - Mit CPU oder Memory Card verknüpfen, 95
 - Nicht gesteckte Module, 43
 - Nicht spezifizierte CPU, 80, 299
 - PID_3STEP (PID-Regler mit Einstellung für Ventile), 195
 - PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 189
 - Prioritätsklassen, 63
 - SCL (Structured Control Language), 107, 108
 - Strukturiert, 97
 - Übersicht über den PID-Regler, 186
 - Programminformationen
 - In der Aufrufstruktur, 130
 - Programmkarte, 67
 - Programmstruktur, 97
 - Projekt
 - Codebaustein schützen, 94
 - Erste Schritte, 45
 - Hinzufügen eines HMI-Geräts, 54
 - HMI-Bild, 56
 - HMI-Verbindung, 55
 - Mit CPU oder Memory Card verknüpfen, 95
 - Netzwerkverbindung, 55
 - Programm, 50
 - Variablen, 46, 50
 - Vergleichen und synchronisieren, 295
 - Zugang zur CPU einschränken, 91
 - Zugriffsschutz, 91
 - Projektansicht, 33, 34
 - Ethernet-Port konfigurieren, 89
 - Gerätekonfiguration, 79
 - Konfigurieren der CPU-Parameter, 84, 88
 - Konfigurieren der Module, 84, 88
 - Neues Gerät hinzufügen, 81
 - PROFINET, 89
 - Protokoll
 - Frei programmierbare Kommunikation, 179
 - ISO on TCP, 149
 - Kommunikation, 179
 - Modbus, 179
 - PROFINET RT, 149
 - TCP, 149
 - UDP, 149
 - USS, 179
 - Protokoll der frei programmierbaren Kommunikation, 179
 - Prozessabbild
 - Beobachten, 287, 288
 - Forcefunktion, 291
 - Forcen, 290
 - Zustand, 287, 288, 290
 - PTO (Impulsfolge), 125
 - Kann nicht geforct werden, 291
 - PtP-Kommunikation, 179
 - Punkt-zu-Punkt-Kommunikation, 179
 - PUT, 169
 - Verbindung konfigurieren, 157
 - PWM
 - Anweisung CTRL_PWM, 125
 - PWM (Impulsdauermodulation)
 - Kann nicht geforct werden, 291
- ## Q
- Querverweis zum Anzeigen der Verwendung, 129
- ## R
- Remanenter Speicher, 16, 67
 - Richtlinien
 - Installation, 27
 - ROUND, 115
 - RS232- und RS485-Kommunikationsmodule, 180

RUN, Betriebszustand
 Programmausführung, 59

S

S7-1200

Abstand, 27
 Anlaufparameter, 84
 ASi,
 ASi-Adresse,
 Baustein aufrufen, 105
 Bedienpanel, 36, 61, 286
 Beobachten, 287
 Codebausteine vergleichen, 295
 Diagnosepuffer, 296
 Ethernet-Port, 89
 Forcefunktion, 291
 Forcen, 290
 Gerätekonfiguration, 79
 HSC-Konfiguration, 142
 IP-Adresse, 89
 Knowhow-Schutz, 94
 Konfigurieren der CPU-Parameter, 84, 88
 Konfigurieren der Module, 84, 88
 Kühlung, 27
 Luftströmung, 27
 Module, 18
 Module hinzufügen, 82
 Montageabmessungen, 24
 Netzwerkverbindung, 144
 Neues Gerät hinzufügen, 81
 Passwortschutz, 91
 PROFIBUS, 162
 PROFIBUS-Adresse, 163
 PROFINET, 89
 Programmausführung, 59
 Schaltflächen RUN/STOP, 36
 Startwerte eines DBs zurücksetzen, 293
 Thermischer Bereich, 24, 27
 TS-Adapter, 18
 Übersicht über die CPU, 15
 Vergleichstabelle der CPU-Varianten, 16
 Werte eines DBs erfassen, 293
 Zugriffsschutz, 91

