

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
Technische Daten	A

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

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

ACHTUNG
bedeutet, dass ein unerwünschtes Ergebnis oder Zustand eintreten kann, wenn der entsprechende Hinweis nicht beachtet wird.

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, dem jüngsten Produkt aus der Reihe der SIMATIC-Steuerungen von Siemens. 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 das Programmierwerkzeug STEP 7 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 SIMATIC STEP 7 programmierbar. Diese geräteübergreifende Programmierbarkeit bedeutet eine erhebliche Einsparung von Entwicklungskosten.



Die kompakte Steuerung S7-1200 umfasst:

- Integriertes PROFINET
- Hochgeschwindigkeits-E/A für Bewegungssteuerung, integrierte analoge Eingänge zur Minimierung des Platzbedarfs und des Nachrüstbedarfs an zusätzlichen E/A, 2 Impulsgeneratoren für Impulsdaueranwendungen (Seite 115) 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 16) durch Signalmodule für Gleichstrom-, Relais- oder analoge E/A sowie durch an der CPU-Vorderseite einzurastende innovative Signalboards.

Die SIMATIC HMI Basic Panels 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 249).

Dieses Handbuch beschreibt die folgenden Produkte:

- STEP 7 V11 Basic und Professional
- S7-1200 CPU Firmware-Version V2.2

Dokumentation und Information

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

- Das S7-1200 Systemhandbuch bietet spezifische Informationen zu Funktionsweise, Programmierung und technischen Daten der gesamten S7-1200 Produktreihe. 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) und als gedrucktes Handbuch 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 spezifische Anweisungen zu 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, einschließlich Systemhandbuch, Easy Book und Informationssystem von STEP 7. In My Documentation Manager können Sie Themen aus verschiedenen Dokumenten per Drag&Drop 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 andere hilfreiche Dokumente für S7-1200 und STEP 7. Die Podcasts zeigen kurze Lernvideos zu spezifischen Funktionen oder Szenarien, um die Interaktionen, die praktische Bedienung und Effizienz von STEP 7 zu demonstrieren. Die Sammlung der Podcasts 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 bieten wir Ihnen im Internet unser technisches Know-how auf der Kundensupport-Website (<http://www.siemens.com/automation/support-request>) 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.

Inhaltsverzeichnis

	Vorwort	3
1	Einführung in die leistungsstarke und flexible S7-1200	13
1.1	Einführung in die S7-1200 SPS	13
1.2	Erweiterung der CPU-Funktionen	16
1.3	S7-1200 Module.....	19
1.4	Grundlegende HMI-Panels	20
1.5	Einbaumaße und notwendiger Freiraum	21
1.6	Neue Funktionen bei der S7-1200 und in STEP 7 V11	24
2	STEP 7 vereinfacht Ihre Arbeit	27
2.1	Einfaches Einfügen von Anweisungen in Ihr Anwenderprogramm.....	28
2.2	Schneller Zugriff auf viel verwendete Operationen über die Funktionsleiste	29
2.3	Einfaches Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen	29
2.4	Erweiterbare Anweisungen	30
2.5	Einfaches Ändern des CPU-Betriebszustands	30
2.6	Ändern des Erscheinungsbilds und der Konfiguration von STEP 7	31
2.7	Projektbibliotheken und globale Bibliotheken für einfachen Zugriff.....	31
2.8	Einfache Auswahl einer Version einer Anweisung	32
2.9	Einfaches Drag & Drop zwischen Editoren.....	32
2.10	Bausteinzustände erfassen und wiederherstellen	33
2.11	Aufruftyp eines DB ändern.....	34
2.12	Geräte vorübergehend vom Netzwerk trennen.....	35
2.13	Einfaches virtuelles "Abziehen" von Baugruppen ohne Verlust der Konfiguration.....	36
3	Erste Schritte	37
3.1	Projekt anlegen	37
3.2	Variablen für die E/A der CPU anlegen	38
3.3	Einfaches Netzwerk im Anwenderprogramm anlegen.....	40
3.4	Adressieren Sie die Anweisungen mithilfe der PLC-Variablen in der Variablen-tabelle.....	42
3.5	"Box"-Anweisung hinzufügen.....	43
3.6	Anweisung CALCULATE für komplexe mathematische Gleichungen verwenden.....	44
3.7	HMI-Gerät zum Projekt hinzufügen	46
3.8	Netzwerkverbindung zwischen CPU und HMI-Gerät herstellen	47

3.9	HMI-Verbindung zur gemeinsamen Nutzung von Variablen erstellen	47
3.10	HMI-Bild anlegen.....	48
3.11	PLC-Variable für das HMI-Element auswählen	49
4	SPS-Grundlagen leicht gemacht.....	51
4.1	Bei jedem Zyklus ausgeführte Arbeitsschritte.....	51
4.2	Betriebszustände der CPU.....	52
4.3	Ausführung des Anwenderprogramms	53
4.3.1	Bearbeitung des Zyklus im Betriebszustand RUN	54
4.3.2	OBs strukturieren Ihr Anwenderprogramm	55
4.3.3	Prioritäten und Warteschlange für die Ausführung von Ereignissen	56
4.4	Speicherbereiche, Adressierung und Datentypen	62
4.4.1	Von der S7-1200 unterstützte Datentypen.....	64
4.4.2	Adressierung der Speicherbereiche.....	66
4.4.3	Zugriff auf eine "Scheibe" eines Variablendatentyps	68
4.4.4	Zugriff auf eine Variable mit einer AT-Überlagerung	70
4.5	Impulsausgänge.....	72
5	Einfache Erstellung der Gerätekonfiguration.....	75
5.1	Konfiguration für eine nicht spezifizierte CPU erkennen	76
5.2	CPU zur Konfiguration hinzufügen.....	77
5.3	Module zur Konfiguration hinzufügen	78
5.4	Konfigurieren des CPU- und Modulbetriebs	80
5.4.1	Systemmerker und Taktmerker bieten Standardfunktionen	82
5.5	Konfigurieren der IP-Adresse der CPU.....	84
5.6	Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach.....	86
5.6.1	Knowhow-Schutz	87
5.6.2	Kopierschutz	88
6	Programmierung leicht gemacht.....	91
6.1	Einfaches Entwerfen Ihres Anwenderprogramms	91
6.1.1	OBs für die Organisation Ihres Anwenderprogramms	93
6.1.2	FBs und FCs vereinfachen die Programmierung der modularen Aufgaben.....	94
6.1.3	Datenbausteine sorgen für die unkomplizierte Speicherung von Programmdateien	96
6.1.4	Neuen Codebaustein anlegen	96
6.1.5	Einen Codebaustein aus einem anderen Codebaustein aufrufen	97
6.2	Anwenderfreundliche Programmiersprachen.....	97
6.2.1	Kontaktplan (KOP)	97
6.2.2	Funktionsplan (FUP)	98
6.2.3	Überblick über SCL	99
6.2.4	SCL-Programmiereditor	100
6.3	Leistungsstarke Anweisungen erleichtern die Programmierung.....	101
6.3.1	Bereitstellung der erwartbaren grundlegenden Operationen.....	101
6.3.2	Anweisungen für Vergleichen und Verschieben	104
6.3.3	Umwandlungsanweisungen	105
6.3.4	Mathematik ganz einfach mit der Anweisung Calculate	107

6.3.5	Zeiten	108
6.3.6	Zähler	113
6.3.7	Impulsdauermodulation (PWM)	115
6.4	Einfache Erstellung von Datenprotokollen	116
6.5	Einfaches Überwachen und Testen Ihres Anwenderprogramms	119
6.5.1	Beobachtungstabellen und Forcetabellen	119
6.5.2	Querverweis zum Anzeigen der Verwendung	120
6.5.3	Aufrufstruktur zur Prüfung der Aufrufhierarchie	121
6.5.4	Diagnoseanweisungen zur Überwachung der Hardware	122
6.5.4.1	Zustände der LEDs der CPU lesen	122
6.5.4.2	Anweisungen zum Lesen des Diagnosezustands der Geräte	123
6.6	Schneller Zähler (HSC)	123
6.6.1	Funktionsweise eines schnellen Zählers	125
6.6.2	Konfiguration eines schnellen Zählers	129
7	Einfache Kommunikation zwischen Geräten	131
7.1	Netzwerkverbindung erstellen	132
7.2	Kommunikationsoptionen	133
7.3	Anzahl der asynchronen Kommunikationsverbindungen	134
7.4	PROFINET- und PROFIBUS-Anweisungen	134
7.5	PROFINET	135
7.5.1	Ad-hoc-Modus	136
7.5.2	Verbindungs-IDs für die PROFINET-Anweisungen	137
7.5.3	Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren	140
7.5.4	Parameter für die PROFINET-Verbindung	142
7.6	PROFIBUS	144
7.6.1	Anweisungen für die dezentrale E/A	146
7.6.2	Konfigurationsbeispiele für PROFIBUS	146
7.6.3	CM 1243-5 (DP-Master) und DP-Slave hinzufügen	150
7.6.4	PROFIBUS-Adressen zum CM 1243-5 und DP-Slave zuweisen	151
7.7	ASi	153
7.7.1	CM 1243-2 AS-i-Mastermodul und AS-i-Slave hinzufügen	153
7.7.2	Dem CM 1243-2 AS-i-Mastermodul und dem AS-i-Slave AS-i-Adressen zuweisen	154
7.8	S7-Kommunikation	156
7.8.1	Anweisungen GET und PUT	156
7.8.2	S7-Verbindung erstellen	156
7.8.3	Parametrieren der GET/PUT-Verbindung	157
7.9	GPRS	158
7.9.1	Anschluss an ein GSM-Netz	158
7.10	Kommunikationsprotokolle PtP, USS und Modbus	165
7.10.1	Arbeiten mit den RS232- und RS485-Kommunikationsschnittstellen	165
7.10.2	PtP-Operationen	166
7.10.3	USS-Anweisungen	167
7.10.4	Modbus-Anweisungen	168

8	Einfache PID-Regelung	171
8.1	Anweisung PID und Technologieobjekt einfügen	172
8.2	Operation PID_Compact	175
8.3	Anweisung PID_3STEP	178
8.4	PID-Regler konfigurieren.....	185
8.5	Inbetriebnahme des PID-Reglers.....	187
9	Webserver für einfachen Internetanschluss.....	189
9.1	Einfache Nutzung der Standard-Webseiten.....	189
9.2	Bedingungen, die sich auf die Nutzung des Webserver auswirken können	190
9.2.1	Einschränkungen bei deaktiviertem JavaScript	191
9.2.2	Eingeschränkte Funktionen, wenn keine Cookies zugelassen sind	192
9.3	Benutzerdefinierte Webseiten einfach anlegen	193
9.3.1	Eigene benutzerdefinierte Webseiten einfach anlegen	193
9.3.2	Einschränkungen bei benutzerdefinierten Webseiten	194
9.3.3	Konfiguration einer benutzerdefinierten Webseite	195
9.3.4	WWW-Anweisung verwenden.....	196
10	Einfache Bewegungssteuerung	197
10.1	Konfiguration der Achse.....	200
10.2	Anweisung MC_Power	203
10.3	Anweisung MC_Reset.....	206
10.4	Anweisung MC_Home.....	207
10.5	Anweisung MC_Halt.....	211
10.6	Anweisung MC_MoveAbsolute	213
10.7	Anweisung MC_MoveRelative	215
10.8	Anweisung MC_MoveVelocity.....	217
10.9	Anweisung MC_MoveJog.....	220
10.10	Anweisung MC_CommandTable	222
10.11	MC_ChangeDynamic	225
11	Einfaches Arbeiten mit den Online-Tools.....	229
11.1	Online-Verbindung mit einer CPU herstellen	229
11.2	Interaktion mit der Online-CPU	230
11.3	Online gehen, um die Werte in der CPU zu beobachten	231
11.4	Einfaches Anzeigen des Status des Anwenderprogramms.....	232
11.5	Beobachtungstabelle zur Überwachung der CPU verwenden.....	232
11.6	Arbeiten mit der Forcetabelle	233
11.7	Online-Werte eines DBs erfassen, um die Startwerte zurückzusetzen	237
11.8	Elemente des Projekts kopieren	238

11.9	Vergleichen von Offline- und Online-CPU's	239
11.10	Diagnoseereignisse anzeigen.....	240
11.11	Einstellen der IP-Adresse und der Uhrzeit.....	241
11.12	Auf Werkseinstellungen zurücksetzen.....	242
11.13	Laden einer permanenten IP-Adresse in eine Online-CPU.....	243
11.14	Verwendung der "nicht spezifizierten CPU" zum Laden der Hardwarekonfiguration aus dem Gerät	243
11.15	Laden im Betriebszustand RUN.....	245
11.15.1	Ändern des Programms im Betriebszustand RUN	247
A	Technische Daten.....	249
A.1	Allgemeine technische Daten	249
A.2	CPU-Module.....	255
A.3	Digitale Erweiterungsmodule	259
A.3.1	SB 1221, SB 1222 und SB 1223 Digitaleingabe/-ausgabe (DI, DO und DI/DO).....	259
A.3.2	SM 1221 Digitaleingabe (DI).....	262
A.3.3	SM 1222 Digitalausgabe (DO).....	263
A.3.4	SM 1223 V-DC-Digitaleingabe/-ausgabe (DI/DO)	264
A.3.5	SM 1223 120/230 V-AC-Eingabe/Relaisausgang.....	265
A.4	Technische Daten der digitalen Eingänge und Ausgänge.....	266
A.4.1	24-V-DC-Digitaleingabe (DI)	266
A.4.2	120/230 V-AC-Digitaleingabe	268
A.4.3	Digitalausgänge (DO)	268
A.5	Analoge Erweiterungsmodule	271
A.5.1	SB 1231 und SB 1232 Analogeingabe (AI) und Analogausgabe (AO).....	271
A.5.2	SM 1231 Analogeingabe (AI).....	272
A.5.3	SM 1232 Analogausgabe (AO).....	272
A.5.4	SM 1234 Analogeingabe/-ausgabe (AI/AO)	272
A.5.5	Schaltpläne SM 1231 (AI), SM 1232 (AO) und SM 1234 (AI/AO)	273
A.6	Technische Daten der analogen E/A	274
A.6.1	Technische Daten der analogen Eingänge (CPU, SM und SB)	274
A.6.2	Eingangsmessbereiche (AI) für Spannung und Strom	275
A.6.3	Schrittantwort der analogen Eingänge (AI).....	276
A.6.4	Abtastzeit und Aktualisierungszeiten der Analogeingänge.....	276
A.6.5	Technische Daten der analogen Ausgänge (SB und SM).....	277
A.6.6	Ausgangsmessbereiche (AO) für Spannung und Strom	278
A.7	RTD- und Thermoelementmodule	279
A.7.1	Technische Daten SB 1231 RTD und SB 1231 TC.....	280
A.7.2	Technische Daten SM 1231 RTD	281
A.7.3	Technische Daten SM 1231 TC.....	282
A.7.4	Technische Daten der analogen Eingänge für RTD und TC (SM und SB)	284
A.7.5	Thermoelementtyp	285
A.7.6	Filterauswahl und Aktualisierungszeiten beim Thermoelement	285
A.7.7	Auswahltabelle RTD-Gebertyp	286
A.7.8	Filterauswahl und Aktualisierungszeiten beim RTD	287
A.8	Kommunikationsschnittstellen.....	288

A.8.1	PROFIBUS-Master/Slave.....	288
A.8.1.1	CM 1242-5 PROFIBUS-Slave.....	288
A.8.1.2	CM 1243-5 PROFIBUS-Master.....	289
A.8.2	GPRS CP	291
A.8.2.1	Technische Daten des CP 1242-7	292
A.8.3	Teleservice (TS).....	294
A.8.4	RS485-, RS232- und RS422-Kommunikation.....	295
A.8.4.1	Technische Daten des CM 1241 RS485.....	295
A.8.4.2	CM 1241 RS485 und RS232.....	296
A.8.4.3	Technische Daten des CM 1241 RS422/485.....	298
A.9	Zugehörige Produkte.....	300
A.9.1	PM 1207 Stromversorgungsmodul	300
A.9.2	CSM 1277 Compact Switch Module	300
Index		301

Einführung in die leistungsstarke und flexible S7-1200

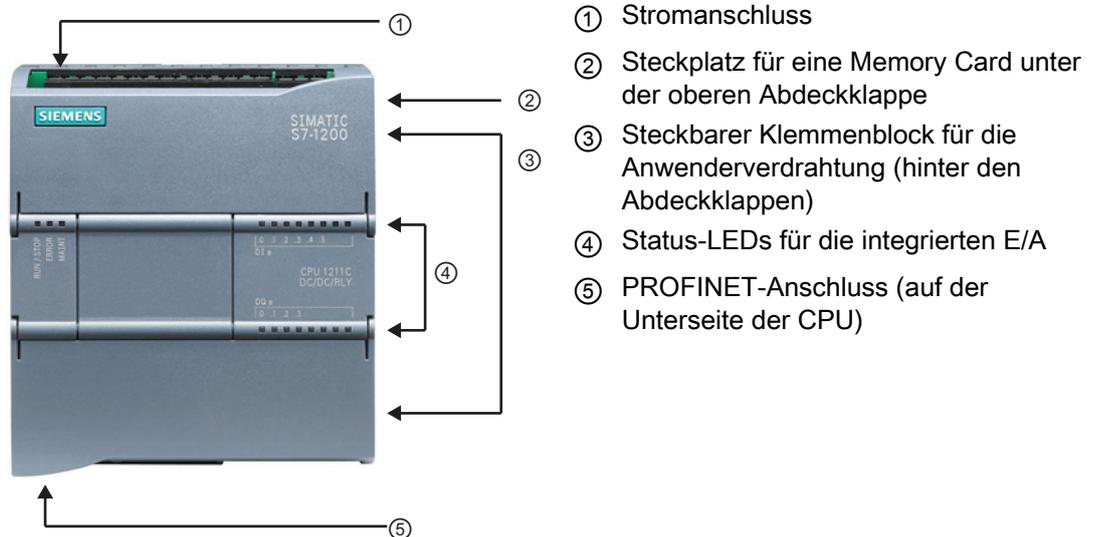
1

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.



1.1 Einführung in die S7-1200 SPS

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

- Jede CPU ist mit einem Passwortschutz (Seite 86) ausgestattet, mit dem der Zugriff auf die CPU-Funktionen nach Bedarf eingerichtet werden kann.
- Sie können mit dem Knowhow-Schutz (Seite 87) den Code in einem bestimmten Baustein verbergen.
- Mit dem Kopierschutz (Seite 88) können Sie Ihr Programm mit einer bestimmten Memory Card oder CPU verknüpfen.

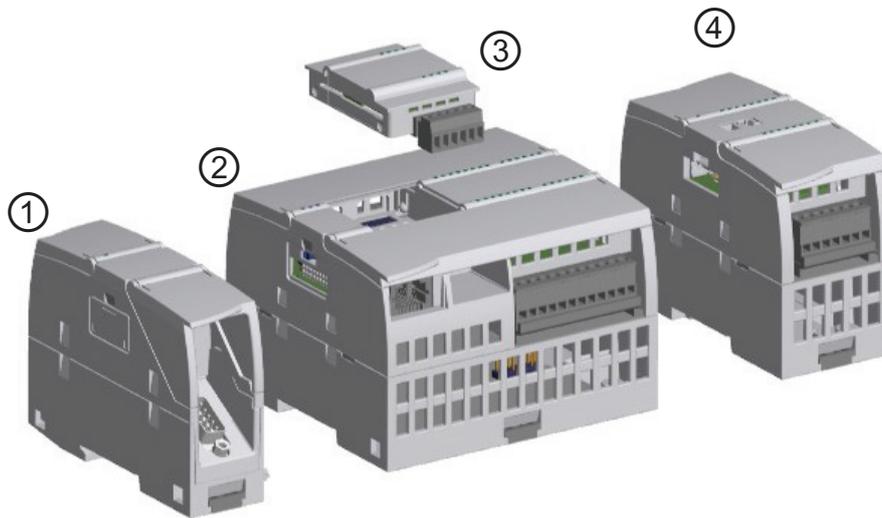
Tabelle 1- 1 Vergleich der CPU-Varianten

Merkmal		CPU 1211C	CPU 1212C	CPU 1214C
Abmessungen (mm)		90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
Anwenderspeicher	Arbeitsspeicher	25 KB	25 KB	50 KB
	Ladespeicher	1 MB	1 MB	2 MB
	Remanent	2 KB	2 KB	2 KB
Integrierte E/A	Digital	6 Eingänge/4 Ausgänge	8 Eingänge/6 Ausgänge	14 Eingänge/10 Ausgänge
	Analog	2 Eingänge	2 Eingänge	2 Eingänge
Größe des Prozessabblids	Eingänge (E)	1024 Byte	1024 Byte	1024 Byte
	Ausgang (A)	1024 Byte	1024 Byte	1024 Byte
Merker (M)		4096 Byte	4096 Byte	8192 Byte
Erweiterung: Signalmodul (SM)		Keine	2	8
Signalboard (SB) oder Kommunikationsboard (CB)		1	1	1
Kommunikationsmodule (CM) (Anbau links)		3	3	3
Schnelle Zähler	Summe	3	4	6
	Einphasenzähler	3 bei 100 kHz	3 bei 100 kHz 1 bei 30 kHz	3 bei 100 kHz 3 bei 30 kHz
	A/B-Zähler	3 bei 80 kHz	3 bei 80 kHz 1 bei 20 kHz	3 bei 80 kHz 3 bei 20 kHz
Impulsausgänge ¹		2	2	2
Memory Card		SIMATIC Memory Card (optional)		
Pufferung Echtzeituhr		Typ. 10 Tage/min. 6 Tage bei 40 Grad C		
PROFINET		1 Ethernet-Kommunikationsport		
Ausführungszeit arithm. Operationen		18 µs/Operation		
Ausführungszeit Boolesche Operationen		0,1 µs/Operation		

¹ Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

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 249).



- ① Kommunikationsmodul (CM), Kommunikationsprozessor (CP) oder TS-Adapter
- ② CPU
- ③ Signalboard (SB) oder Kommunikationsboard (CB)
- 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 Relaisausgang (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 • 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

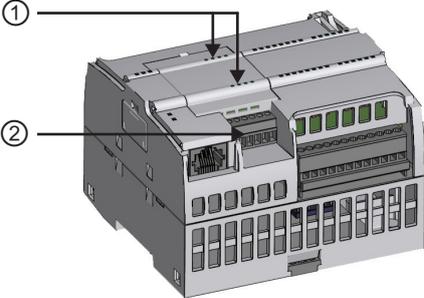
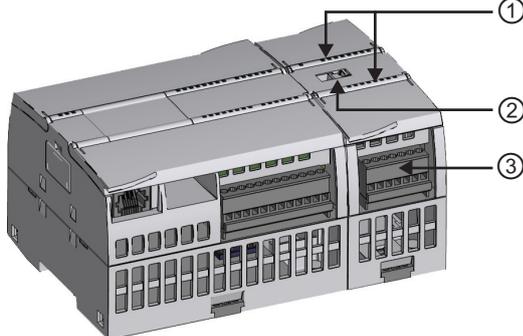
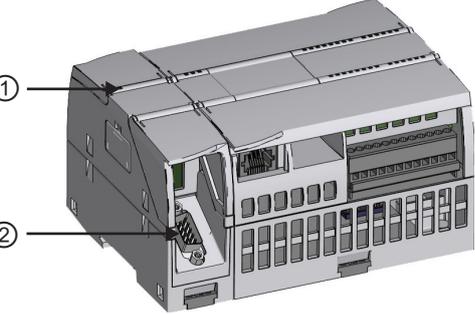
Tabelle 1- 5 Kommunikationsschnittstellen

Modul	Typ	Beschreibung
① Kommunikationsmodul (CM)	RS232	Vollduplex
	RS485	Halbduplex
	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
① Kommunikationsboard (CB)	RS485	Halbduplex
① TeleService	TS-Adapter IE Basic ¹	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 können Sie verschiedene Kommunikationsschnittstellen an den PROFINET-Port der CPU anschließen. Sie installieren den TS-Adapter an der linken Seite der CPU und stecken dann (bis zu 3) TS-Adaptermodule auf den TS-Adapter.

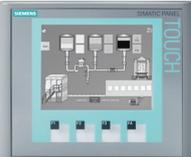
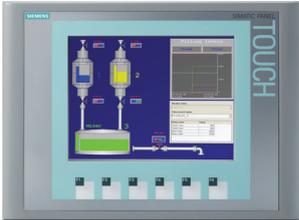
1.3 S7-1200 Module

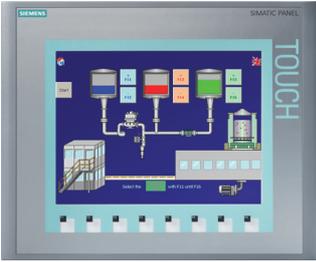
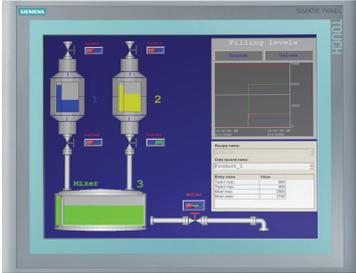
Tabelle 1-6 S7-1200 Erweiterungsmodule

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. 		①	Status-LEDs am SB
<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 		①	Status-LEDs
<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). 		①	Status-LEDs
		②	Kommunikationsstecker
		②	Steckbarer Klemmenblock für die Anwenderverdrahtung

1.4 Grundlegende HMI-Panels

Visualisierung gehört heute bei den meisten Maschinen zum Standardrepertoire. Deshalb bieten die SIMATIC HMI Basic Panels Geräte mit Touchscreen für grundlegende Aufgaben des Bedienens und Beobachtens. 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 Schalttafeleinbau ebenso wie bei der Montage auf einer Standard-Hutschiene gestattet die kompakte Baugröße eine optimale Platzausnutzung.

Alle CPUs, SMs, CMs und CPs können auf der DIN-Schiene oder im Schaltschrank montiert werden. Verwenden Sie die Hutschienenklemmen zur 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 Hutschienenklemmen am Gerät beträgt 4,3 mm.

1.5 Einbaumaße und notwendiger Freiraum

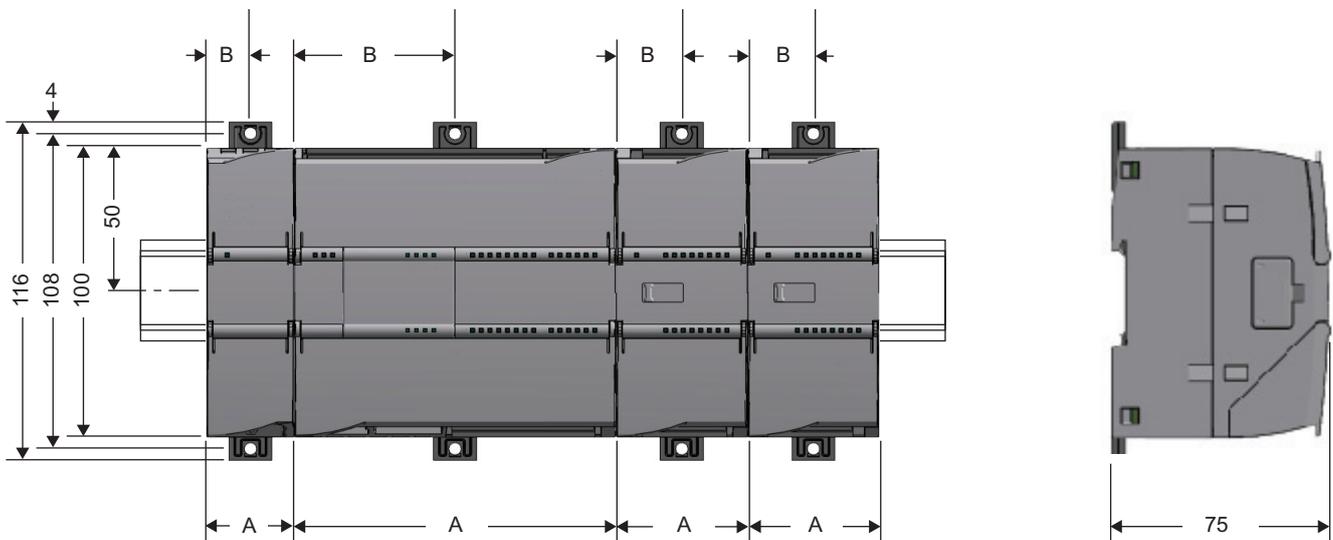


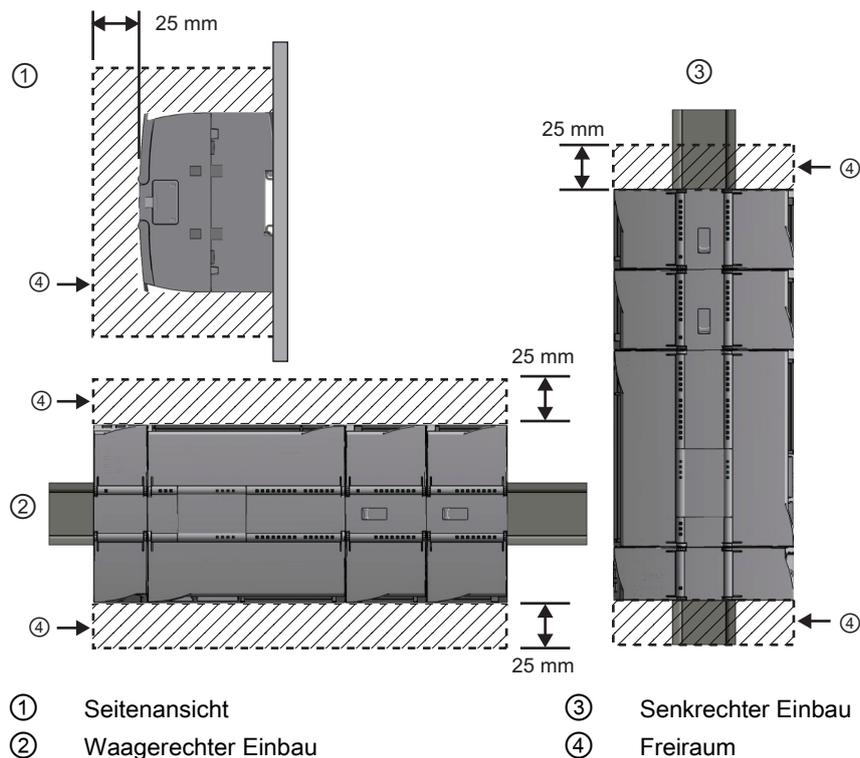
Tabelle 1- 7 Abmessungen für die Montage (mm)

S71200 Geräte		Breite A	Breite B
CPU	CPU 1211C und CPU 1212C	90 mm	45 mm
	CPU 1214C	110 mm	55 mm
Signalmodule	8 und 16 digitale E/A, 2, 4 und 8 analoge E/A, 4 und 8 Thermoelement-E/A, 4 RTD-E/A	45 mm	22,5 mm
	16 analoge E/A, 8 RTD-E/A	70 mm	35 mm
Kommunikationsschnittstellen	CM 1241 RS232, CM 1241 RS485	30 mm	15 mm
	CM 1243-5 PROFIBUS-Master, CM 1242-5 PROFIBUS-Slave	30 mm	15 mm
	CP 1242-7 GPRS	30 mm	15 mm
	Teleservice-Adapter IE Basic	30 mm	15 mm

Beachten Sie bei der Planung des Einbaus in jedem Fall die folgenden Hinweise:

- Halten Sie die Geräte fern von Wärme, Hochspannung und elektrischen Störungen.
- Lassen Sie genügend Abstand für Kühlung und Verdrahtung. Ober- und unterhalb des Geräts muss ein Freiraum von 25 mm zur Belüftung als Schutz vor Überhitzung eingehalten werden.

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



Ober- und unterhalb des Geräts muss ein Freiraum von 25 mm zur Belüftung als Schutz vor Überhitzung eingehalten werden.

! 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. austauschen.

 **WARNUNG**

Falscher Einbau eines S7-1200 Moduls kann zu unvorhersehbarer Funktionsweise 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 bei der S7-1200 und in STEP 7 V11

STEP 7 V11 und die Firmware V2.2 der S7-1200 CPU bieten zusätzliche Funktionen und Merkmale.

- Um Ihnen bei der Definition der Daten in Ihrem Anwenderprogramm größere Flexibilität zu bieten, unterstützt die S7-1200 zusätzliche Datentypen wie Pointer, indizierte Arrays und Strukturen.
- STEP 7 V11 führt implizite Datentypumwandlungen durch, und zwar bei Anweisungen, in denen kleinere Datentypen (wie SInt oder Byte) automatisch in größere Datentypen (wie DInt, DWord, Real oder LReal) umgewandelt werden. Beispiel: Ein ganzzahliger Wert (Int) wird von einer Anweisung, für die der Datentyp DInt oder Real erforderlich ist, automatisch in eine doppelte Ganzzahl (DInt) oder in den Datentyp Real umgewandelt. Sie benötigen keine zusätzliche Umwandlungsanweisung, um den Wert umzuwandeln.
- Der Befehlssatz wurde erweitert. Zu den neuen Anweisungen gehören die folgenden:
 - Die Kommunikationsanweisungen umfassen die Anweisungen GET und PUT für die S7-Kommunikation, die Anweisungen RDREC, WRREC und RALRM für die dezentrale Peripherie, die neuen PROFINET-Anweisungen TUSEND und TURCV sowie die Teleservice-Anweisungen GPRS und TM_MAIL.
 - Mit der Anweisung Calculate können Sie eine Gleichung direkt in Ihr KOP- oder FUP-Programm eingeben.
 - Eine neue Anweisung MC_CommandTable führt eine Reihe einzelner Bewegungen für eine Motorsteuerungsachse aus, die Sie zu einer Bewegungsfolge verbinden können. Einzelne Bewegungen werden in einer Befehlstabelle eines Technologieobjekts für die Impulsfolge konfiguriert (TO_CommandTable_PTO).
 - Mit zusätzlichen Alarmanweisungen können Sie Verzögerungs- und Weckalarme einstellen und abfragen.
 - Mit den neuen Diagnoseanweisungen können Sie den LED-Zustand oder andere Diagnoseinformationen über Module und Geräte auslesen.
 - Außerdem gibt es eine neue bedienungsfreundliche Anweisung PID_3Step.

- In Ihrer Programmlogik können Sie eine Variable als Array-Index für den Zugriff auf ein einzelnes Array-Element in KOP, FUP und SCL verwenden. Der Zugriff auf ein Array über eine Variable ist direkter als die Verwendung der KOP/FUP-Anweisungen FeldLesen und FeldSchreiben, die weiterhin vorhanden sind, um Lese- und Schreibzugriff auf die einzelnen Elemente in einem Array zu bieten.
- Die Zuweisung der Verbindungsparameter der GET/PUT-Anweisungen ist eine Hilfe für den Anwender bei der Konfiguration von S7-Verbindungen, wobei die Eigenschaftsansicht der GET- und PUT-Kommunikationsbausteine verwendet wird, die die Kommunikation von CPU zu CPU unterstützen.
- Mit der Download-in-RUN-Funktion können Sie im Betriebszustand RUN Änderungen am Programm vornehmen, ohne die CPU in STOP versetzen zu müssen.
- PROFINET UDP wird jetzt unterstützt. UDP bietet eine "Broadcast"-Kommunikationsfunktion.
- Die S71200 CPU ist eine PROFINET IO-Steuerung.
- STEP 7 V11 bietet eine Funktion "Rückgängig".
- In STEP 7 gibt es in der Funktionsleiste die Schaltflächen STOP und RUN (Seite 30) zum Stoppen bzw. Starten der CPU.
- Die Forcetabelle (Seite 233) ist unabhängig von der Beobachtungstabelle und ermöglicht das Forcen der Eingänge und Ausgänge.
- Sie können Ihr Anwenderprogramm oder Ihre Codebausteine mit einem Kopierschutz (Seite 88) versehen, indem Sie diese mit einer bestimmten CPU oder Memory Card verknüpfen.
- Sie können die Werte eines DB erfassen (Seite 237), um diese Werte als Startwerte festzulegen.
- Mit einem Klick können Sie die Daten von Tabellen in STEP 7 (z. B. PLC-Variablen Tabellen oder Beobachtungstabellen) in Microsoft Excel exportieren. Zudem können Sie mit den Tastenkombinationen Strg+C und Strg+V zwischen STEP 7 und Microsoft Excel Daten kopieren und einfügen.
- Trennen von Peripheriegeräten (Seite 35) vom konfigurierten Netzwerk ohne Verlust des konfigurierten Geräts und ohne Neukonfiguration des Netzwerks.
- Ändern der Zuweisung eines DB (Seite 34) zu einem FB oder einer Anweisung (z. B. um die Zuweisung eines FB von einem Einzelinstanz-DB zu einem Multiinstanz-DB zu ändern).
- Die Möglichkeit, auf ein einzelnes Bit, Byte oder Wort ("Scheibe") (Seite 68) in einer PLC-Variablen, einer Datenbausteinvariablen oder einer größeren Adresse im Speicher zuzugreifen.
- Die Möglichkeit, einen Datentypparameter mit anderen Typen, einem Array oder Strukturen zu überlagern (Seite 70).
- Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Koreanischen Zertifizierung (KC-Kennzeichen).

Programmierpakete STEP 7 Basic und STEP 7 Professional

STEP 7 bietet zwei Programmierpakete mit den jeweils erforderlichen Funktionen. Beide umfassen die textbasierte, höhere Programmiersprache SCL (Structured Control Language).

- STEP 7 Basic bietet alle Werkzeuge für Ihr S7-1200 Projekt.

Mit dem Software-Paket STEP 7 Basic können Sie Ihre S7-1200 CPUs und die HMI Basic Panels an ein PROFINET-Netzwerk anschließen. Sie können die Verbindung zu anderen Arten von Netzwerken, z. B. PROFIBUS oder RS485, herstellen, indem Sie ein Kommunikationsmodul (CM), einen Kommunikationsprozessor (CP) oder ein Kommunikationsboard (CB) in die Gerätekonfiguration der CPU aufnehmen.

- STEP 7 Professional erweitert die S7-1200 um die Welt der S7-300 und S7-400. Sie können nun Netzwerke mit all diesen SIMATIC-Steuerungen und E/A-Geräten anlegen.

Webserver-Funktionalität

Um über das Internet Zugriff auf die CPU zu bieten, unterstützt die S7-1200 den S7-Webserver mit Standard-Webseiten im CPU-Speicher. Sie können auch eigene Webseiten für den Zugriff auf Daten in der CPU anlegen.

Datenprotokolle

Die S7-1200 unterstützt die Erstellung von Datenprotokolldateien, um Prozesswerte zu speichern. Sie nutzen spezifische DataLog-Anweisungen, um Datenprotokolle anzulegen und zu verwalten. Die Datenprotokolldateien werden in einem CSV-Standardformat gespeichert, das in den meisten Tabellenkalkulationsprogrammen geöffnet werden kann.

Neue Module für die S7-1200

Eine Vielzahl neuer Module erweitern die Leistungsstärke der S7-1200 CPU und bieten Ihnen die erforderliche Flexibilität, um Ihre Automatisierungsanforderungen zu erfüllen:

- Neue E/A-Signalmodule (SMs) einschließlich einem neuen SM 1222 DO8 RLS (Umschaltung)
- Neue Signalboards (SBs) bieten Thermoelement- (TC) und RTD-Funktionen
- Neue Signalboards (SBs) bieten schnelle E/A (200 kHz).
- Neue Kommunikationsmodule (CMs) ermöglichen der S7-1200, als PROFIBUS-Master oder -Slave zu fungieren. Zur Verfügung steht auch ein neues CM 1241 RS422/485.
- Neue Kommunikationsschnittstellen unterstützen die TeleService-Kommunikation (Modem, ISDN, GSM/GPRS und RS232).
- Ein neues Kommunikationsboard (CB) lässt sich in die Front der CPU stecken und bietet RS485-Funktionalität.
- Ein neues AS-i-Master-Kommunikationsmodul, CM 1243-2 AS-i-Master

Siehe auch

Bausteinzustände erfassen und wiederherstellen (Seite 33)

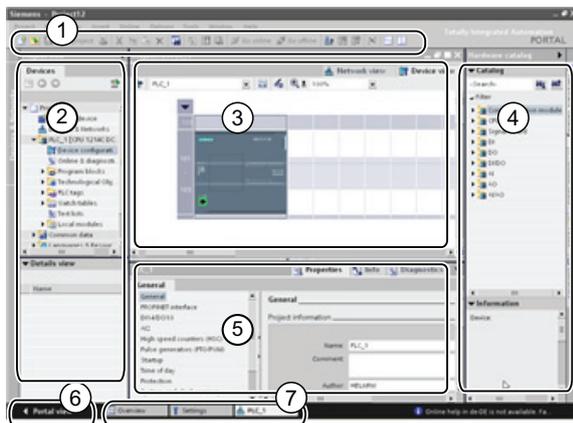
STEP 7 vereinfacht Ihre Arbeit

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

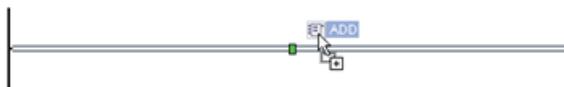
- ① Menü und Funktionsleiste
- ② Projektnavigator
- ③ Arbeitsbereich
- ④ Taskcards
- ⑤ Inspektorfenster
- ⑥ Änderungen an der Portalansicht
- ⑦ Editorleiste

Da sich alle Komponenten an einer Stelle befinden, haben Sie schnellen Zugriff auf jeden Bereich Ihres Projekts. 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 Anwenderprogramm

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.

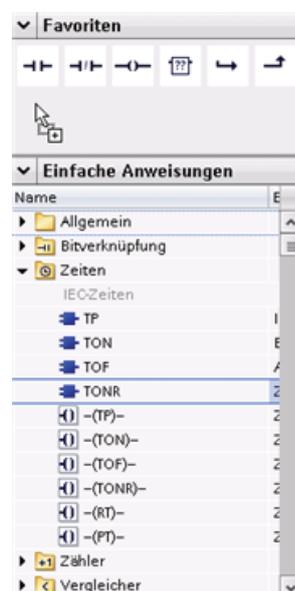
Einfache Anweisungen	
Name	
▶	Allgemein
▶	Bitverknüpfung
▶	Zeiten
▶	Zähler
▶	Vergleicher
▶	Mathematische Funktionen
▶	Verschieben
▶	Umwandler
▶	Programmsteuerung
▶	Wortverknüpfungen
▶	Schieben und Rotieren

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.



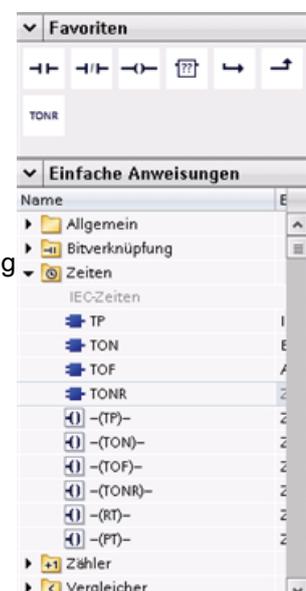
(Um im Anweisungsverzeichnis die "Favoriten" aufzurufen, doppelklicken Sie auf das Symbol.)