S7-Kommunikation

Verbindung konfigurieren, 157

SCALE_X (Skalieren), 116

Schaltflächen RUN/STOP, 36

Schaltpläne

CB 1241 RS 485, 366
 CPU 1214C AC/DC/Relais, 322
 CPU 1214C DC/DC/DC, 323

SB 1221 DI 4, 200 kHz, 325
 SB 1222 DO 4, 200 kHz, 325
 SB 1223 DI 2/DO 2, 200 kHz, 325
 SB 1231 AI 1 x 12 Bit, 337
 SB 1231 AI 1 x 16 Bit RTD, 349
 SB 1231 AI 1 x 16 Bit TC, 349
 SB 1232 AO 1 x 12 Bit, 337
 SM 1221 DI 16, 24 V DC, 327
 SM 1221 DI 8, 24 V DC, 327
 SM 1222 DO 16 Relais, 328
 SM 1222 DO 16, 24 V DC, 328
 SM 1223 DI 16 V DC/DO 16 Relais, 330
 SM 1223 DI 16 V DC/DO 16, 24 V DC, 330
 SM 1223 DI 8 120/230 V AC/DO 8 Relais, 331
 SM 1231 AI 8 x 13 Bit, 339
 SM 1231 RTD 4 x 16 Bit, 351
 SM 1231 RTD 8 x 16 Bit, 351
 SM 1232 AO 4 x 13 Bit, 339
 SM 1234 AI 4 x 13 Bit/AO 2 x 14 Bit, 339
 SM 1278 IO-Link-Master, 373

Schneller Zähler

Funktionsweise, 134
 HSC, 133
 Kann nicht geforct werden, 291
 Konfiguration, 142

Schutz

CPU, 91
 Knowhow-Schutz für einen Codebaustein, 94

Schutzart, 316

Schutzgrad, 316

Schutzstufe

Codebaustein, 94
 CPU, 91
 Mit CPU oder Memory Card verknüpfen, 95

SCL (Structured Control Language)

Algorithmus PID_3Step, 186
 Algorithmus PID_Compact, 186
 CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 115
 CONV (Umwandeln), 114
 DeviceStates, 132
 FLOOR, 115
 Ganzzahl erzeugen, 115
 GET_DIAG, 132
 LED-Zustand, 131
 MC_ChangeDynamic (Dynamikeinstellungen der Achse ändern), 260
 MC_CommandTable, 257
 MC_Halt (Achse pausieren), 245
 MC_Home (Referenzpunktfahrt der Achse durchführen), 242