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

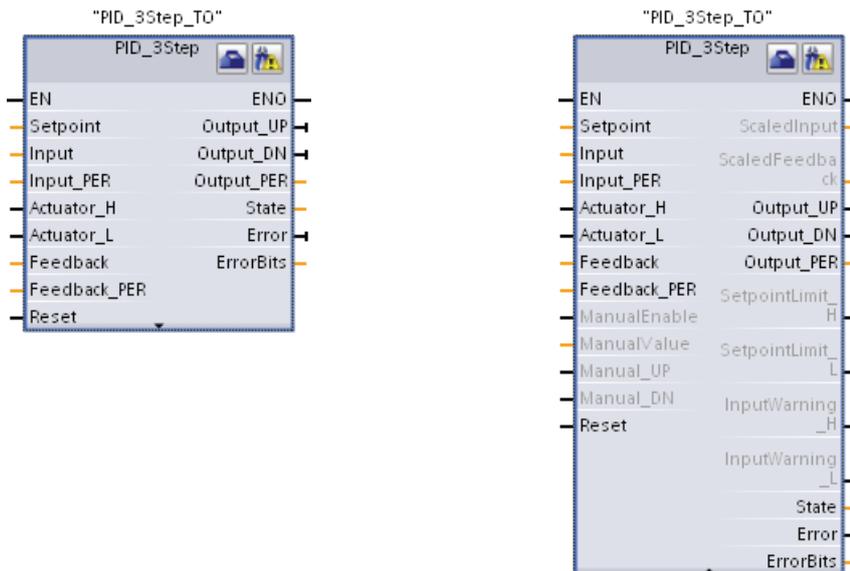


Bei einigen Anweisungen können Sie weitere Eingänge oder Ausgänge erstellen.

- 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 die 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 80).

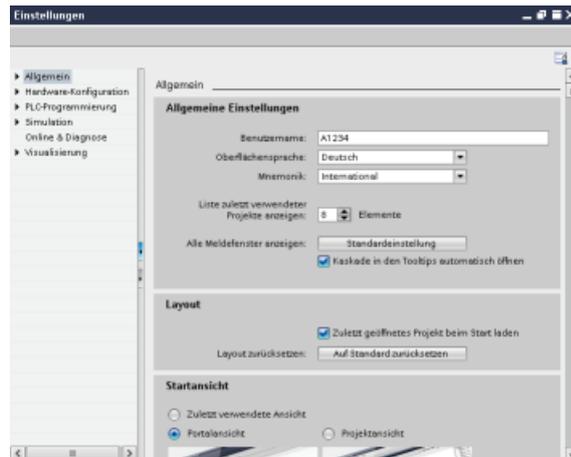
Im Portal "Online & Diagnose" steht Ihnen auch ein Bedienpanel zur Verfügung, in dem Sie den Betriebszustand der Online-CPU ändern können. Um das CPU-Bedienpanel verwenden zu können, müssen Sie mit der CPU online verbunden sein. Das in der Taskcard "Online-Tools" enthaltene Bedienpanel zeigt den Betriebszustand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

2.6 Ändern des Erscheinungsbilds und der Konfiguration von STEP 7

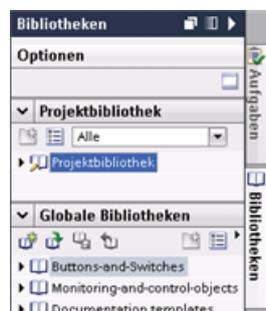


Sie haben zahlreiche Einstellmöglichkeiten; diese betreffen z. B. das Aussehen der Bedienoberfläche, die Sprache oder den Ordner zum Speichern Ihrer Arbeitsergebnisse.

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.

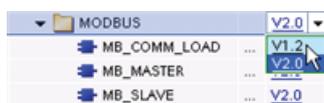
Sie können auch selbst eine globale Bibliothek anlegen, um die Objekte abzulegen, die für andere Projekte verfügbar sein sollen. Beim Anlegen einer neuen globalen Bibliothek speichern Sie diese Bibliothek an einem Speicherort auf Ihrem Computer oder Netzwerk.

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.

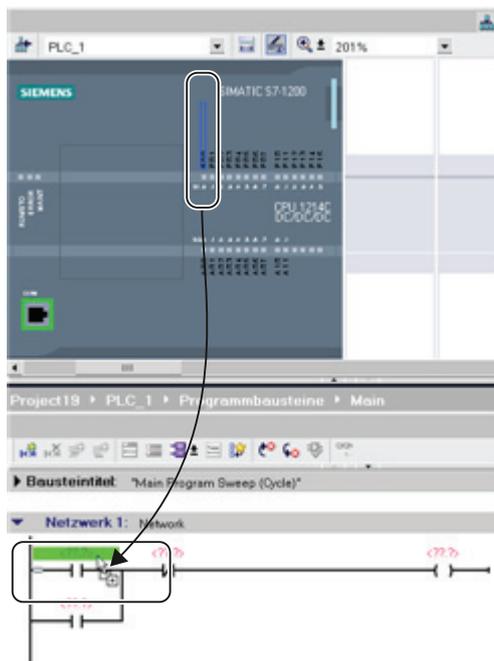


Klicken Sie in der Taskcard mit dem Anweisungsverzeichnis auf das Symbol, um die Überschriften und Spalten im Anweisungsverzeichnis zu aktivieren.



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.

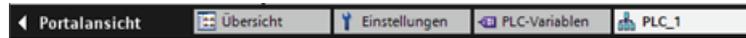
Sie brauchen zum Auswählen der Eingänge oder Ausgänge der CPU mindestens den Zoomfaktor 200 %.

Beachten Sie, dass die Variablennamen 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 Bausteinzustände erfassen und wiederherstellen

STEP 7 bietet die Möglichkeit, den Zustand eines Codebausteins zu erfassen, um einen Bezugs- oder Referenzpunkt für das Anwenderprogramm zu erstellen. Ein Bausteinzustand stellt den Status eines Codebausteins zu einem bestimmten Zeitpunkt dar. Wenn Sie einen Bausteinzustand erfassen, können Sie den Baustein jederzeit auf diesen Zustand zurücksetzen und alle danach vorgenommenen Änderungen verwerfen. Sie können im Anwenderprogramm den Zustand des Bausteins wiederherstellen, auch wenn Sie am Programm Änderungen vorgenommen und gespeichert haben.

Sie können bis zu 10 Bausteinzustände in Ihrem Projekt erfassen. Die Bausteinzustände sind auch nach dem Speichern des Projekts noch abrufbar. Wenn Sie das Projekt jedoch schließen, werden die erfassten Bausteinzustände gelöscht.

Die Funktion zum Erfassen und Wiederherstellen des Zustands des Programmbausteins leistet mehr als die Funktion "Rückgängig", weil der Bausteinzustand auch nach dem Speichern noch abrufbar ist.



Um den aktuellen Zustand des Anwenderprogramms zu speichern, klicken Sie auf die Schaltfläche "Bausteinzustand erfassen". Nachdem Sie einen Zustand des Anwenderprogramms erfasst haben, zeigt der Programmbaustein das Symbol "Bausteinzustand" an.

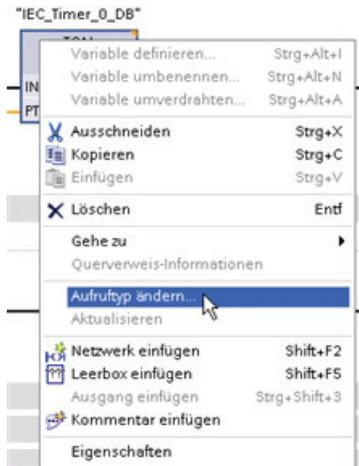


Um für den Programmbaustein den erfassten Bausteinzustand wiederherzustellen, klicken Sie auf die Schaltfläche "Bausteinzustand wiederherstellen".



Um den erfassten Bausteinzustand zu löschen, klicken Sie auf die Schaltfläche "Bausteinzustand löschen".

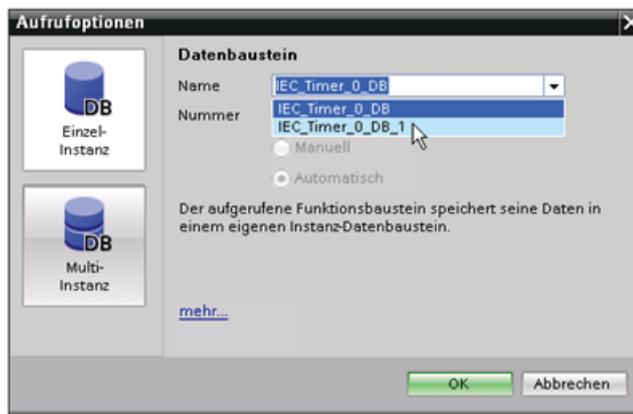
2.11 Aufrufotyp eines DB ändern



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).

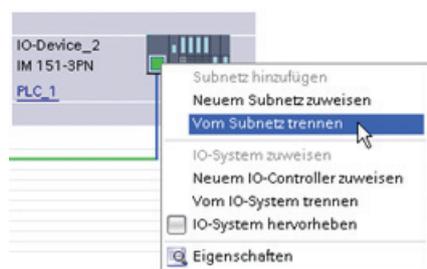
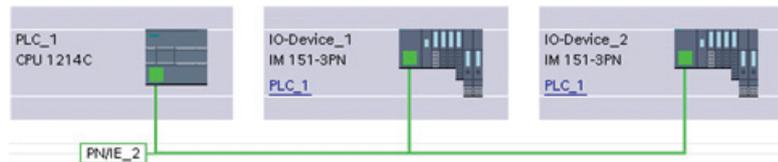
Sie können den Befehl "Aufrufotyp ändern" auswählen, indem Sie entweder im Programmiereditor mit der rechten Maustaste auf die Anweisung oder den FB klicken oder indem Sie im Menü "Optionen" den Befehl "Bausteinanruf" auswählen.



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

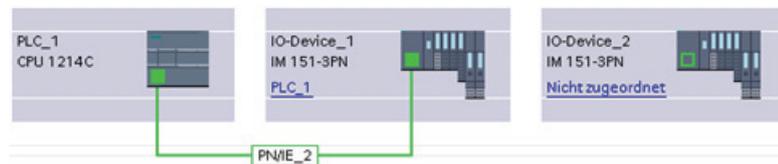
2.12 Geräte vorübergehend vom Netzwerk trennen

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".

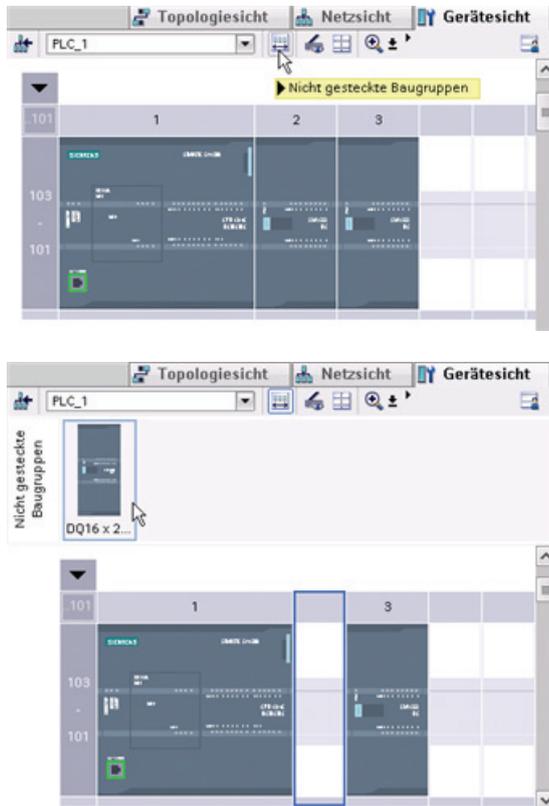
STEP 7 konfiguriert die Netzwerkverbindungen neu, entfernt das getrennte Gerät jedoch nicht aus dem Projekt. Die Netzwerkverbindung wird zwar gelöscht, doch die Schnittstellenadressen werden nicht verändert.



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

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

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



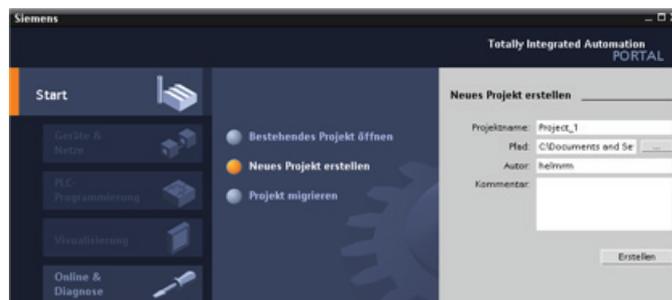
STEP 7 bietet Ihnen eine virtuelle Ablage für "nicht gesteckte" Baugruppen. Sie können eine Baugruppe vom Baugruppenträger "abziehen" und dabei die Konfiguration der Baugruppe speichern. Diese abgezogenen Baugruppen werden mit dem Projekt zusammen gespeichert; dadurch können Sie sie später wieder einsetzen,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



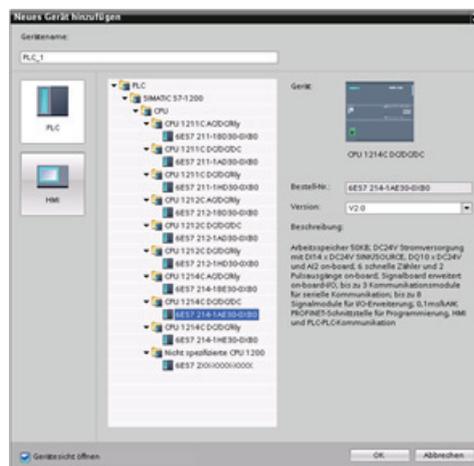
Klicken Sie im Portal "Start" auf die Aufgabe "Neues Projekt erstellen".

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



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

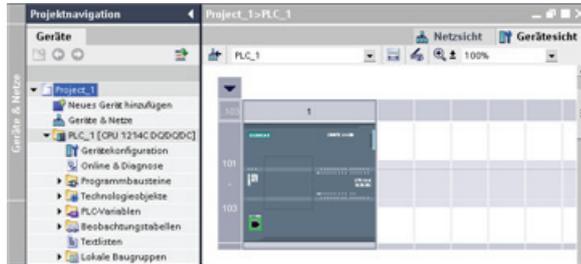
Klicken Sie auf "Neues Gerät hinzufügen".



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".

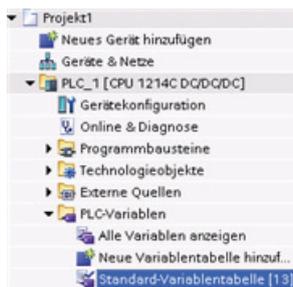
Beachten Sie, dass die Option "Gerätesicht öffnen" ausgewählt ist. Bei Auswahl dieser Option öffnen Sie durch Klicken auf "Hinzufügen" die "Gerätekonfiguration" der Projektansicht.



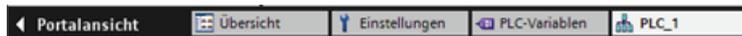
In der Gerätesicht wird die hinzugefügte CPU angezeigt.

3.2 Variablen für die E/A der CPU anlegen

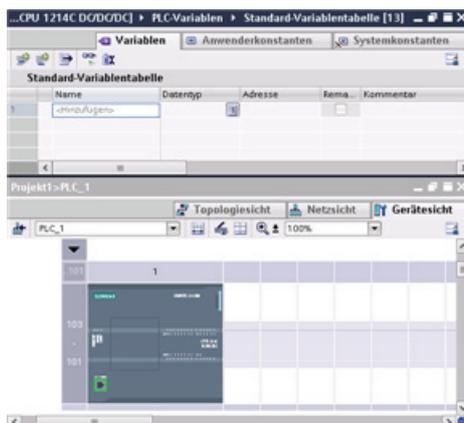
"PLC-Variablen" sind die symbolischen Namen für E/A und Adressen. Wenn Sie eine PLC-Variable anlegen, speichert STEP 7 die Variable in einer Variablen-tabelle. Der Zugriff auf die Variablen-tabelle kann über alle Editoren erfolgen (Programmieditor, Geräteeditor, Visualisierungseditor und Beobachtungstabelleneditor).



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



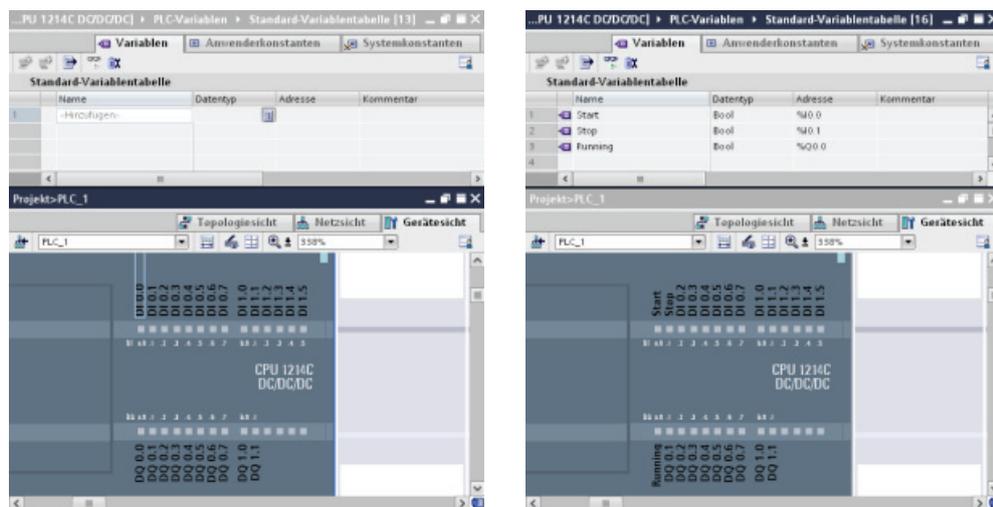
Klicken Sie in der Funktionsleiste auf die Schaltfläche zum horizontalen Teilen des Editorbereichs.



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:

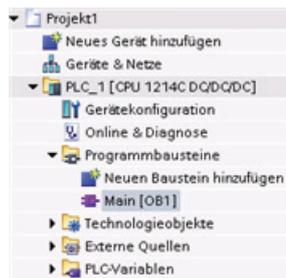
1. Wählen Sie "E0.0" und ziehen den Eingang in die erste Zeile der Variablen-tabelle.
2. Ändern Sie den Variablennamen von "E0.0" in "Start".
3. Ziehen Sie E0.1 in die Variablen-tabelle und ändern Sie den Namen in "Stop".
4. Ziehen Sie A0.0 (im unteren Bereich der CPU) in die Variablen-tabelle und ändern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

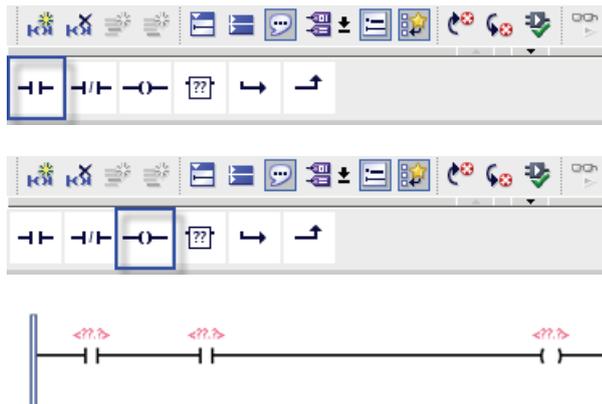
Ihr Programmcode besteht aus Anweisungen, die von der CPU der Reihe nach ausgeführt werden. Legen Sie in diesem Beispiel den Programmcode im Kontaktplan (KOP) an. Das KOP-Programm besteht aus einer Folge von Netzwerken, die den Strompfaden eines Schaltplans ähneln.



Um den Programmiereditor zu öffnen, gehen Sie folgendermaßen vor:

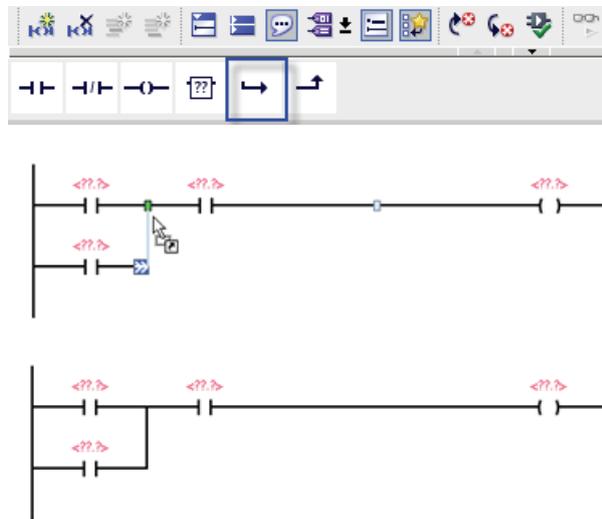
1. Erweitern Sie den Ordner "Programmbausteine" in der Projektnavigation zur Anzeige des Bausteins "Main [OB1]".
 2. Doppelklicken Sie auf den Baustein "Main [OB1]".
- Der Programmiereditor öffnet den Programmbaustein (OB1).

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.

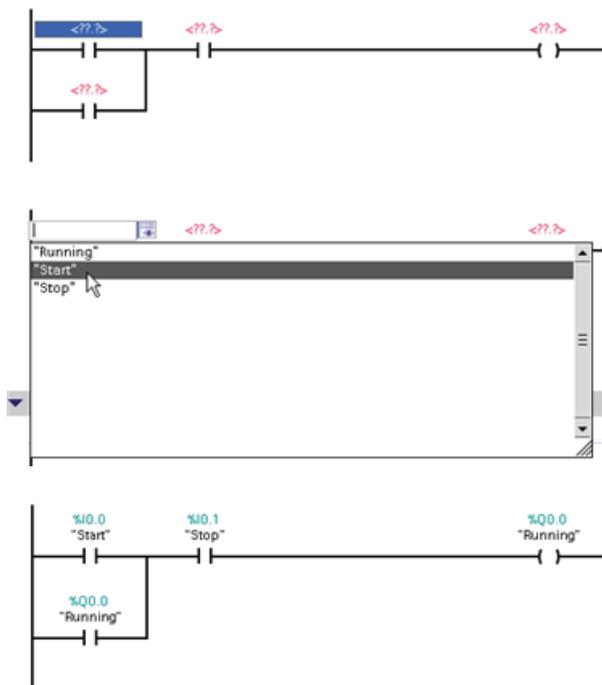


1. Wählen Sie die linke Schiene, um die Schiene für die Verzweigung auszuwählen.
2. Um eine Verzweigung zur Schiene des Netzwerks hinzuzufügen, klicken Sie auf das Symbol für "Verzweigung öffnen".
3. Fügen Sie einen weiteren Schließer in die geöffnete Verzweigung ein
4. Ziehen Sie den Pfeil mit zwei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

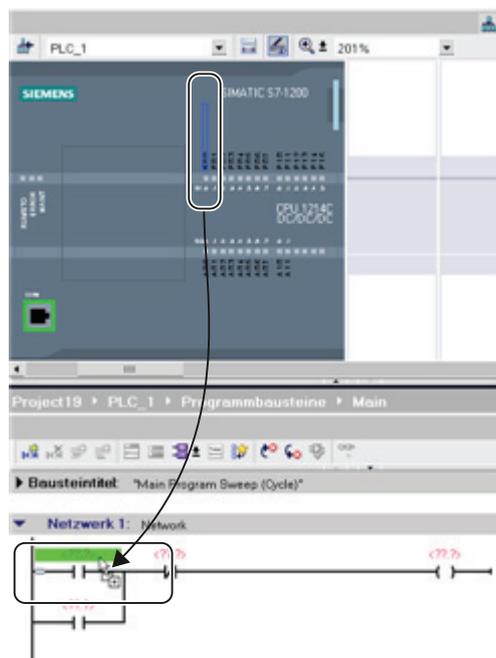
Um das Projekt zu speichern, klicken Sie in der Funktionsleiste auf "Projekt speichern". Sie können auch dann speichern, wenn die Bearbeitung des Strompfads noch nicht abgeschlossen ist. Sie können jetzt den Variablennamen die entsprechenden Anweisungen zuordnen.

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".



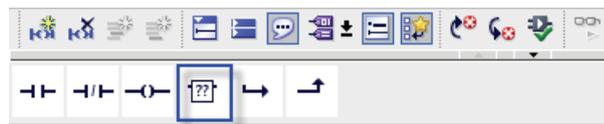
Sie können die E/A-Adressen auch direkt von der CPU ziehen. Dazu müssen Sie nur den Arbeitsbereich der Projektansicht teilen (Seite 32).

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

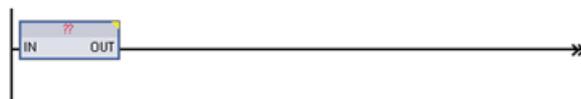
So können Sie die E/A der CPU aus der "Gerätekonfiguration" in die KOP-Anweisung im Programmiereditor ziehen und damit nicht nur die Adresse für die Anweisung, sondern gleichzeitig einen Eintrag in der PLC-Variablen-tabelle anlegen.

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.



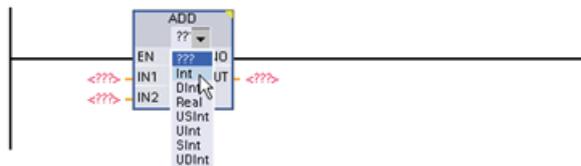
Klicken Sie in der Funktionsleiste "Favoriten" auf die allgemeine "Box"-Anweisung.



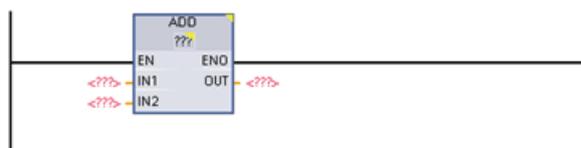
Unter der allgemeinen "Box"-Anweisung sind eine Reihe von Anweisungen verfügbar. Für dieses Beispiel erstellen Sie eine Anweisung ADD:



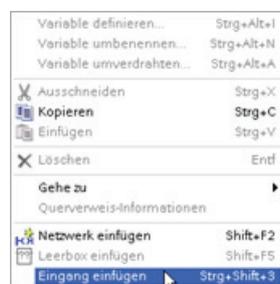
1. Öffnen Sie die Klappliste mit Anweisungen, indem Sie auf die gelbe Ecke der "Box"-Anweisung klicken.
2. Blättern Sie in der Liste nach unten und wählen Sie die Anweisung ADD aus.



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:

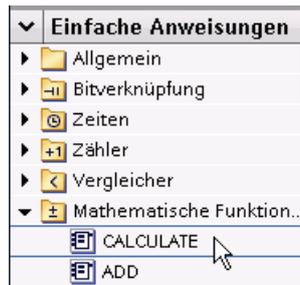
1. Klicken Sie auf einen der Eingänge in der Box.
2. Klicken Sie mit der rechten Maustaste, um das Kontextmenü aufzurufen, und wählen Sie "Eingang einfügen".



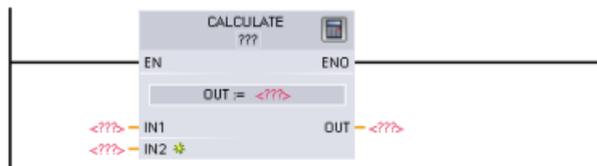
Die Anweisung ADD verwendet nun drei Eingänge.

3.6 Anweisung CALCULATE für komplexe mathematische Gleichungen verwenden

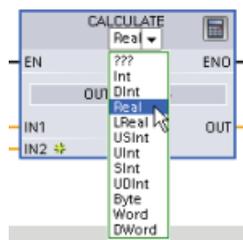
Mit der Anweisung Calculate können Sie eine mathematische Funktion erstellen, die mehrere Eingangsparameter verarbeitet und das Ergebnis entsprechend der von Ihnen vorgegebenen Gleichung ausgibt.



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.)

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

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

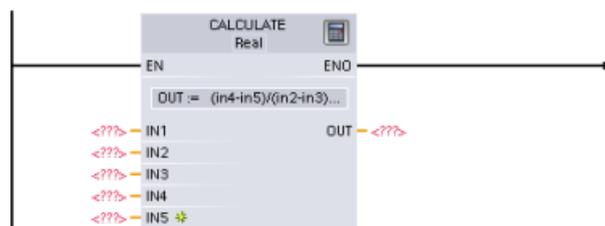
Erläuterung:	Out _{value}	(Out)	Skalierter Ausgangswert
	In _{value}	(in1)	Analogeingangswert
	In _{high}	(in2)	Oberer Grenzwert für den skalierten Eingangswert
	In _{low}	(in3)	Unterer Grenzwert für den skalierten Eingangswert
	Out _{high}	(in4)	Oberer Grenzwert für den skalierten Ausgangswert
	Out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

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

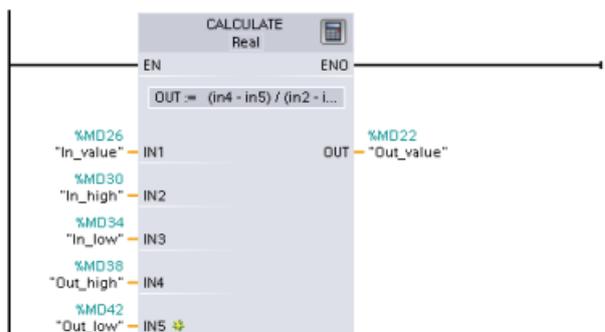
$$\text{OUT} = ((\text{in4} - \text{in5}) / (\text{in2} - \text{in3})) * (\text{in1} - \text{in3}) + \text{in5}$$



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

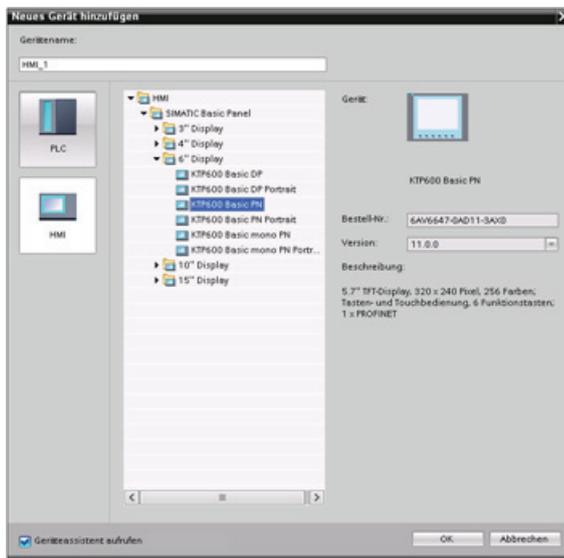


Geben Sie die Variablennamen für die Werte ein, die den Parametern entsprechen.



3.7 HMI-Gerät zum Projekt hinzufügen

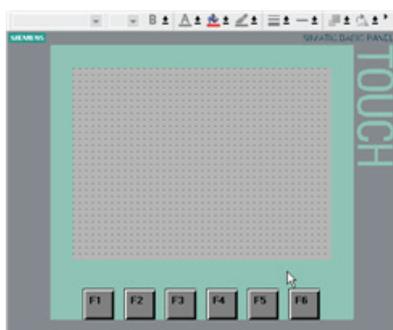
Das Hinzufügen eines HMI-Geräts zum Projekt ist einfach!



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.

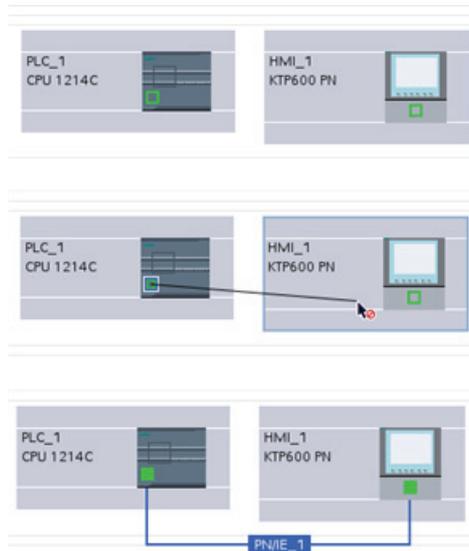
4. Klicken Sie auf "OK", um das HMI-Gerät zum Projekt hinzuzufügen.



Das HMI-Gerät wird zum Projekt hinzugefügt. STEP 7 bietet Ihnen einen HMI-Assistenten zur leichteren Konfiguration aller Bilder und der Struktur Ihres HMI-Geräts.

Wenn Sie den Geräteassistenten nicht ausführen, legt STEP 7 ein einfaches HMI-Standardbild an.

3.8 Netzwerkverbindung zwischen CPU und HMI-Gerät herstellen

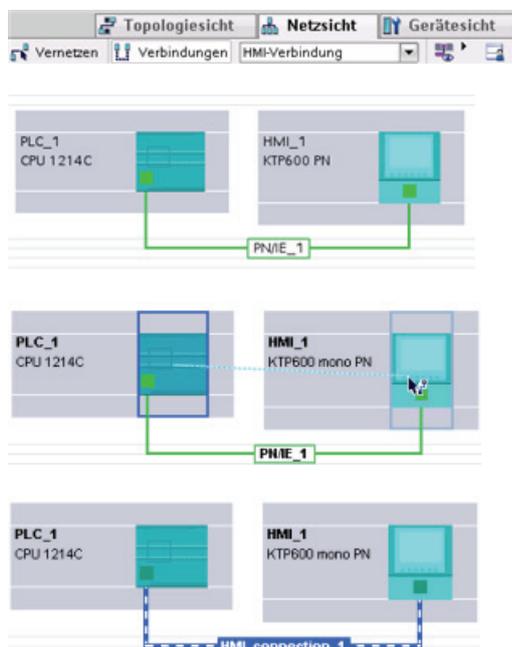


Das Erstellen einer Netzwerkverbindung ist einfach!

- Navigieren Sie zu "Geräte & Netze" und wählen Sie die Netzansicht, 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.

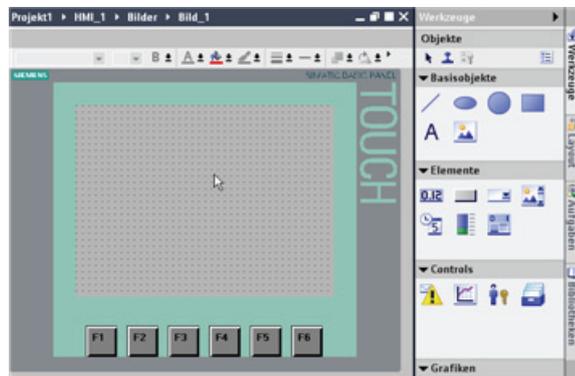
- Klicken Sie bei ausgewählter Netzwerkverbindung auf die Schaltfläche "Verbindungen" und wählen Sie dann in der Klappliste "HMI-Verbindung" aus.
- Hierdurch erscheinen die beiden Geräte in blauer Anzeigefarbe.
- Wählen Sie die CPU aus und ziehen Sie die Linie zum HMI-Gerät.
- Das Konfigurieren der Variablen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

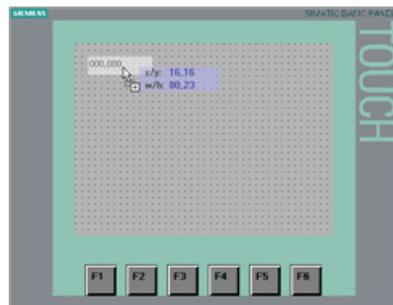
3.10 HMI-Bild anlegen

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.

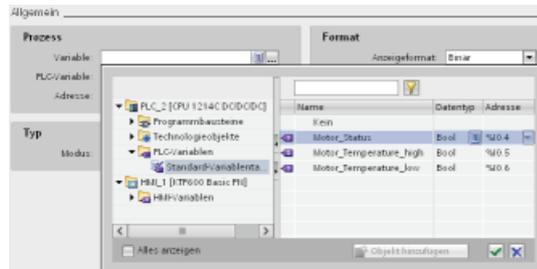
Um ein Element einzufügen, brauchen Sie das Element nur auf das Bild zu ziehen und dort abzulegen. Das Konfigurieren von Aussehen und Verhalten des Elements erfolgt über die Elementeigenschaften im Inspektorenfenster.



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

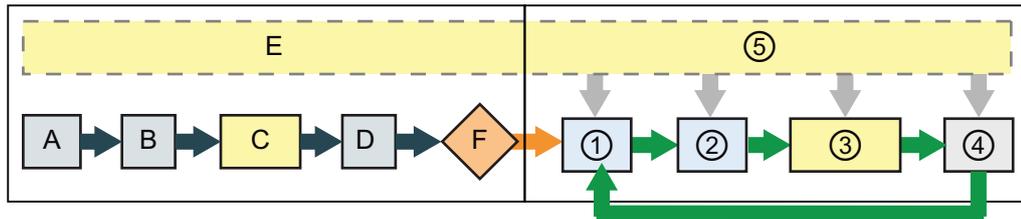
Jeder Zyklus umfasst das Schreiben der Ausgänge, das Lesen der Eingänge, das Bearbeiten der Anweisungen des Anwenderprogramms und die Durchführung der Systemwartung oder Hintergrundverarbeitung.



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.

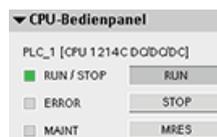
4.2 Betriebszustände der CPU

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.
- Im Betriebszustand STARTUP führt die CPU eine Anlauflogik (sofern vorhanden) aus. Alarmereignisse werden im Betriebszustand STARTUP nicht bearbeitet.
- Im Betriebszustand RUN wird der Zyklus wiederholt ausgeführt. Alarmereignisse können auftreten und an beliebigen Punkten innerhalb der Programmzyklusphase abgearbeitet werden. Einige Teile eines Projekts können im Betriebszustand RUN in die CPU geladen werden.

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 die Programmiersoftware auslösen. Beim Umräumen werden der Arbeitsspeicher sowie alle remanenten und nicht remanenten Speicherbereiche gelöscht und der Ladespeicher in den Arbeitsspeicher kopiert. Der Diagnosepuffer und die dauerhaft gespeicherten IP-Adressen werden beim Umräumen nicht gelöscht. Während eines Warmstarts werden alle nicht remanenten System- und Anwenderdaten initialisiert.

Die Einstellung für das Anlaufverhalten der CPU nach NETZ-EIN kann einschließlich der Anlaufart vollständig über die Programmiersoftware 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".



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 Bausteinarten für den Aufbau einer geeigneten Struktur Ihres Anwenderprogramms:

- Organisationsbausteine (OBs) legen die Struktur des Programms fest. Für einige OBs gibt es vordefiniertes Verhalten und Startereignisse, Sie können aber auch OBs mit eigenen Startereignissen (Seite 56) 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. Gültige FC- und FB-Nummern liegen im Bereich von 1 bis 65.535.
- Datenbausteine (DBs) speichern Daten, die von den Programmbausteinen verwendet werden können. Gültige DB-Nummern liegen im Bereich von 1 bis 65.535.

Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU (Seite 13) 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 bearbeitet.

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. Jedem OB muss eine eindeutige Nummer zugeordnet sein. Die Nummern unter 200 sind für bestimmte OBs belegt. Alle anderen OBs müssen eine Nummer über 200 besitzen.

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 Startereignis, z. B. ein Diagnosealarm oder ein Zeitintervall, kann die Ausführung eines OBs auslösen. 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 27 (für Zeitfehler).

OBs steuern die folgenden Abläufe:

- Programmzyklus-OBs werden zyklisch ausgeführt, wenn die CPU im Betriebszustand RUN ist. Der Hauptbaustein des Programms ist ein Programmzyklus-OB. Er enthält die Anweisungen für die Steuerung Ihrer Anwendung und aus ihm heraus werden weitere Anwenderbausteine aufgerufen. Mehrere Programmzyklus-OBs sind zulässig, sie werden in numerischer Reihenfolge ausgeführt. OB 1 ist der Standardbaustein. Andere Programmzyklus-OBs müssen als OB 200 oder höher gekennzeichnet werden.
- Anlauf-OBs werden einmal ausgeführt, wenn der Betriebszustand der CPU von STOP nach RUN wechselt, beim Hochfahren in den Betriebszustand RUN und bei einem vorgegebenen Wechsel von STOP nach RUN. Anschließend beginnt die Ausführung des Zyklus-OBs. Es sind mehrere Anlauf-OBs zulässig. OB 100 ist der Standardbaustein. Alle anderen OBs müssen Nummern ab 200 haben.
- Weckalarm-OBs werden in bestimmten Abständen ausgeführt. Ein Weckalarm-OB unterbricht den Programmablauf in bestimmten, benutzerdefinierten Abständen, z. B. alle 2 Sekunden. Es können insgesamt bis zu 4 Verzögerungsereignisse und zyklische Ereignisse zu beliebigen Zeitpunkten konfiguriert werden, wobei für jedes konfigurierte Verzögerungsereignis oder zyklische Ereignis ein OB zulässig ist. Der OB muss die Nummer 200 oder höher haben.
- Prozessalarm-OBs werden ausgeführt, sobald das entsprechende Prozessereignis auftritt, z.B. steigende oder fallende Flanke an einem integrierten digitalen Eingang oder ein HSC-Ereignis. Ein Prozessalarm-OB unterbricht den normalen Programmablauf durch ein Signal eines Prozessereignisses. Sie definieren die Ereignisse in den Eigenschaften der Hardwarekonfiguration. Für jeden Prozessalarm ist ein OB zulässig. Der OB muss die Nummer 200 oder höher haben.

- Ein Zeitfehler-OB wird ausgeführt, wenn entweder die maximale Zykluszeit überschritten wird oder ein Zeitfehlerereignis auftritt. Die Verarbeitung von Zeitfehleralarmen wird von OB 80 durchgeführt. Wird dieser OB ausgelöst, beginnt die Ausführung, die den normalen Programmablauf oder auch einen anderen Ereignis-OB unterbricht. Die Ereignisse, die den Zeitfehleralarm und die Reaktion der CPU auf diese Ereignisse auslösen, werden im Folgenden beschrieben:
 - Überschreiten der maximalen Zykluszeit: Sie können die maximale Zykluszeit in den Eigenschaften der CPU konfigurieren. Wenn OB 80 nicht vorhanden ist, reagiert die CPU auf das Überschreiten der maximalen Zykluszeit, indem sie in STOP geht.
 - Zeitfehler: Wenn OB 80 nicht vorhanden ist, reagiert die CPU, indem sie in RUN bleibt. Zeitfehler treten auf, wenn ein Uhrzeitereignis verpasst oder wiederholt wird, wenn die Warteschlange überläuft oder ein Ereignis-OB (Zeitverzögerungsereignis, Uhrzeitereignis oder Weckalarm) startet, bevor die CPU die Ausführung des vorherigen OBs beendet hat.

Das Auftreten eines dieser Ereignisse erzeugt einen Eintrag im Diagnosepuffer, der das Ereignis beschreibt. Der Eintrag im Diagnosepuffer wird unabhängig davon erzeugt, ob OB 80 vorhanden ist oder nicht.

- Diagnosefehler-OBs werden ausgeführt, wenn ein Diagnosefehler erkannt und gemeldet wird. Ein Diagnose-OB unterbricht den normalen Programmablauf, wenn eine diagnosefähige Baugruppe einen Fehler erkennt (sofern für die Baugruppe der Diagnosealarm aktiviert wurde). Für Diagnosealarme ist ausschließlich OB 82 zulässig. Sie können eine Anweisung STP (CPU in STOP versetzen) in Ihren OB 82 einfügen, wenn Sie möchten, dass Ihre CPU bei einem Fehler dieser Art in den Betriebszustand STOP wechselt. Umfasst das Programm keinen Diagnose-OB, ignoriert die CPU den Fehler (und bleibt in RUN).

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 Zeitverzögerungsereignisse treten einmalig auf. Einige Ereignisse treten auf, wenn es zu einer von der Hardware ausgelösten Veränderung kommt, z.B. ein Flankenereignis an einem Eingang oder ein Ereignis eines schnellen Zählers. Außerdem gibt es Ereignisse wie das Diagnosefehler- und das Zeitfehlerereignis, die nur im Fehlerfall auftreten. Die Ereignisprioritäten und Warteschlangen dienen zum Festlegen der Verarbeitungsreihenfolge der Alarm-OBs.

Das Programmzyklusereignis tritt einmal in jedem Programmzyklus auf. Während des Programmzyklus schreibt die CPU in die Ausgänge, liest die Eingänge und führt Programmzyklus-OBs aus. Das Programmzyklusereignis ist erforderlich und immer aktiviert. Für das Programmzyklusereignis haben Sie möglicherweise keinen Programmzyklus-OB oder Sie haben möglicherweise mehrere OBs. Nachdem das Programmzyklusereignis ausgelöst wurde, wird der Programmzyklus-OB mit der kleinsten Nummer ausgeführt (in der Regel OB 1). Die anderen Programmzyklus-OBs werden sequentiell (in numerischer Reihenfolge) innerhalb des Programmzyklus ausgeführt.