- MC_MoveAbsolute (Achse absolut positionieren), 247
- MC_MoveJog (Achse im Tipbetrieb bewegen), 254
- MC_MoveRelative (Achse relativ positionieren), 249
- MC_MoveVelocity (Achse mit vordefinierter Geschwindigkeit bewegen), 251
- MC_Power (Achse freigeben/sperrern), 238
- MC_ReadParam (Parameter eines Technologieobjekts lesen), 264
- MC_Reset (Fehler bestätigen), 241
- MC_WriteParam (Parameter des Technologieobjekts schreiben), 262
- ModuleStates, 132
- NORM_X (Normalisieren), 116
- PID_3STEP (PID-Regler mit Einstellung für Ventile), 195
- PID_Compact (Universal-PID-Regler mit integrierter Einstellung), 189
- Programmiereditor, 108
- SCALE_X (Skalieren), 116
- Übersicht, 107
- Übersicht über den PID-Regler, 186
- Var-Abschnitt, 108
- Zahl runden, 115
- Serielle Kommunikation, 179
- Service und Support, 6
- Sicherheit
 - Kopierschutz, 95
 - Mit CPU oder Memory Card verknüpfen, 95
 - Zugriffsschutz, 91
- Signalboard (SB)
 - Schrittantwortzeiten der analogen Eingänge, 343
- Signalboard (SM)
 - Neues Gerät hinzufügen, 81
- Signalboards (SB)
 - Darstellung Analogausgang (Spannung), 346
 - Darstellung Analogausgang (Strom), 346
 - Darstellung Eingang (Spannung), 342
 - Darstellung Eingang (Strom), 342
 - Gerätekonfiguration, 79
 - Module hinzufügen, 82
 - SB 1221 4 DI, 24 V DC, 200 kHz, 323
 - SB 1221 4 DI, 5 V DC, 200 kHz, 323
 - SB 1222 4 DO, 24 V DC, 200 kHz, 323
 - SB 1222 4 DO, 5 V DC, 200 kHz, 323
 - SB 1223 2 DI/2 DO, 24 V DC, 324
 - SB 1223 DI/DO, 24 V DC, 200 kHz, 324
 - SB 1223 DI/DO, 5 V DC, 200 kHz, 324
 - SB 1231 AI 1 x 12 Bit, 336
 - SB 1231 AI 1 x 16 Bit RTD, 348
 - SB 1231 AI 1 x 16 Bit TC, 348
 - SB 1231 RTD, 348
 - SB 1231 TC, 348
 - SB 1232 AO 1 x 12 Bit, 336
- Übersicht, 21
- Signalmodul (SM)
 - Neues Gerät hinzufügen, 81
 - Schrittantwortzeiten der analogen Eingänge, 343
- Signalmodule (SM)
 - Darstellung Analogausgang (Spannung), 346
 - Darstellung Analogausgang (Strom), 346
 - Darstellung Analogeingang (Spannung), 342
 - Darstellung Analogeingang (Strom), 342
 - Gerätekonfiguration, 79
 - Module hinzufügen, 82
 - SM 1221 DI 16, 24 V DC, 326
 - SM 1221 DI 8, 24 V DC, 326
 - SM 1222 DO 16 Relais, 327
 - SM 1222 DO 16, 24 V DC, 327
 - SM 1222 DO 8 Relais, 327
 - SM 1222 DO 8 Relais (Umschaltung), 327
 - SM 1222 DO 8, 24 V DC, 327
 - SM 1223 DI 1223 DI 120/230 V AC/DO Relais, 330
 - SM 1223 DI 16/DO 16 Relais, 329
 - SM 1223 DI 8/DO 8, 329
 - SM 1223 DI 8/DO 8/DO Relais, 329
 - SM 1231 AI 4 x 13 Bit, 337
 - SM 1231 AI 4 x 16 Bit, 337
 - SM 1231 AI 4 x 16 Bit TC, 352
 - SM 1231 AI 4 x RTD x 16 Bit, 350
 - SM 1231 AI 8 x 13 Bit, 337
 - SM 1231 AI 8 x 16 Bit TC, 352
 - SM 1231 AI 8 x RTD x 16 Bit, 350
 - SM 1232 AO 2 x 14 Bit, 338
 - SM 1232 AO 4 x 14 Bit, 338
 - SM 1234 AI 4 x 13 Bit/AO 2 x 14 Bit, 338
 - SM 1278 4xIO-Link-Master, 370
 - Übersicht, 21
- SM und SB
 - Vergleichstabelle, 18
- SMS, 173
- Spalten und Überschriften in Taskcards, 39
- Spannungsversorgungsmodul
 - PM1207, 374
- Speicher
 - Adressen der Peripherieeingänge (Forcetable), 290
 - Arbeitsspeicher, 67
 - Ladespeicher, 67
 - Remanenter Speicher, 67
 - Systemspeicher, 86
 - Taktmerker, 86
 - Temporärer Speicher (L), 71
- Speicherbereich E