Die Weckalarmereignisse ermöglichen Ihnen, die Ausführung eines Alarm-OBs zu einer konfigurierten Zykluszeit einzurichten. Die anfängliche Zykluszeit wird konfiguriert, wenn der OB angelegt und als Weckalarm-OB eingerichtet wird. Ein zyklisches Ereignis unterbricht den Programmzyklus und führt den Weckalarm-OB aus (das zyklische Ereignis befindet sich in einer Klasse mit höherer Priorität als das Programmzyklusereignis).

Einem zyklischen Ereignis darf nur ein Weckalarm-OB zugeordnet werden.

Jedem zyklischen Ereignis kann eine Phasenverschiebung zugewiesen werden, so dass die Ausführung von Weckalarmen mit derselben Zykluszeit um den Wert der Phasenverschiebung verschoben werden kann. Der voreingestellte Wert der Phasenverschiebung ist 0. Um die anfängliche Phasenverschiebung zu ändern oder um die anfängliche Zykluszeit eines zyklischen Ereignisses zu ändern, klicken Sie in der Projektnavigation mit der rechten Maustaste auf den Weckalarm-OB, dann auf "Eigenschaften" und auf "Weckalarm". Geben Sie hier die neuen anfänglichen Werte ein. Sie können die Zykluszeit und Phasenverschiebung auch aus Ihrem Programm abfragen und ändern. Verwenden Sie dazu die Anweisungen Weckalarm-Parameter abfragen (QRY_CINT) und Weckalarm-Parameter setzen (SET_CINT). Die über die Anweisung SET_CINT eingegebenen Werte für Zykluszeit und Phasenverschiebung werden bei Ausschalten oder Wechsel in STOP nicht gespeichert. Bei Wiedereinschalten oder Rückkehr in RUN werden wieder die ursprünglichen anfänglichen Werte verwendet. Die CPU unterstützt insgesamt vier Weckalarm- und Verzögerungsereignisse.

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

Die Zeitverzögerungsereignisse ermöglichen Ihnen, die Ausführung eines Alarm-OB nach Ablauf einer vorgegebenen Zeitverzögerung einzurichten. Die Verzögerungszeit wird mit der Anweisung SRT_DINT angegeben. Die Zeitverzögerungsereignisse unterbrechen den Programmzyklus, um den Zeitverzögerungs-OB auszuführen. Einem Zeitverzögerungsereignis darf nur ein Zeitverzögerungs-OB zugeordnet werden. Die CPU unterstützt vier Zeitverzögerungsereignisse.

Die Prozessalarmereignisse werden durch eine Veränderung in der Hardware ausgelöst, z.B. eine steigende oder fallende Flanke an einem Eingang oder ein HSC-Ereignis (schneller Zähler). Für jedes Prozessalarmereignis kann ein Alarm-OB ausgewählt werden. Die Hardware-Ereignisse werden in der Gerätekonfiguration aktiviert. Die OBs für das Ereignis werden in der Gerätekonfiguration oder mit einer Anweisung ATTACH im Anwenderprogramm angegeben. Die CPU unterstützt mehrere Prozessalarmereignisse. Die genaue Anzahl der Ereignisse richtet sich nach der Variante der CPU und der Anzahl der Eingänge.

Die Zeit- und Diagnosefehlerereignisse werden ausgelöst, wenn die CPU einen Fehler erkennt. Diese Ereignisse sind in einer Klasse mit höherer Priorität als die anderen Alarmereignisse und können die Ausführung der Zeitverzögerungs-, Weckalarm- und Prozessalarmereignisse unterbrechen. Für jedes Zeitfehler- und Diagnosefehlerereignis kann ein Alarm-OB angegeben werden.

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 folgenden Abschnitt "Wissenswertes zu Zeitfehlerereignissen".

Jeder Ereignis einer CPU hat eine Priorität. Sie können die Priorität eines OBs nicht ändern. Die Ereignisse werden im Allgemeinen in der Reihenfolge ihrer Priorität (höchste zuerst) bearbeitet. Ereignisse mit gleicher Priorität werden nach dem First-In-First-Out-Prinzip bearbeitet.

Tabelle 4- 1 OB-Ereignisse

Ereignis	OB-Nummer	Zulässige Anzahl	Startereignis	OB-Priorität
Programmzyklus	OB 1, OB 200 bis OB 65535	1 Programmzyklusereignis Mehrere OBs zulässig	<ul style="list-style-type: none"> Anlauf-OB endet Letzter Programmzyklus-OB endet 	1
Anlauf	OB 100, OB 200 bis OB 65535	1 Anlaufereignis ^{1,2} Mehrere OBs zulässig	Wechsel von STOP nach RUN	1
Zeit	OB 200 bis OB 65535	Bis zu 4 Zeitereignisse ³ 1 OB je Ereignis	Verzögerungs-OB-Ereignis ist geplant	3
			Zyklus-OB-Ereignis ist geplant	4
Prozess	OB 200 bis OB 65535	Bis zu 50 Prozessereignisse ⁴ 1 OB je Ereignis	Flanken: <ul style="list-style-type: none"> Ereignisse steigende Flanke: max. 16 Ereignisse fallende Flanke: max. 16 	5
			Bei HSC: <ul style="list-style-type: none"> CV=PV: max. 6 Richtungswechsel: max. 6 Externes Rücksetzen: max. 6 	6

Ereignis	OB-Nummer	Zulässige Anzahl	Startereignis	OB-Priorität
Diagnosefehler	OB 82	1 Ereignis (nur, wenn OB 82 geladen war)	Modul sendet einen Fehler	9
Zeitfehler	OB 80	1 Ereignis (nur, wenn OB 80 geladen war) ⁵	<ul style="list-style-type: none"> • Maximale Zykluszeit wurde überschritten • Ein zweiter Alarm (Weck- oder Verzögerungsalarm) wurde gestartet, bevor die CPU die Ausführung des ersten Alarms beendet hat 	26

- ¹ Das Anlauf- und das Programmzyklus-Ereignis treten nie gleichzeitig ein, weil der Anlauf zuerst beendet sein muss, bevor der Programmzyklus gestartet wird (Steuerung durch das Betriebssystem).
- ² Nur das Diagnosefehlerereignis (OB 82) unterbricht das Anlaufereignis. Alle anderen Ereignisse werden für die Bearbeitung nach dem Anlaufereignis in die Warteschlange gestellt.
- ³ Die CPU stellt insgesamt 4 Zeitereignisse zur Verfügung, die gemeinsam von den Verzögerungs-OBs und den Zyklus-OBs genutzt werden. Die Anzahl der Verzögerungs- und Zyklus-OBs im Anwenderprogramm darf nicht größer als 4 sein.
- ⁴ Bei Verwendung der Anweisungen DETACH und ATTACH sind mehr als 50 Prozessereignisse möglich.
- ⁵ Sie können die CPU so konfigurieren, dass sie in RUN bleibt, wenn die maximale Zykluszeit überschritten wurde, 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.

Nachdem die Ausführung eines OBs mit einer Priorität von 2 bis 25 gestartet ist, kann die Bearbeitung dieses OBs nicht durch Auftreten eines weiteren Ereignisses unterbrochen werden. Ausnahme ist OB 80 (Zeitfehlerereignis mit der Priorität 26). Alle anderen Ereignisse werden für die spätere Bearbeitung in die Warteschlange gestellt, damit der aktuelle OB beendet werden kann.

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.

Wissenswertes zu Zeitfehlerereignissen

Das Auftreten eines von verschiedenen Zeitfehlern führt zu einem Zeitfehlerereignis. Die folgenden Zeitfehler werden unterstützt:

- Maximale Zykluszeit überschritten
- Gewünschter OB kann nicht gestartet werden
- Warteschlangenüberlauf

Der Fehler "Maximale Zykluszeit überschritten" tritt auf, wenn der Programmzyklus nicht innerhalb der angegebenen maximalen Zykluszeit beendet wird. Weitere Informationen zum Fehler "Maximale Zykluszeit überschritten", zum Konfigurieren der maximalen Zykluszeit und zum Zurücksetzen der Zykluszeit finden Sie im Abschnitt "Überwachen der Zykluszeit" (Seite 231).

Der Fehler "Gewünschter OB kann nicht gestartet werden" tritt auf, wenn ein OB von einem Weckalarm, einem Verzögerungsalarm oder einem Uhrzeitalarm angefordert wird, doch bereits ausgeführt wird.

Der Fehler "Warteschlangenüberlauf" tritt auf, wenn die Alarmer schneller auftreten als sie verarbeitet werden können. Die Zahl anstehender Ereignisse kann begrenzt werden, indem jedem Ereignistyp eine eigene Warteschlange zugewiesen wird. Tritt ein Ereignis auf, wenn die entsprechende Warteschlange voll ist, wird ein Zeitfehlerereignis erzeugt.

Alle Zeitfehlerereignisse lösen die Ausführung von OB 80 (sofern vorhanden) aus. Wenn das Anwenderprogramm keinen OB 80 enthält, legt die Gerätekonfiguration der CPU die Reaktion der CPU auf den Zeitfehler fest:

- Bei der Standardkonfiguration für Zeitfehler, wenn z. B. ein zweiter Weckalarm gestartet wird, bevor die CPU die Ausführung des ersten beendet hat, bleibt die CPU in RUN.
- Bei der Standardkonfiguration für das Überschreiten der maximalen Zykluszeit wird die CPU in STOP versetzt.

Sie können die maximale Zykluszeit mit der Anweisung RE_TRIGR zurücksetzen. Wenn jedoch der Fehler "Maximale Zykluszeit überschritten" zwei Mal in demselben Programmzyklus auftritt, ohne dass die Zykluszeit zurückgesetzt wird, geht die CPU in STOP, unabhängig davon, ob OB 80 vorhanden ist. Siehe hierzu den Abschnitt "Überwachen der Zykluszeit" (Seite 231).

OB 80 enthält Anlaufinformationen, anhand deren Sie ermitteln können, welches Ereignis und welcher OB den Zeitfehler erzeugt hat. Sie können in OB 80 Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Tabelle 4-2 Anlaufinformationen für OB 80

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

Wenn Sie ein neues Projekt anlegen, ist kein Zeitfehler-OB 80 vorhanden. Wenn Sie möchten, fügen Sie einen Zeitfehler-OB 80 in Ihr Projekt ein. Hierfür doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", dann wählen Sie "Organisationsbaustein" und "Zeitfehler".

Wissenswertes zu Diagnosefehlerereignissen

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

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

Diagnosefehlerereignisse lösen die Ausführung von OB 82 (sofern vorhanden) aus. Ist OB 82 nicht vorhanden, ignoriert die CPU den Fehler. Wenn Sie ein neues Projekt anlegen, ist kein Diagnosefehler-OB 82 vorhanden. Wenn Sie möchten, fügen Sie einen Diagnosefehler-OB 82 in Ihr Projekt ein. Hierfür doppelklicken Sie in der Projektnavigation unter "Programmbausteine" auf "Neuen Baustein hinzufügen", dann wählen Sie "Organisationsbaustein" und "Diagnosefehler".

Hinweis

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

Der Diagnosefehleralarm von OB 82 kann nur die Diagnosefehler jeweils eines Kanals melden.

Wenn in zwei Kanälen eines mehrkanaligen Geräts Fehler auftreten, löst der zweite Fehler den OB 82 nur unter folgenden Bedingungen aus: Der Fehler des ersten Kanals wird behoben, die vom ersten Fehler ausgelöste Ausführung von OB 82 ist beendet und der zweite Fehler liegt weiterhin vor.

OB 82 enthält Anlaufinformationen, anhand deren Sie ermitteln können, ob das Ereignis wegen des Auftretens oder Verschwindens eines Fehlers ausgelöst wurde, und welches Gerät und welcher Kanal den Fehler gemeldet haben. Sie können in OB 82 Anweisungen programmieren, um diese Anlaufwerte zu untersuchen und entsprechende Maßnahmen zu ergreifen.

Tabelle 4-3 Anlaufinformationen für OB 82

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

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

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 oder als Übertragungskarte 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. geforderte E/A). Die CPU führt das Anwenderprogramm dann von der Programmkarte aus.
- Auf der Programmkarte können Sie auch Datenprotokolldateien speichern (Seite 116). Die Programmkarte bietet eine größere Kapazität als der interne Speicher der CPU. Mit der Webserver-Funktion (Seite 189) der CPU können Sie die Datenprotokolldateien auf einen Computer herunterladen.

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- 4 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.
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 65535 (16#FFFF), dies entspricht dem 6. Juni 2169. 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 (USInt): 1 bis 12 – Wochentag (USInt): 1 (Sonntag) bis 7 (Samstag) – Stunden (USInt): 0 bis 23 – Minuten (USInt): 0 bis 59 – Sekunden (USInt): 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.

Datentypen	Beschreibung
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>
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 Wert nur in einen physikalischen Eingang (Ex.y:P) oder in einen physikalischen Ausgang (Ax.y:P) geschrieben. 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 233).

Tabelle 4- 5 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)

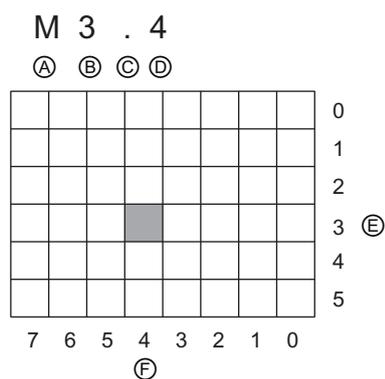
Speicherbereich	Beschreibung	Forcen	Remanent
L temporärer Speicher	Temporäre, lokale Daten für einen Baustein	Nein	Nein
DB Datenbaustein	Datenspeicher und auch Parameterspeicher für FBs	Nein	Ja (optional)

- 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).

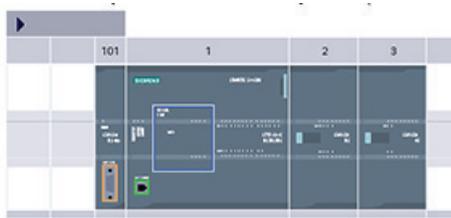


Absolute Adresse eines Speicherbereichs:

- 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 DC/DC	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 (t)		
HSC_2	1.17			Schneller Zähler (t)		
HSC_3	1.18			Schneller Zähler (t)		
HSC_4	1.19			Schneller Zähler (t)		
HSC_5	1.20			Schneller Zähler (t)		
HSC_6	1.21			Schneller Zähler (t)		
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 im Konfigurationsbildschirm 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).

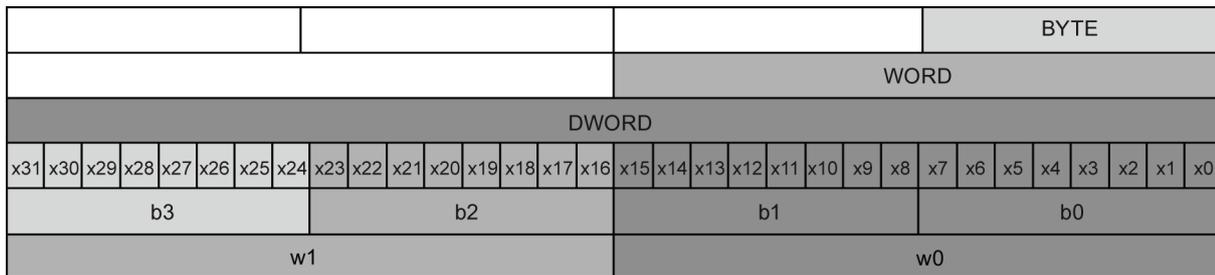
Das Bild zeigt ein Beispiel für eine CPU 1214C mit zwei SMs und einem SB. In diesem Beispiel können Sie in der Adresse des Moduls DE8 die 8 in 2 ändern. Das Werkzeug unterstützt Sie, indem Adressbereiche, die die falsche Größe haben oder mit anderen Adressen in Konflikt stehen, geändert werden.

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 0-2 oder das Wort 0 zugegriffen werden. Auf eine Variable mit Bytegröße kann über die Bits 0-8 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-Variablentabelle 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>

Unter Auto hotspot finden Sie Informationen zur Syntax für die Adressierung lokaler Variablen und von PLC-Variablen.

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

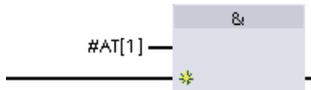
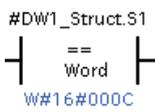
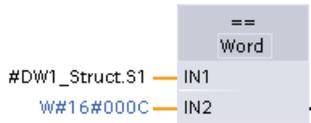
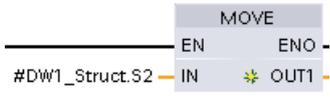
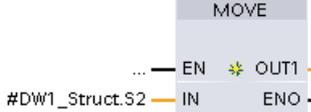
Tabelle 4- 6 Überlagerung eines Bytes mit einem Booleschen Array

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>

Unter Auto hotspot finden Sie Informationen zur Syntax für die Adressierung lokaler Variablen und von PLC-Variablen.

Regeln

- Die Überlagerung von Variablen ist nur in FB- und FC-Bausteinen mit Standardzugriff möglich.
- Sie können Parameter für alle Bausteintypen 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 zwei 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 (A0.0 bis A0.3 und A4.0 bis A4.3) 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.

ACHTUNG

Überschreiten Sie nicht die maximale Impulsfrequenz.

Wie im *S7-1200 Systemhandbuch* beschrieben, beträgt die maximale Impulsfrequenz der Impulsgeneratoren 100 kHz für die digitalen Ausgänge der CPU, 20 kHz für die digitalen Ausgänge des Signalboards und 200 kHz für die digitalen Ausgänge der schnellen SBs.

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.

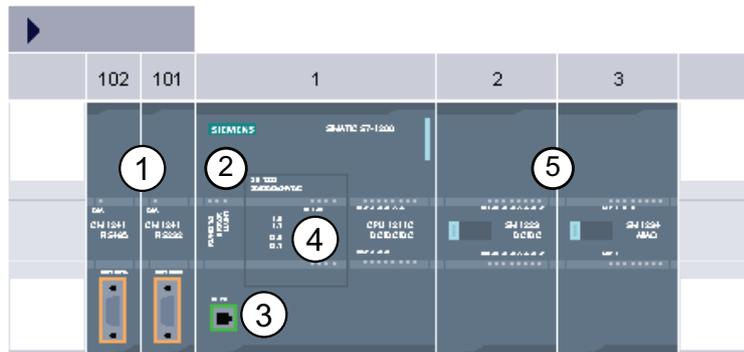
Die zwei Impulsgeneratoren sind spezifischen digitalen Ausgängen zugeordnet (siehe folgende Tabelle). Sie können integrierte CPU-Ausgänge oder die Ausgänge eines optionalen Signalboards nutzen. In der folgenden Tabelle sind die Adressen der Ausgänge aufgeführt (wobei die Standardkonfiguration der Ausgänge vorausgesetzt wird). Wenn Sie die Adressen der Ausgänge geändert haben, entsprechen die Adressen den von Ihnen zugewiesenen. Unabhängig davon nutzt PTO1/PWM1 die ersten beiden digitalen Ausgänge und PTO2/PWM2 nutzt die nächsten beiden digitalen Ausgänge, entweder auf der CPU oder dem gesteckten Signalboard. 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.

Tabelle 4- 7 Ausgangszuweisungen für die Impulsgeneratoren

Beschreibung	Standardmäßige Ausgangsbelegung	Impuls	Richtung
PTO 1	In CPU integriert	A0.0	A0.1
	Signalboard	A4.0	A4.1
PWM 1	In CPU integriert	A0.0	--
	Signalboard	A4.0	--
PTO 2	In CPU integriert	A0.2	A0.3
	Signalboard	A4.2	A4.3
PWM 2	In CPU integriert	A0.2	--
	Signalboard	A4.2	--

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): bis zu 3, in Steckplätzen 101, 102 und 103
- ② CPU: Steckplatz 1
- ③ Ethernet-Anschluss der CPU
- ④ Signalboard (SB): 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, 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 Projektamen auf "Neues Gerät hinzufügen".

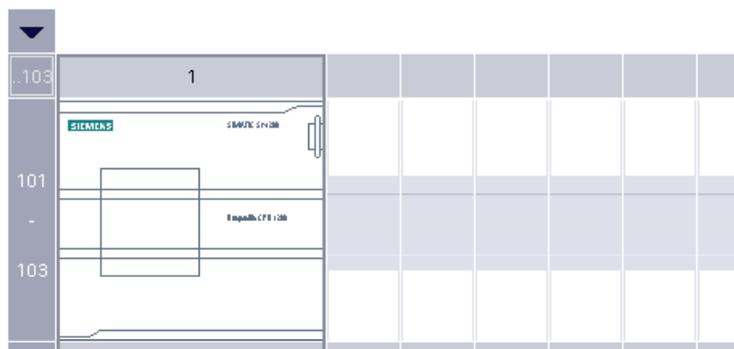


5.1 Konfiguration für eine nicht spezifizierte CPU erkennen



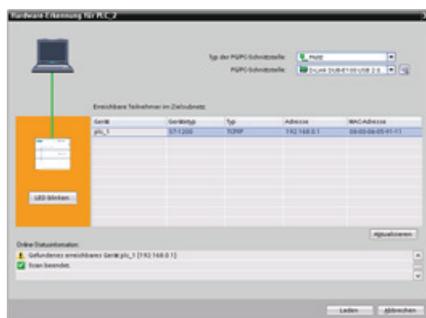
Wenn eine Verbindung zu einer CPU besteht, können Sie die Konfiguration dieser CPU einschließlich evtl. vorhandener Module aus dem Gerät in Ihr Projekt laden. Legen Sie dazu einfach ein neues Projekt an und wählen Sie anstelle einer bestimmten CPU die "nicht spezifizierte CPU". (Sie können auch die Gerätekonfiguration 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.

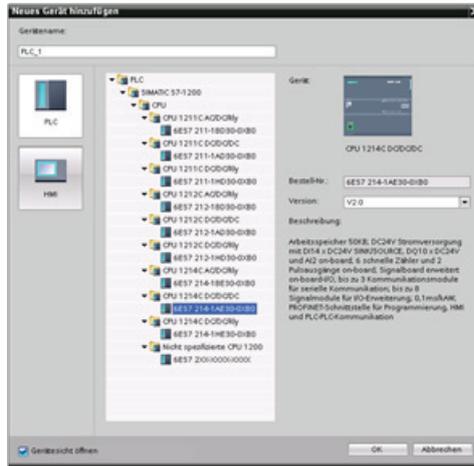


Das Gerät ist nicht spezifiziert.
 → Bitte verwenden Sie den [Hardware-Katalog](#) um die CPU zu spezifizieren,
 → oder [ermitteln](#) Sie die Konfiguration des angeschlossenen Gerätes.

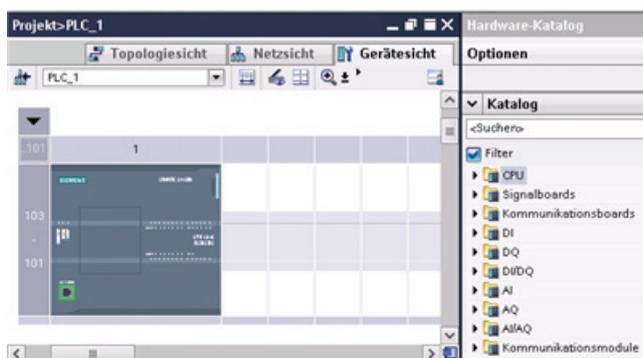
Nachdem Sie im Online-Dialog die CPU ausgewählt und auf die Schaltfläche zum Laden geklickt haben, lädt STEP 7 die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Sie können dann die Parameter für die CPU und die Module (Seite 80) konfigurieren.



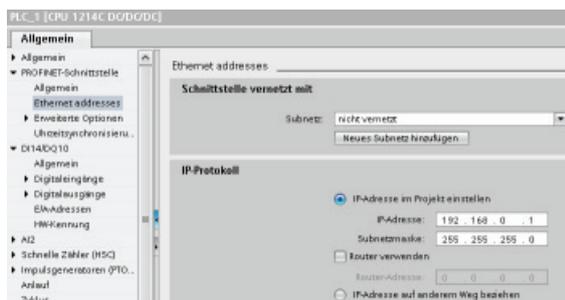
5.2 CPU zur Konfiguration hinzufügen



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 80).

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.

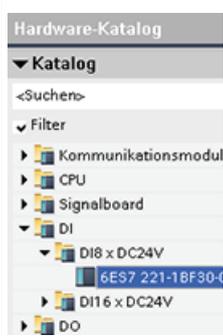
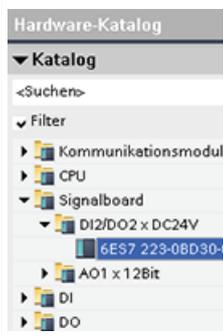
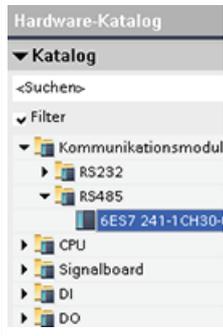
5.3 Module zur Konfiguration hinzufügen

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.
- 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.

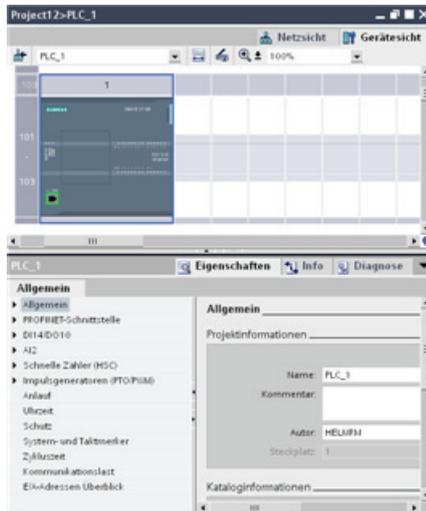
Um ein Modul in die Hardwarekonfiguration einzufügen, selektieren Sie das Modul im Hardwarekatalog und doppelklicken, oder Sie ziehen es in den markierten Steckplatz.

Tabelle 5- 1 Modul zur Gerätekonfiguration hinzufügen

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

5.4 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".



- 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

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

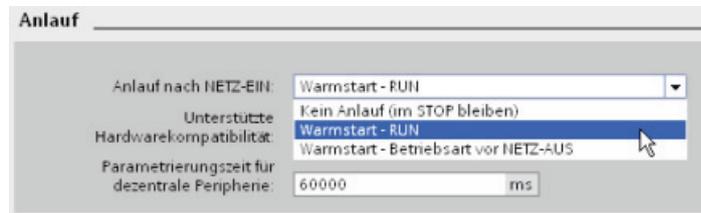
Immer wenn der Betriebszustand von STOP nach RUN wechselt, löscht die CPU das Prozessabbild der Eingänge, initialisiert das Prozessabbild der Ausgänge und verarbeitet die Anlauf-OBs. (Deshalb ergeben alle Lesezugriffe auf das Prozessabbild der Eingänge von Anweisungen im Anlauf-OB den Wert Null und nicht 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).

5.4.1 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.)
 - Das Bit "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 Flankenanweisungen, 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

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:

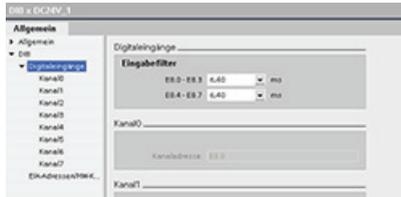
Tabelle 5- 3 Taktmerker

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.

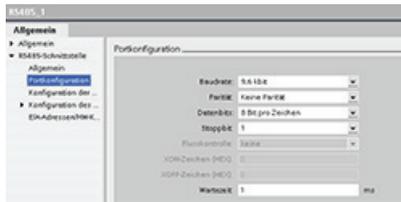
Konfigurieren des Betriebs der E/A und der Kommunikationsmodule

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.

5.5 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.

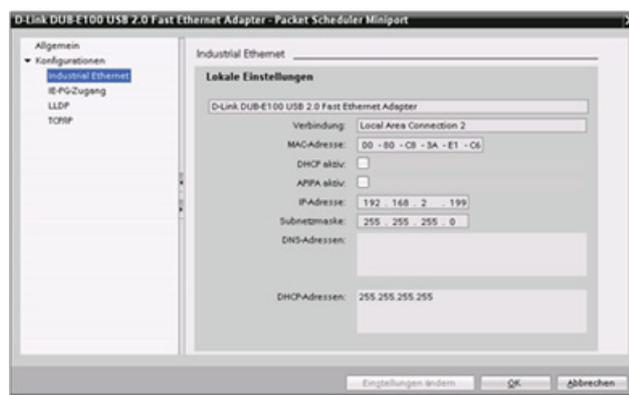
Vergewissern Sie sich vor dem Laden einer IP-Adresse in die CPU, dass die IP-Adresse Ihrer CPU mit der IP-Adresse Ihres Programmiergeräts kompatibel ist.

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

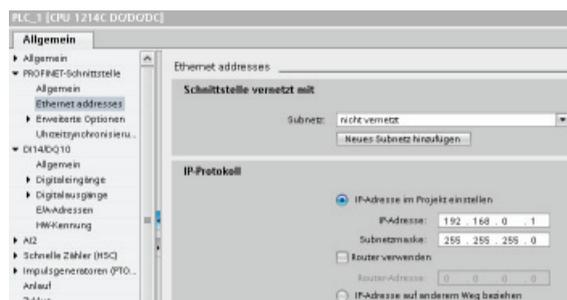
1. Erweitern Sie in der Projektnavigation den Ordner "Online-Zugänge", um Ihre Netzwerke anzuzeigen.
2. Wählen Sie das Netzwerk für die Verbindung zur CPU aus.
3. Klicken Sie mit der rechten Maustaste auf das spezifische Netzwerk, um das Kontextmenü aufzurufen.
4. Wählen Sie "Eigenschaften".

Hinweis

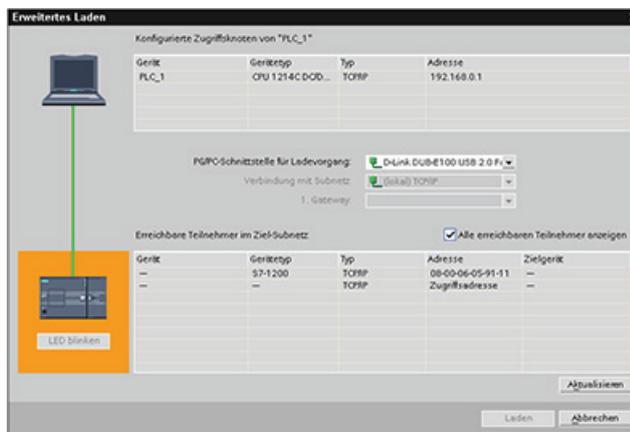
Die IP-Adresse der CPU muss mit der IP-Adresse und Subnetzmaske des Programmiergeräts kompatibel sein. Die geeignete IP-Adresse und Subnetzmaske für Ihre CPU erfahren Sie von Ihrem Netzwerkspezialisten.



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.

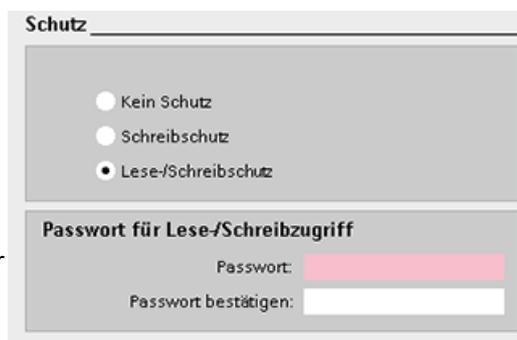
5.6 Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach

Die CPU bietet 3 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.

Die Groß- und Kleinschreibung des Passwortes ist zu beachten.

Um das Passwort 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 ein Passwort einzugeben.



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. Auf eine passwortgeschützte CPU hat jeweils immer nur ein Benutzer uneingeschränkten Zugriff. Der Passwortschutz gilt nicht für die Ausführung der Anweisungen des Anwenderprogramms einschließlich Kommunikationsfunktionen. Die Eingabe des richtigen Passworts gestattet den ungehinderten Zugriff auf alle Funktionen.

Die Kommunikation zwischen CPUs (über die Kommunikationsfunktionen in den Codebausteinen) wird durch die Schutzstufe der CPU nicht eingeschränkt. Auch die HMI-Funktionalität bleibt uneingeschränkt erhalten.

Tabelle 5-4 Schutzstufen der CPU

Schutzstufe	Zugangsbeschränkungen
Kein Schutz	Ungehinderter Zugang ohne Passwortschutz.
Schreibschutz	HMI-Zugang und ungehinderte 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.
Lese-/Schreibschutz	HMI-Zugang und ungehinderte 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.

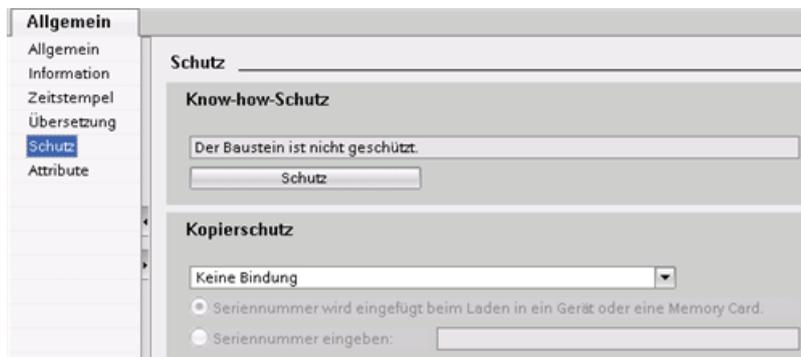
5.6.1 Knowhow-Schutz

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".



5.6.2 Kopierschutz

Eine weitere Sicherheitsfunktion ermöglicht Ihnen, das Programm oder die Codebausteine mit einer bestimmten Memory Card oder CPU zu verknüpfen. Diese Funktion ist vor allem zum Schutz geistigen Eigentums nützlich. Wird ein Programm oder ein Baustein mit einem bestimmten Gerät verknüpft, so ist die Verwendung dieses Programms oder dieses Bausteins 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.

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

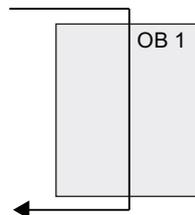
Beim Erstellen eines Anwenderprogramms für Automatisierungslösungen fügen Sie die Anweisungen des Programms in Codebausteine (OB, FB oder FC) ein.

Art der Struktur für das Anwenderprogramm wählen

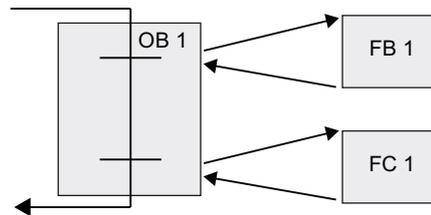
Je nach den Anforderungen Ihrer Anwendung können Sie eine lineare oder eine modulare Struktur für Ihr Anwenderprogramm wählen.

- 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:

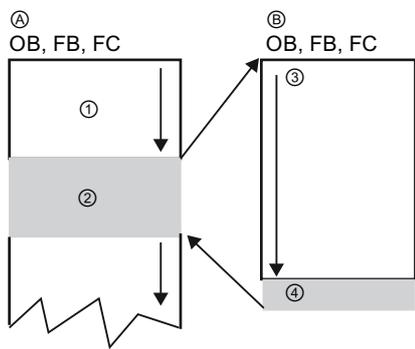


Modulare Struktur:



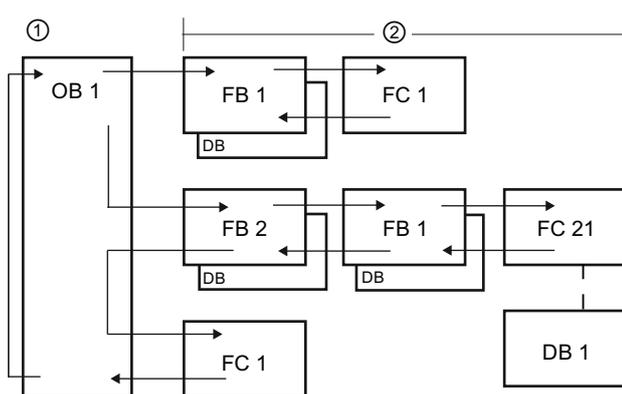
Modulare Codebausteine erstellen Sie durch den Entwurf von FBs und FCs für die Ausführung allgemeiner Aufgaben. Anschließend strukturieren Sie Ihr Programm, indem andere Codebausteine wiederum die wiederverwendbaren Module aufrufen. Der aufrufende Baustein gibt gerätespezifische Parameter an den aufgerufenen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

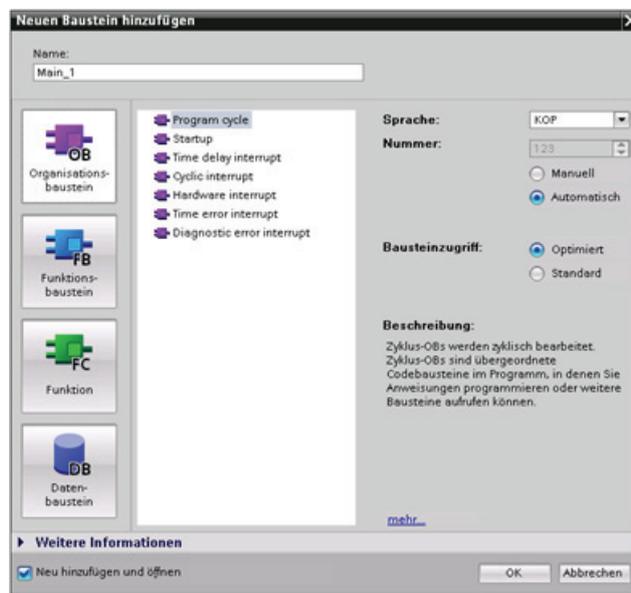
- Sie können wiederverwendbare Codebausteine für Standardaufgaben, wie z. B. für die Steuerung einer Pumpe oder eines Motors, erstellen. Sie können diese allgemeinen Codebausteine ferner in einer Bibliothek speichern, die für verschiedene Anwendungen oder Lösungen verwendet werden kann.
- Durch die modulare 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 Anlaufereignisse und voreingestelltes Anlaufverhalten.

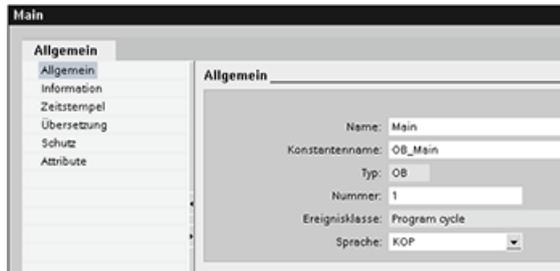
Der Programmzyklus-OB enthält das Hauptprogramm. Sie können mehrere Programmzyklus-OBs in Ihr Anwenderprogramm aufnehmen. Im Betriebszustand RUN werden die Programmzyklus-OBs mit der niedrigsten Prioritätsstufe ausgeführt und können durch alle anderen Arten der Programmverarbeitung unterbrochen werden. (Anlauf-OBs unterbrechen die Programmzyklus-OBs nicht, weil die CPU die Anlauf-OBs vor dem Wechsel nach 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. Bei vielen Anwendungen ist das gesamte Anwenderprogramm in einem einzigen OB enthalten (z. B. in dem standardmäßigen Programmzyklus-OB "OB 1").



Sie können andere OBs anlegen, um bestimmte Funktionen auszuführen, z. B. beim Anlaufen, für die Alarm- und Fehlerbearbeitung oder für die Ausführung eines bestimmten Programmcodes in bestimmten Abständen.

Einen neuen OB für Ihr Anwenderprogramm legen Sie im Dialog "Neuen Baustein hinzufügen" an.

Die CPU ermittelt die Reihenfolge für die Bearbeitung von Alarmereignissen anhand der den einzelnen OBs zugeordneten Prioritäten (Seite 56).



Die Ablaufparameter eines OBs können Sie ändern. Beispielsweise können Sie die Zeit für einen Verzögerungs-OB oder für einen Weckalarm-OB einstellen.

Anlegen eines weiteren OBs in einer OB-Klasse: Sie können mehrere OBs für Ihr Anwenderprogramm anlegen, auch für die Klassen der Programmzyklus- und Anlauf-OBs. Im Dialog "Neuen Baustein hinzufügen" können Sie einen OB anlegen. Geben Sie den Namen für Ihren OB und eine OB-Nummer ein, die größer als 200 sein muss.

Wenn Sie mehrere Programmzyklus-OBs für Ihr Anwenderprogramm anlegen, führt die CPU die einzelnen Programmzyklus-OBs in numerischer Reihenfolge aus. Dabei ist der erste OB der OB mit der kleinsten Nummer, typischerweise OB 1. Beispiel: Nach dem Ende des ersten Programmzyklus-OB (OB 1) führt die CPU den zweiten Programmzyklus-OB aus (z. B. OB 200).

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.

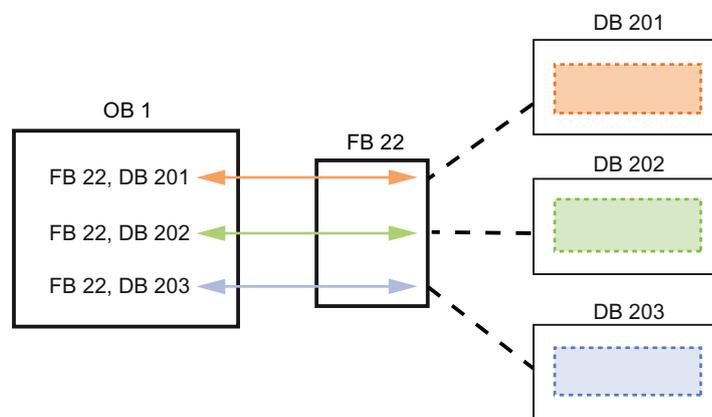
Ein Funktionsbaustein (FB) ist eine Art Unterprogramm mit Speicher. Ein Funktionsbaustein (FB) ist ein Codebaustein, dessen Aufrufe mit Bausteinparametern programmiert werden können. Ein FB speichert die Eingangs- (IN), Ausgangs- (OUT) und Durchgangparameter (IN_OUT) in einem variablen Speicherbereich,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

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).

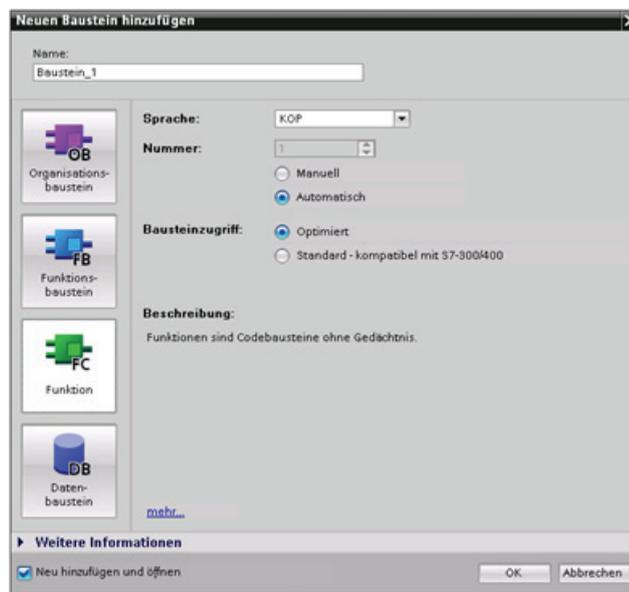
Ihr Anwenderprogramm kann Daten in den speziellen Speicherbereichen der CPU, z. B. für Eingänge (E), Ausgänge (A) und Merker (M), speichern. Außerdem können Sie einen Datenbaustein (DB) für den schnellen Zugriff auf die im Programm selbst gespeicherten Daten verwenden.