- Adressen der Peripherieeingänge (Forcetabelle), 290
 - Beobachten, 287
 - Beobachtungstabelle, 287
 - Forcefunktion, 291
 - Forcen, 290
 - Forcetabelle, 290
 - KOP beobachten, 288
 - Speicherbereiche
 - Adressierung von Booleschen Werten oder Bitwerten, 72
 - Datenbaustein, 71
 - Direkter Zugriff, 71
 - Globaler Speicher, 71
 - Prozessabbild, 71
 - Temporärer Speicher, 71
 - Standard-Webseiten, 209, 210
 - Cookie-Einschränkungen, 215
 - JavaScript-Einschränkungen, 214
 - STARTUP, Betriebszustand
 - Forcefunktion, 291
 - Programmausführung, 59
 - Startwerte eines DBs zurücksetzen, 293
 - STEP 7
 - Ändern von Einstellungen, 38
 - ASi,
 - Bausteine aus einer Online-CPU kopieren, 294
 - Bedienpanel, 36, 61, 286
 - Beobachten, 287, 288
 - Diagnosepuffer, 296
 - Drag & Drop zwischen Editoren, 40
 - Einfügen von Anweisungen, 34
 - Ergänzen von Eingängen oder Ausgängen in einer KOP- oder FUP-Anweisung, 35
 - Erweiterbare Eingänge und Ausgänge, 36
 - Ethernet-Port, 89
 - Favoriten, 35
 - Forcefunktion, 291
 - Forcen, 290
 - Gerätekonfiguration, 79
 - HSC-Konfiguration, 142
 - Konfigurieren der CPU, 84, 88
 - Konfigurieren der Module, 84, 88
 - Module hinzufügen, 82
 - Netzwerkverbindung, 144
 - Neues Gerät hinzufügen, 81
 - Nicht gesteckte Module, 43
 - Passwortschutz, 94
 - Portalansicht, 33
 - Prioritätsklasse (OB), 63
 - PROFIBUS, 162
 - PROFINET, 89
 - Projektansicht, 33
 - Schaltflächen RUN/STOP, 36
 - Startwerte eines DBs zurücksetzen, 293
 - Vergleichen und synchronisieren, 295
 - Werte eines DBs erfassen, 293
 - STEP 7 Webseiten, 5
 - STEP 7-Programmierung
 - Benutzerdefinierte Webseiten, 219
 - Steuer-DB für benutzerdefinierte Webseiten
 - Parameter WWW-Anweisung, 219
 - STOP, Betriebszustand
 - Bedienpanel, 36
 - Stoßwellenfestigkeit, 314
 - Strukturierte Programmierung, 97
 - Subnetzmaske, 89
 - Support, 6
 - Systemspeicherbyte, 87
- T**
- Tageszeit
 - CPU online konfigurieren, 296
 - Taktmerkerbyte, 87
 - Taskcards
 - Spalten und Überschriften, 39
 - TCON
 - Konfiguration, 156
 - Verbindungs-IDs, 150
 - Verbindungsparameter, 153
 - TCON_Param, 153
 - TCP
 - Ad-hoc-Modus, 150
 - Parameter, 153
 - Protokolle, 149
 - Verbindungs-IDs, 150
 - Verbindungskonfiguration, 156
 - TCP/IP-Kommunikation, 143, 149
 - Technische Daten, 311
 - Allgemeine technische Daten, 311
 - Bemessungsspannungen, 316
 - CB 1241 RS485, 366
 - CM 1241 RS232, 368
 - CM 1241 RS422/485, 367
 - CPU 1211C, 319
 - CPU 1212C, 319
 - CPU 1214C, 319
 - CPU 1215C, 319
 - CPU 1217C, 319
 - Darstellung Analogausgang (Spannung), 346
 - Darstellung Analogausgang (Strom), 346
 - Darstellung Analogeingang (Spannung), 342
 - Darstellung Analogeingang (Strom), 342

Elektromagnetische Verträglichkeit (EMV), 314
 Industrieumgebungen, 313
 SB 1221 4 DI, 24 V DC, 200 kHz, 323
 SB 1221 4 DI, 5 V DC, 200 kHz, 323
 SB 1222 4 DO, 24 V DC, 200 kHz, 323
 SB 1222 4 DO, 5 V DC, 200 kHz, 323
 SB 1223 2 DI/2 DO, 24 V DC, 324
 SB 1223 DI/DO, 24 V DC, 200 kHz, 324
 SB 1223 DI/DO, 5 V DC, 200 kHz, 324
 SB 1231 AI 1 x 12 Bit, 336
 SB 1231 AI 1 x 16 Bit RTD, 348
 SB 1231 AI 1 x 16 Bit TC, 348
 SB 1232 AO 1 x 12 Bit, 336
 Schrittantwortzeiten der Eingänge, 343
 SM 1221 DI 16, 24 V DC, 326
 SM 1221 DI 8, 24 V DC, 326
 SM 1222 DO 16 Relais, 327
 SM 1222 DO 16, 24 V DC, 327
 SM 1222 DO 8 Relais, 327
 SM 1222 DO 8 Relais (Umschaltung), 327
 SM 1222 DO 8, 24 V DC, 327
 SM 1223 DI 120/230 V AC/DO Relais, 330
 SM 1223 DI 16/DO 16 Relais, 329
 SM 1223 DI 8/DO 8, 329
 SM 1223 DI 8/DO 8 Relais, 329
 SM 1231 AI 4 x 13 Bit, 337
 SM 1231 AI 4 x 16 Bit, 337
 SM 1231 AI 4 x 16 Bit TC, 352
 SM 1231 AI 4 x RTD x 16 Bit, 350
 SM 1231 AI 8 x 13 Bit, 337
 SM 1231 AI 8 x 16 Bit TC, 352
 SM 1231 AI 8 x RTD x 16 Bit, 350
 SM 1232 AO 2 x 14 Bit, 338
 SM 1232 AO 4 x 14 Bit, 338
 SM 1234 AI 4 x 13 Bit/AO 2 x 14 Bit, 338
 SM 1278 4xIO-Link-Master, 370
 Umgebungsbedingungen, 315
 Zulassungen, 311
 Technischer Support, 6
 Technischer Support von Siemens, 6
 Technologiemodul
 SM 1278, 370
 Technologieobjekte
 Bewegungssteuerung, 229
 PID, 187
 Telecontrol, 176
 TeleService über GPRS, 175, 176
 Temporärer Speicher (L), 71
 Testen
 Laden im Betriebszustand RUN, 301
 Thermischer Bereich, 24, 27
 TIA Portal