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 zu ergänzenden 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 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 eine Aufrufanweisung zu erstellen.

Hinweis

Mit dem Anwenderprogramm können Sie keinen OB aufrufen, da OBs ereignisgesteuert (Seite 56) 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:

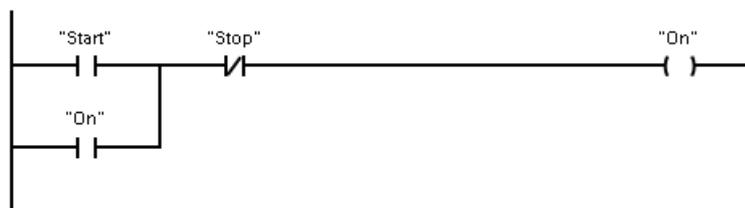
- KOP (Kontaktplan) ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen.
- FUP (Funktionsplan) ist eine Programmiersprache, die auf den grafischen Logiksymbolen der Booleschen Algebra basiert.
- SCL (Structured Control Language) ist eine textbasierte, höhere Programmiersprache.

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.

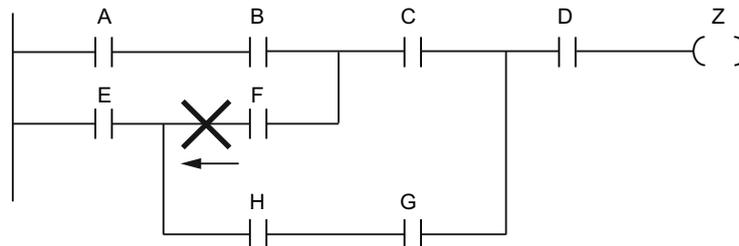
STEP 7 begrenzt die maximale Anzahl von Anweisungen (Zeilen und Spalten) in einem KOP-Netzwerk nicht.

Hinweis

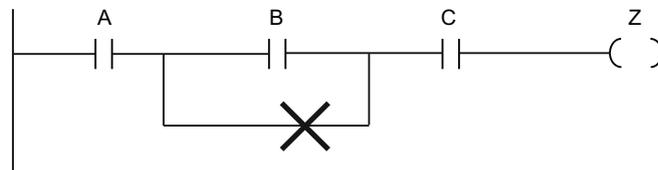
Jedes KOP-Netzwerk muss mit einer Spule oder einer Box abgeschlossen werden.

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 Bausteinstruktur 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)

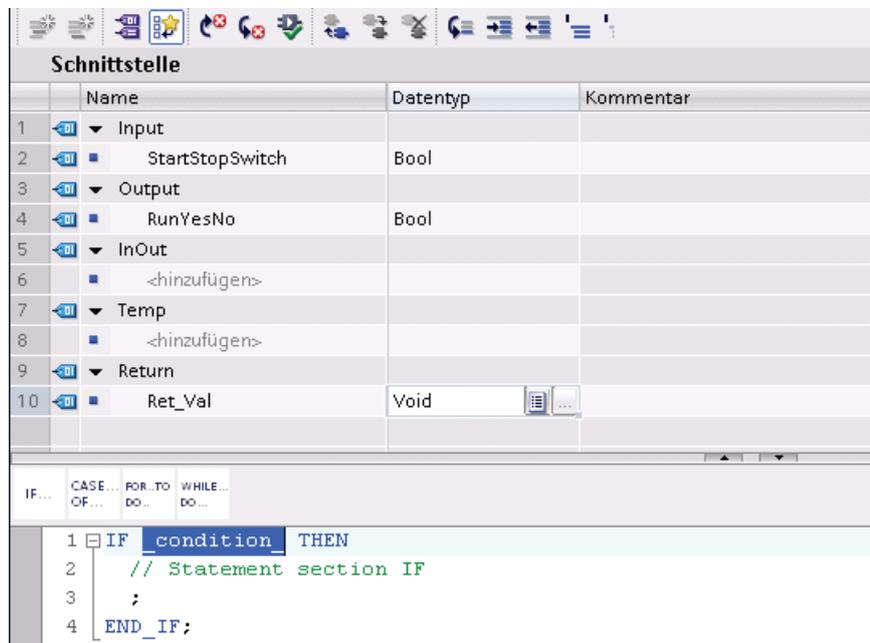
Arithmetische Operatoren können verschiedene numerische Datentypen verarbeiten. Der Datentyp des Ergebnisses wird vom Datentyp des höchstwertigen Operanden festgelegt. Beispiel: Eine Multiplikationsoperation mit einem Operanden vom Typ INT und einem Operanden vom Typ REAL ergibt einen Wert vom Typ REAL 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. 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 Abschnitt mit dem SCL-Codebaustein können Sie die folgenden Arten von Parametern deklarieren:

- Eingang, Ausgang, Durchgang und Rückgabewert: 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 Variablenliste verwendet.
- Statisch (nur FBs; die obige Abbildung zeigt einen FC): Statische Variablen dienen zum Speichern von statischen Zwischenergebnissen im Instanzdatenbaustein. Statische Daten werden gespeichert, 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.



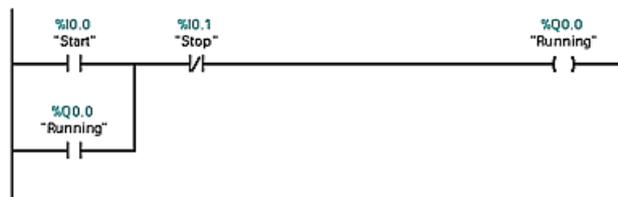
In diesem Beispiel entsprechen die Variablen für "Start" und "On" (aus der Variablen-tabelle des Systems) den Variablen "StartStopSwitch" und "RunYesNo" in der Deklarationstabelle des SCL-Programms.

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.

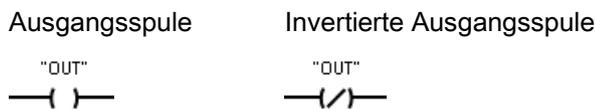
Wenn Sie eine Spule mit derselben Adresse an mehreren Stellen im Programm verwenden, so hängt vom Ergebnis der letzten Berechnung im Anwenderprogramm der Zustand des Werts ab, der während der Aktualisierung der Ausgänge in den physikalischen Ausgang geschrieben wird.

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

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.

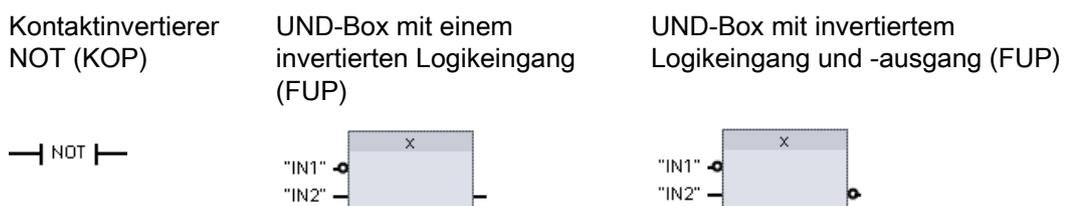


- Ist ein Signalfluss durch eine Ausgangsspule vorhanden, so wird das Ausgangsbit auf 1 gesetzt.
- Ist kein Signalfluss durch eine Ausgangsspule vorhanden, so wird das Ausgangsbit auf 0 gesetzt.
- Ist ein Signalfluss durch eine invertierte Ausgangsspule vorhanden, so wird das Ausgangsbit auf 0 gesetzt.
- Ist kein Signalfluss 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 Speicherkennung A, so schaltet die CPU das Ausgangsbit im Prozessabbildregister ein oder aus und setzt das angegebene Bit jeweils entsprechend dem Signalfluss. Die Ausgangssignale für Ihre Steuerstellglieder werden mit den Ausgangsklemmen der 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.



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

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

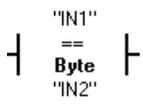
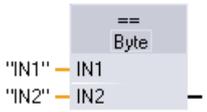
Bei der FUP-Programmierung werden die KOP-Netzwerke aus Kontakten durch die Box-Netzwerke UND (&), ODER (>=1) und Exklusiv ODER (x) dargestellt, in denen Sie Bitwerte für die Ein- und -Ausgänge der Box angeben können. Sie können ferner Verschaltungen mit anderen Logik-Boxen herstellen und so Ihre eigene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

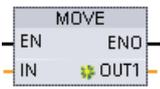
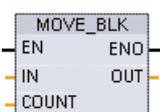
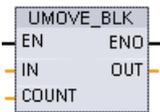
Anweisung	SCL	Beschreibung
<p>KOP:</p> 	<pre> out := in1 == in2; out := in1 <> in2; out := in1 >= in2; out := in1 <= in2; out := in1 > in2; out := in1 < in2; </pre>	<ul style="list-style-type: none"> • Gleich (==): Der Vergleich ist wahr, wenn IN1 gleich IN2 ist. • Ungleich (<>): Der Vergleich ist wahr,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

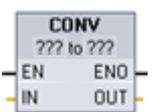
- MOVE kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse. Um einen weiteren Ausgang hinzuzufügen, klicken Sie auf das Symbol neben dem Parameter OUT1.
- MOVE_BLK (Bereich kopieren) und UMOVE_BLK (Bereich ununterbrechbar kopieren) kopieren einen Baustein mit Datenelementen 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 Variablennamen der Parameter IN und OUT in der PLC-Variablentabelle 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>out := 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>out := 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 Nachkommastellen der Realzahl werden auf den nächsten ganzzahligen Wert gerundet (IEEE- aufrunden). Wenn die Zahl genau die Hälfte der Spanne zwischen zwei Ganzzahlen ist (z. B. 10,5), wird die Zahl auf die gerade Ganzzahl gerundet. Beispiel:</p> <ul style="list-style-type: none"> • ROUND (10.5) = 10 • ROUND (11.5) = 12
	<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

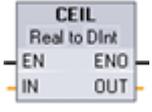
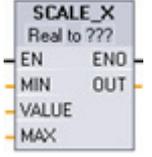
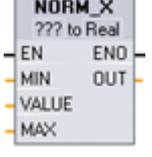
KOP/FUP	SCL	Beschreibung
	<pre>out := CEIL (in);</pre>	<p>Konvertiert eine Realzahl (Real oder LReal) in die nächste Ganzzahl, die größer oder gleich der ausgewählten Realzahl ist (IEEE - Runden auf +unendlich).</p>
	<pre>out := FLOOR (in);</pre>	<p>Konvertiert eine Realzahl (Real oder LReal) in die 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_); or out := value (max-min) + min;</pre>	<p>Skaliert den normalisierten Realparameter VALUE (0,0 <= VALUE <= 1,0) in den mit den Parametern MIN und MAX vorgegebenen Datentyp und Wertebereich: $OUT = VALUE (MAX - MIN) + MIN$</p>
	<pre>out := NORM_X(min := _undef_in_ value := _undef_in_ max := _undef_in_); or out := (value-min) / (max-min);</pre>	<p>Normalisiert den Parameter VALUE innerhalb des von den Parametern MIN und MAX angegebenen Wertebereichs: $OUT = (VALUE - MIN) / (MAX - MIN)$, dabei ist (0,0 <= OUT <= 1,0)</p>

6.3.4 Mathematik ganz einfach mit der Anweisung Calculate

Tabelle 6- 7 Anweisung CALCULATE

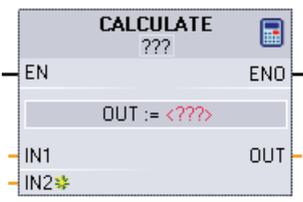
KOP/FUP	SCL	Beschreibung
	<p>Verwenden Sie die herkömmlichen mathematischen SCL-Ausdrücke, um die Gleichung zu erstellen.</p>	<p>Mit der Anweisung CALCULATE können Sie eine mathematische Funktion erstellen, die Eingänge (IN1, IN2, .. INn) verarbeitet und das Ergebnis an OUT entsprechend der von Ihnen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

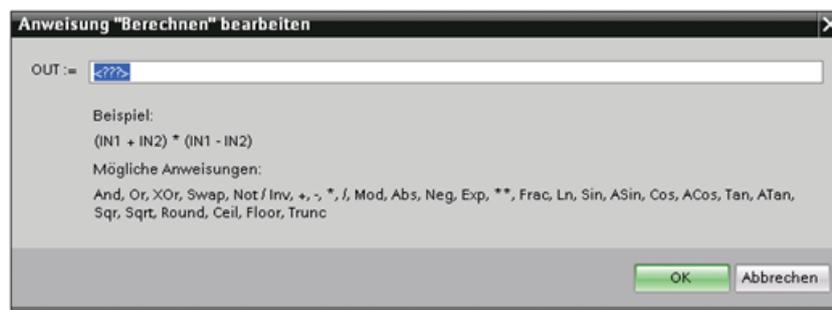
Tabelle 6- 8 Datentypen für die Parameter

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)

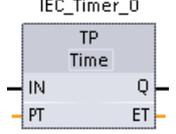
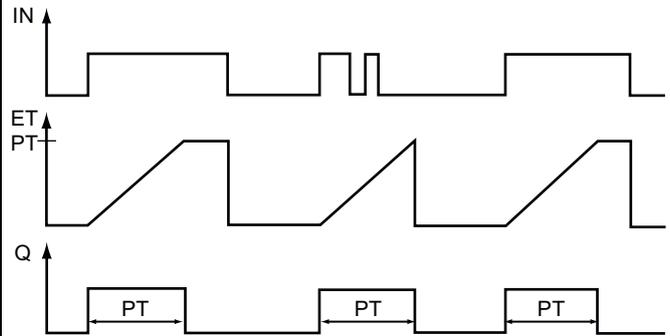
KOP/FUP	SCL	Zeitdiagramm
 <p>IEC_Timer_0</p> <p>TP Time</p> <p>IN Q PT ET</p>	<pre>"timer_db".TP(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	 <p>The timing diagram shows three input pulses (IN). The first is a single pulse, the second is a double pulse, and the third is a wider pulse. The ET signal (top) shows a ramp up from 0 to PT for each pulse. The Q signal (bottom) shows a pulse of width PT for each input pulse.</p>
<p>TP_DB</p> <p>(TP)</p> <p>"PRESET_Tag"</p>		

Tabelle 6-10 TON (Einschaltverzögerung)

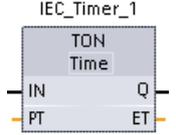
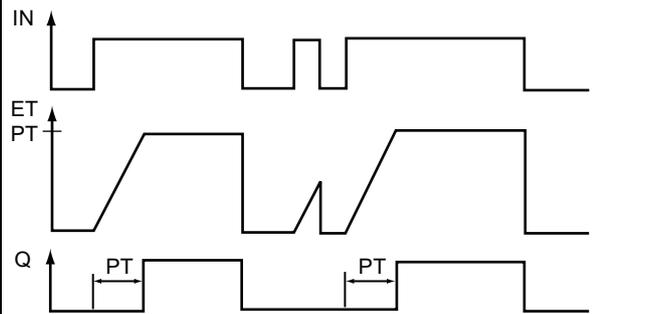
KOP/FUP	SCL	Timingdiagramm
 <p>IEC_Timer_1</p> <p>TON Time</p> <p>IN Q PT ET</p>	<pre>"timer_db".TON(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	 <p>The timing diagram shows three input pulses (IN). The ET signal (top) shows a ramp up from 0 to PT for each pulse. The Q signal (bottom) shows a pulse that starts at the rising edge of IN and has a width of PT.</p>
<p>TON_DB</p> <p>(TON)</p> <p>"PRESET_Tag"</p>		

Tabelle 6-11 TOF (Ausschaltverzögerung)

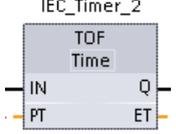
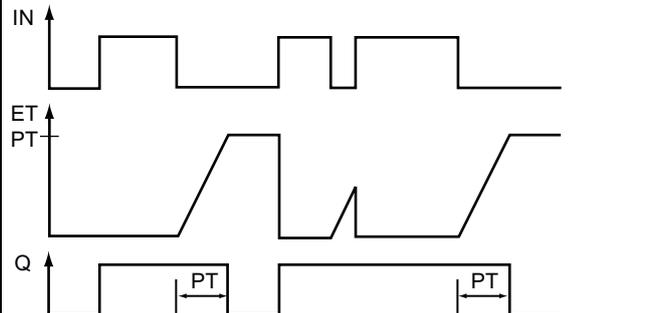
KOP/FUP	SCL	Timingdiagramm
 <p>IEC_Timer_2</p> <p>TOF Time</p> <p>IN Q PT ET</p>	<pre>"timer_db".TOF(IN:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	 <p>The timing diagram shows three input pulses (IN). The ET signal (top) shows a ramp up from 0 to PT for each pulse. The Q signal (bottom) shows a pulse that starts at the rising edge of IN and has a width of PT.</p>
<p>TOF_DB</p> <p>(TOF)</p> <p>"PRESET_Tag"</p>		

Tabelle 6- 12 TONR (Zeit akkumulieren)

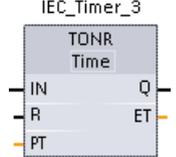
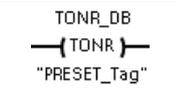
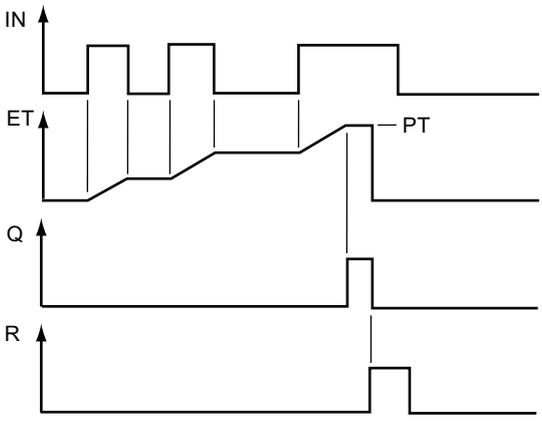
KOP/FUP	SCL	Timingdiagramm
 	<pre>"timer_db".TONR(IN:=_bool_in_, R:=_bool_in_, PT:=_undef_in_, Q=>_bool_out_, ET=>_undef_out_);</pre>	

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

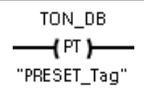
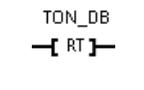
KOP	Beschreibung
	<p>Sie können die Spulenanweisungen Zeit voreinstellen -(PT)- und Zeit rücksetzen -(RT)- mit Box-Anweisungen oder Spulen verwenden. Diese Spulenanweisungen können in der Mitte eines Strompfads angeordnet werden. Der Signalzustand des Spulenausgangs ist immer der gleiche wie am Spuleneingang.</p>
	<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 0 zurückgesetzt. • 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

Parameter	Datentyp	Beschreibung
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 als DInt -Daten	T#-24d_20h_31m_23s_648ms bis T#24d_20h_31m_23s_647ms Gespeichert als -2.147.483.648 ms bis +2.147.483.647 ms

¹ Der negative Bereich des oben dargestellten Datentyps TIME kann für die Zeiten nicht verwendet werden. Negative Werte für PT (voreingestellte Zeit) werden bei Ausführung der Zeitanweisung auf Null gesetzt. ET (abgelaufene Zeit) ist immer ein positiver Wert.

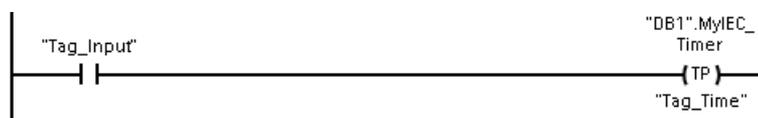
Programmierung von Zeiten

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

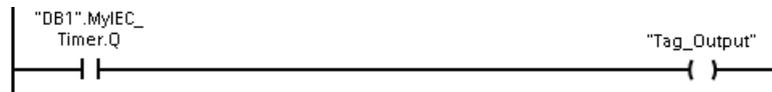
- Sie können mehrere Aktualisierungen einer Zeit im gleichen Zyklus haben. Die Zeit wird bei jeder Ausführung der Zeitanweisung (TP, TON, TOF, TONR) aktualisiert sowie jedes Mal, wenn das Element ELAPSED oder Q der Zeitstruktur als Parameter einer anderen ausgeführten Anweisung verwendet wird. Dies ist 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 Vorwärtszähleingang (CU) oder der Rückwärtszähleingang (CD) von 0 nach 1 wechselt.

Die S7-1200 bietet außerdem schnelle Zähler (Seite 123) (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.

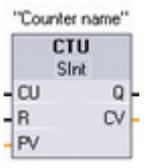
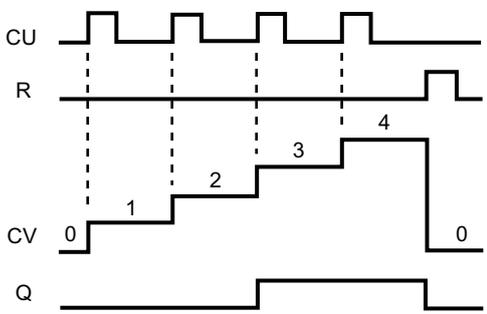
Hinweis

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).

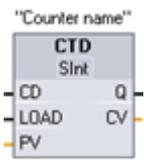
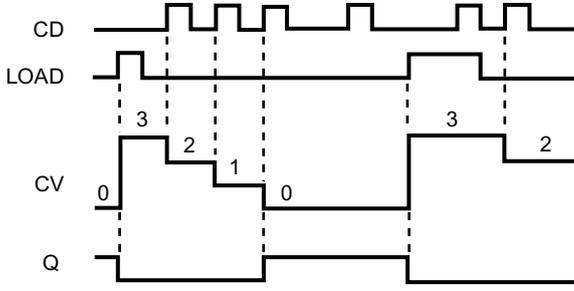
Tabelle 6- 17 CTU Vorwärtszähler

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.