Konfigurieren der CPU, 88
 TIA-Portal
 Gerätekonfiguration, 79
 Konfigurieren der CPU, 84
 Konfigurieren der Module, 88
 Neues Gerät hinzufügen, 81
 Portalansicht, 33
 PROFINET, 89
 Projektansicht, 33
 Trace-Funktion, 304
 TRCV
 Ad-hoc-Modus, 150
 Verbindungs-IDs, 150
 TRCV_C
 Ad-hoc-Modus, 150
 Konfiguration, 156
 Verbindungs-IDs, 150
 Verbindungsparameter, 153
 TRUNC (Ganzzahl erzeugen), 115
 TS-Adapter, 18
 TSAP, 149
 TSAP (Transport Service Access Point), 158
 TSEND
 Verbindungs-IDs, 150
 TSEND_C
 Konfiguration, 156
 Verbindungs-IDs, 150
 Verbindungsparameter, 153
 TURCV
 Konfiguration, 156
 Verbindungsparameter, 153
 TUSEND
 Konfiguration, 156
 Parameter, 153

U

Übersicht über RTD-Module, 347
 Übersicht über Thermoelementmodule, 347
 Überwachung
 Anweisung LED, 131
 Überwachung der Speicherauslastung, online, 286
 UDP
 Parameter, 153
 Verbindungskonfiguration, 156
 UDP-Protokoll, 149
 Umgebungsbedingungen
 Betriebsbedingungen, 315
 Transport und Lagerung, 315

V

- Variablen
 - Beobachten, 287
 - Erste Schritte, 46, 50
 - Forcefunktion, 291
 - Forcen, 290
 - Zustand, 287
- Verbindungen
 - Partner, 171
- Verbindungen mit mehreren Teilnehmern
 - Ethernet-Protokolle, 170
 - Verbindungsarten, 170
- Verdrahtungsrichtlinien
 - Abstand für Luftströmung und Kühlung, 27
- Vergleichsanweisungen, 113
- Vergleichstabelle
 - CPU-Varianten, 16
 - HMI-Geräte, 23
 - Module, 18
- Verpolschutz, 317
- Versionen von Anweisungen, 39
- Visualisierung
 - HMI-Geräte, 23

W

- Warteschlangen, 63
- Webseiten
 - STEP 7-Service, -Support und -Dokumentation, 5
- Webseiten, benutzerdefiniert, 216
- Webserver, 209
 - Einschränkungen, 213
 - maximale Anzahl HTTP-Verbindungen, 213
- Werte eines Online-DBs erfassen, 293
- WWW, 219

Z

- Zähler
 - Anzahl, 18
 - Funktionsweise des HSC, 134
 - Größe, 18
 - HSC (schneller Zähler), 133
 - HSC-Konfiguration, 142
- Zähleranweisungen, 122
- Zeiten
 - Anzahl, 17
 - Größe, 17
- Zugriffsschutz, CPU, 91
- Zulassung für das Seewesen, 313
- Zulassung koreanische Zertifizierung, 313

Zulassungen

- ATEX, 312
- CE, 311
- C-Tick, 313
- cULus, 312
- FM, 312
- Koreanische Zertifizierung, 313
- Seewesen, 313
- Zurücksetzen auf Werkseinstellungen, 297
- Zustand
 - Anweisung LED, 131
 - LED-Anzeigen (Kommunikationsschnittstelle), 180
- Zyklus
 - Forcefunktion, 291
 - Forcen, 290
- Zykluszeitüberwachung, 286