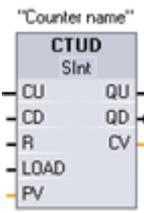
Tabelle 6- 18 CTD Rückwärtszähler

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

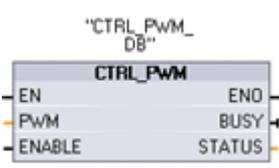
KOP/FUP	SCL	Funktionsweise
	<pre>"ctud_db".CTUD (CU:=_bool_in, CD:=_bool_in, R:=_bool_in, LOAD:=_bool_in, PV:=_undef_in, QU=>_bool_out, QD=>_bool_out, CV=>_undef_out);</pre>	

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, dann wird CV auf 0 zurückgesetzt.

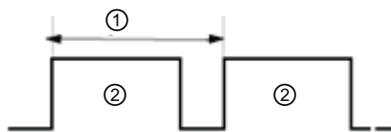
6.3.7 Impulsdauermodulation (PWM)

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.

Ihre Datenprotokolldateien verwalten Sie mit dem integrierten PLC-Webserver. Laden Sie die letzten Datensätze oder alle Daten herunter, setzen Sie Datensätze zurück oder löschen Sie Protokolldateien. Alle diese Funktionen stehen auf der Standard-Webseite "Datenprotokolle" zur Verfügung. Nachdem Sie eine Datenprotokolldatei auf Ihren PC übertragen haben, können Sie die Daten mit gängigen Tabellenkalkulationsprogrammen wie Microsoft Excel auswerten.

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

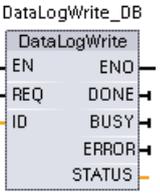
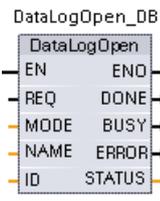
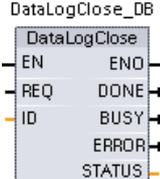
KOP/FUP	SCL	Beschreibung
 <p>The diagram shows a function block named 'DataLogWrite_DB'. It has several inputs and outputs: 'EN' (enable) and 'END' (end) are inputs; 'REQ' (request) and 'DONE' (done) are outputs; 'ID' (ID) is an input; 'BUSY' (busy) is an output; 'ERROR' (error) is an output; and 'STATUS' (status) is an output.</p>	<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>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>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
 <p>DataLogOpen_DB</p> <p>DataLogOpen</p> <p>EN END</p> <p>REQ DONE</p> <p>MODE BUSY</p> <p>NAME ERROR</p> <p>ID STATUS</p>	<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. Mehrere Datenprotokolle können gleichzeitig geöffnet sein.</p>
 <p>DataLogClose_DB</p> <p>DataLogClose</p> <p>EN END</p> <p>REQ DONE</p> <p>ID BUSY</p> <p> ERROR</p> <p> STATUS</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 233) 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").

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

Spalte	Beschreibung
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.

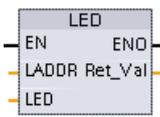
- Die meisten Zeitstempel- und Schnittstellenkonflikte lassen sich durch erneutes Übersetzen der Codebausteine beheben.
- Wenn durch Übersetzen die Inkonsistenzen nicht beseitigt werden, navigieren Sie mit dem Programmierer über die Verknüpfung in der Spalte "Details" zur Quelle des Problems. Hier können 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.

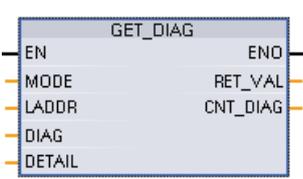
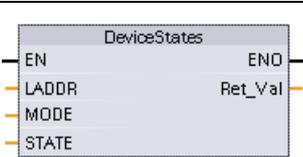
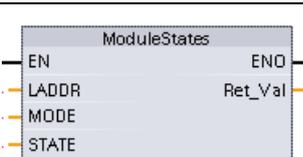
Tabelle 6- 25 Anweisung LED

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 rot • 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>	Mit der Anweisung GET_DIAG lesen Sie die Diagnoseinformationen aus einem angegebenen Hardwaregerät.
	<pre>ret_val := DeviceStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout_);</pre>	Mit der Anweisung DeviceStates lesen Sie den Zustand der PROFINET IO- oder PROFIBUS DP-Geräte.
	<pre>ret_val := ModuleStates(laddr:=_hw_io_in_, mode:=_uint_in_, state:=_variant_inout);</pre>	Mit der Anweisung ModuleStates lesen Sie den Zustand der PROFINET IO-Geräte.

6.6 Schneller Zähler (HSC)

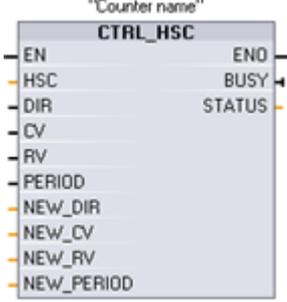
Die schnellen Zähler (HSC) zählen Ereignisse, die schneller als die Ausführungsrate des OBs auftreten. Die Anweisung CTRL_HSC steuert die Funktionsweise des HSC.

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 DB abgelegte Struktur, um Daten zu speichern.</p> <p>Der HSC 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.</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 Prozessabbaus, 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 das 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.

- 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 eines schnellen Zählers

Der schnelle Zähler (HSC) zählt Ereignisse, die schneller als die Ausführungsrate des OBs auftreten. 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. Mit der Anweisung CTRL_HSC kann Ihr Anwenderprogramm 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.

Das Anwenderprogramm lädt den ersten von mehreren voreingestellten Werten in den schnellen Zähler. Die Ausgänge werden vom Anwenderprogramm für die Zeitspanne aktiviert, während der der aktuelle Zählwert kleiner als der voreingestellte Wert ist. Das Anwenderprogramm konfiguriert den HSC so, dass ein Alarm ausgelöst wird, wenn der Zählerwert gleich dem Referenzwert ist ($CV = RV$), wenn der Zähler zurückgesetzt wird oder wenn ein Richtungswechsel auftritt.

Wenn ein Alarmereignis $CV = RV$ auftritt, lädt das Anwenderprogramm einen neuen Referenzwert und setzt den nächsten Zustand für die Ausgänge innerhalb des Alarm-OBs $CV = RV$. Wenn das Alarmereignis zum Zurücksetzen auftritt, lädt das Anwenderprogramm den ersten Referenzwert und setzt die ersten Ausgangszustände im Alarm-OB für Zurücksetzen, der Zyklus wird wiederholt.

Da die Alarme 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 Alarme 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.)

Tabelle 6- 28 Maximale Frequenz (kHz)

HSC		Einphasenzähler	Zweiphasenzähler und A/B-Zähler
HSC1	CPU	100 kHz	80 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC2	CPU	100 kHz	80 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz
HSC3	CPU	100 kHz	80 kHz
HSC4	CPU	30 kHz	20 kHz
HSC5	CPU	30 kHz	20 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz

HSC		Einphasenzähler	Zweiphasenzähler und A/B-Zähler
HSC6	CPU	30 kHz	20 kHz
	Schnelles SB	200 kHz	160 kHz
	SB	30 kHz	20 kHz

Funktionalität für den HSC auswählen

Alle HSCs arbeiten in der gleichen Zählerart auf die gleiche Weise. 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 immer in Hertz (Impulse pro Sekunde) angegeben, unabhängig von der eingestellten Frequenzmesszeit.

- 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. Ist HSC1 in einer Betriebsart, in der die integrierten Eingänge, aber nicht der externe Rücksetzeingang (E0.3) verwendet wird, so kann E0.3 beispielsweise für Flankenalarme oder für HSC2 belegt werden.

Tabelle 6- 29 Zählarten für den HSC

Typ	Eingang 1	Eingang 2	Eingang 3	Funktion
Einphasenzähler mit interner Richtungssteuerung	Takt	(Optional: Richtung)	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
Einphasenzähler mit externer Richtungssteuerung	Takt	Richtung	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert

Typ	Eingang 1	Eingang 2	Eingang 3	Funktion
Zweiphasenzähler mit 2 Takteingängen	Takt vorwärts	Takt rückwärts	-	Zählwert oder Frequenz
			Rücksetzen	Zählwert
A/B-Zähler	Phase A	Phase B	-	Zählwert oder Frequenz
			Phase Z	Zählwert

Eingangsadressen für den HSC

Hinweis

Die von schnellen Zählern verwendeten digitalen E/A werden während der Gerätekonfiguration zugewiesen. Wenn diesen Funktionen digitale E/A zugewiesen werden, können die Werte der Adressen der zugewiesenen E/A nicht durch die Forcefunktion einer Beobachtungstabelle geändert werden.

Wenn Sie die CPU konfigurieren, können Sie jeden HSC aktivieren und konfigurieren. Die CPU weist die Eingangsadressen für jeden HSC automatisch entsprechend der Konfiguration zu. (Bei einigen der HSCs können Sie auswählen, ob Sie die integrierten Eingänge der CPU oder die Eingänge eines SB nutzen möchten.)

ACHTUNG

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.

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

Die folgende Tabelle zeigt beispielsweise die HSC-Eingangszuweisungen für die integrierten E/A der CPU 1212C und eines SBs. (Wenn das SB nur zwei Eingänge hat, sind nur die Eingänge 4.0 und 4.1 verfügbar.)

- Bei Einphasenzählern: C ist der Zählengang, [d] ist der optionale Richtungseingang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei Zweiphasenzählern: CU ist der Vorwärtzählengang, CD ist der Rückwärtzählengang und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)
- Bei A/B-Zählern: A ist der Zählengang A, B ist der Zählengang B und [R] ist ein optionaler Eingang für externes Rücksetzen. (Rücksetzen ist nur verfügbar im Modus "Zählen".)

Tabelle 6- 30 HSC-Eingangszuweisungen bei der CPU 1212C

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

- ¹ HSC 1 und HSC 2 können für integrierte Eingänge oder für ein SB konfiguriert werden.
- ² HSC 5 und HSC 6 sind nur bei einem SB verfügbar. HSC 6 ist nur bei einem SB mit 4 Eingängen verfügbar.
- ³ Ein SB mit nur zwei Digitaleingängen bietet nur die Eingänge 4.0 und 4.1.

Zugreifen auf den aktuellen Wert des HSC

Wenn Sie einen Impulsgenerator als PTO aktivieren, wird diesem PTO ein entsprechender HSC zugewiesen. HSC1 wird PTO1 zugewiesen, und HSC2 wird PTO2 zugewiesen. Der zugewiesene HSC gehört vollständig dem PTO-Kanal, und der gewöhnliche Ausgang des HSC ist deaktiviert. Der HSC-Wert wird nur für die interne Funktionalität verwendet. Sie können den aktuellen Wert (z. B. in ID1000) nicht überwachen, wenn Impulse auftreten.

6.6.2 Konfiguration eines schnellen Zählers



Die CPU ermöglicht das Konfigurieren von maximal 6 schnellen Zählern. Unter den "Eigenschaften" der CPU können Sie die Parameter für jeden schnellen Zähler einrichten.

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

Sie aktivieren den spezifischen HSC durch Auswahl der Option "Aktivieren" für den HSC.

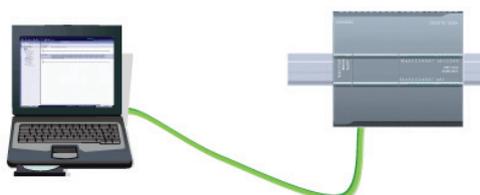


Nach der Aktivierung des HSC konfigurieren Sie die anderen Parameter wie Zählerfunktion, Anfangswerte, Rücksetzoptionen und Interruptereignisse.



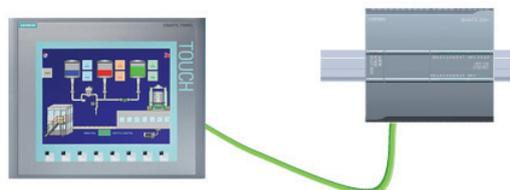
Weitere Informationen zum Konfigurieren des HSC finden Sie im Abschnitt zum Konfigurieren der CPU (Seite 80).

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.

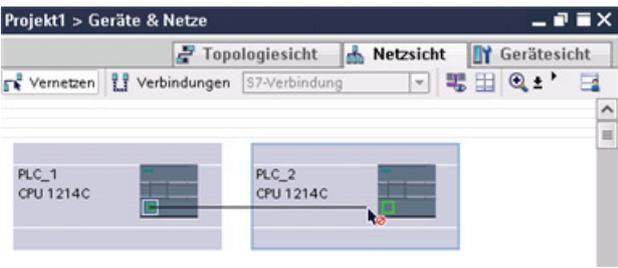
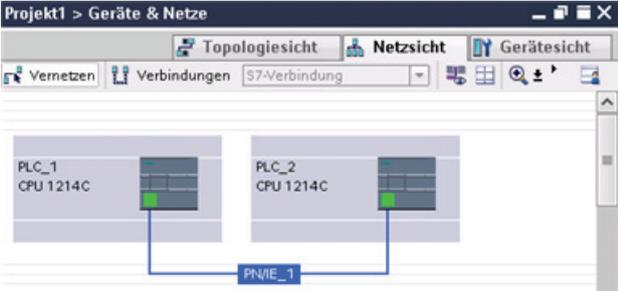
Durch Hinzufügen eines PROFIBUS CMs kann Ihre CPU auch als Master oder Slave in einem PROFIBUS-Netzwerk fungieren.

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:

- PROFINET (für den Austausch von Daten über das Anwenderprogramm mit anderen Kommunikationspartnern via Ethernet):
 - Für PROFINET und PROFIBUS unterstützt die CPU insgesamt 16 Geräte und 256 Submodule, mit maximal 8 PROFINET IO-Geräten und 128 Submodulen (wenn maximal acht PROFIBUS-Slaves oder -Submodule konfiguriert sind).
 - S7-Kommunikation
 - User Datagram Protocol (UDP)
 - ISO on TCP (RFC 1006)
 - Transport Control Protocol (TCP)

Als E/A-Steuerung über PROFINET RT kommuniziert die S7-1200 mit bis zu 8 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:
 - CM 1242-5: Betrieb als DP-Slave
 - CM 1243-5: Betrieb als DP-Master Klasse 1
 - Für PROFINET und PROFIBUS unterstützt die CPU insgesamt 16 Geräte und 256 Submodule mit maximal 16 PROFIBUS DP-Slavegeräten und 256 Submodulen (falls keine PROFINET IO-Geräte oder -Submodule konfiguriert sind).

Hinweis

Zu den 16 Geräten gehören u. a.:

- Die an den DP-Master (CM 1243-5) angeschlossenen DP-Slavemodule
- Alle an die CPU angeschlossenen DP-Slavemodule (CM 1242-5)
- Alle über den PROFINET-Port an die CPU angeschlossenen PROFINET-Geräte

So würde beispielsweise eine Konfiguration mit drei PROFIBUS-CMs (mit einem CM 1243-5-Master und zwei CM 1242-5-Slavemodulen) die maximale Anzahl von Slavemodulen, auf die der DP-Master (CM 1243-5) zugreifen kann, auf 14 verringern.

- AS-i: 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
- Teleservice-Kommunikation

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 CPU stellt zweckbestimmte HMI-Verbindungen bereit, um bis zu 3 HMI-Geräte zu unterstützen. (Sie können bis zu 2 SIMATIC Comfort Panels haben.) Wie viele HMI-Geräte insgesamt unterstützt werden, hängt von den Typen der HMI-Panels in Ihrer Konfiguration ab. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.
- PG-Verbindungen: Die CPU bietet Verbindungen, um 1 Programmiergerät (PG) zu unterstützen.
- Webserver-Verbindungen (HTTP): Die CPU bietet Verbindungen für den Webserver.

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 diese 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 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 GPRS

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 der Anweisung DPRD_DAT (konsistente Daten lesen) und DPWR_DAT (konsistente Daten schreiben) übertragen Sie maximal vier fortlaufende Bytes mit Ladeanweisungen, die auf den DP-Normslave bzw. das PROFINET IO-Gerät zugreifen.

7.5 PROFINET

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	

Protokoll	Verwendungsbeispiele	Eintragen der Daten in den Empfangsbereich	Kommunikationsanweisungen	Adressierungsart
ISO on TCP	CPU-zu-CPU-Kommunikation Fragmentierung und Zusammensetzung von Meldungen	Ad-hoc-Modus	Nur TRCV_C und TRCV	Weist den lokalen Geräten (aktiv) und Partnergeräten (passiv) TSAPs zu
		Protokollgesteuert	TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV	
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 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 Parameter LEN auf 65535.

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.

7.5.2 Verbindungs-IDs für die PROFINET-Anweisungen

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.

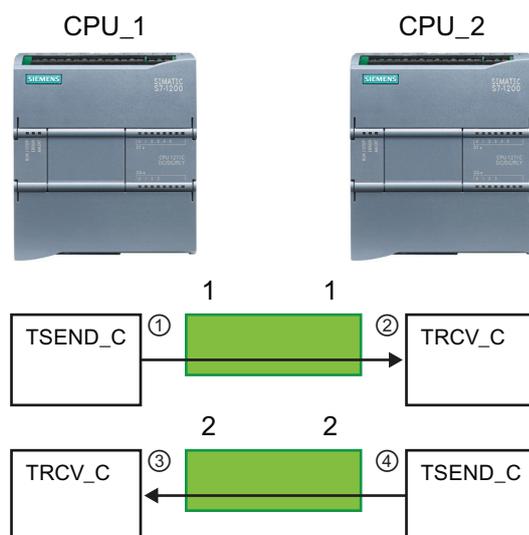
- Die Verbindungs-ID muss für die CPU eindeutig sein. Jede Verbindung, die Sie erstellen, benötigt einen anderen DB und eine andere Verbindungs-ID.
- Die lokale CPU und die Partner-CPU können dieselbe Verbindungs-ID für dieselbe Verbindung verwenden, doch die Verbindungs-IDs müssen nicht übereinstimmen.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

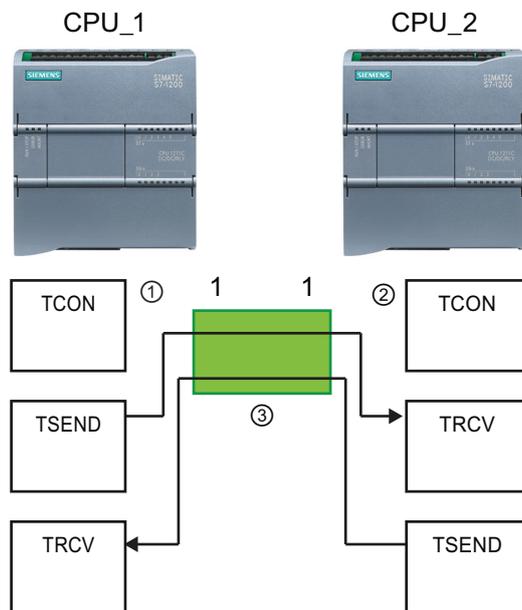
- Die Anweisung TSEND_C in CPU_1 bezieht sich auf die Anweisung TRCV_V in CPU_2 über die erste Verbindung ("Verbindungs-ID 1" bei beiden CPUs, CPU_1 und CPU_2).
- Die Anweisung TRCV_C in CPU_1 bezieht sich auf die Anweisung TSEND_C in CPU_2 über die zweite Verbindung ("Verbindungs-ID 2" bei beiden CPUs,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

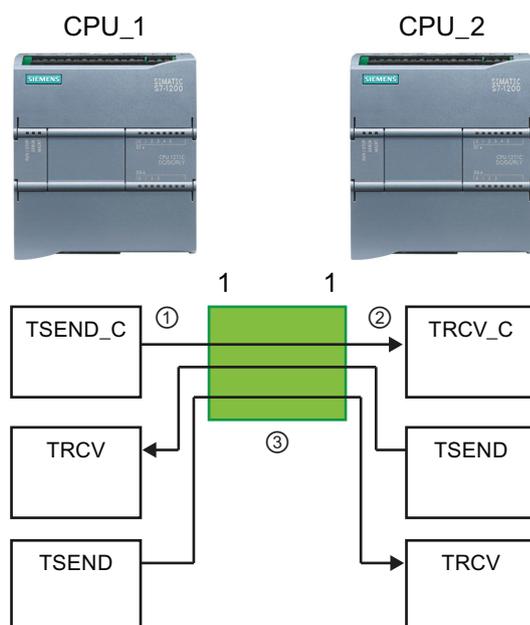
Das folgende Beispiel zeigt die Kommunikation zwischen zwei CPUs, die nur eine Verbindung zum Senden und Empfangen von Daten nutzen.

- 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.

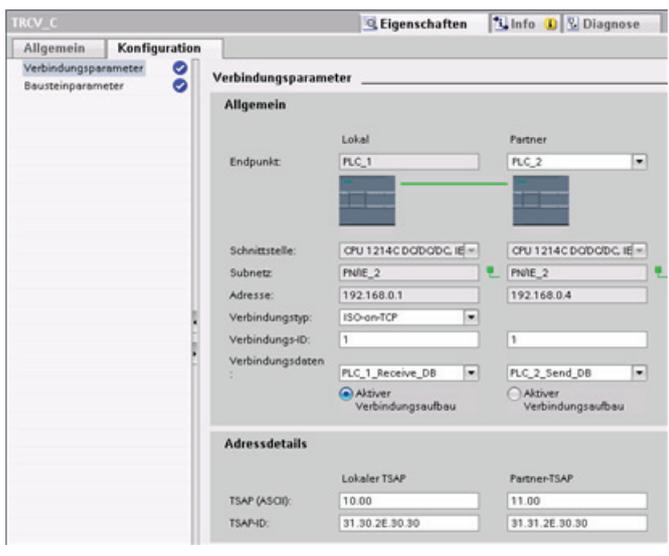


- ① TSEND_C in CPU_1 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID zu (ID = 1).
- ② TRCV_C in CPU_2 erstellt eine Verbindung und weist dieser Verbindung eine Verbindungs-ID in CPU_2 zu (ID = 1).
- ③ TSEND und TRCV in CPU_1 nutzen die von TSEND_C in CPU_1 erstellte Verbindungs-ID (ID = 1). TSEND und TRCV in CPU_2 nutzen die von TRCV_C in CPU_2 erstellte Verbindungs-ID (ID = 1).

7.5.3 Verbindungspfad zwischen lokaler und Partner-CPU konfigurieren

Im Inspektorfenster werden 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- 3 Verbindungspfad konfigurieren (über die Eigenschaften der Anweisung)

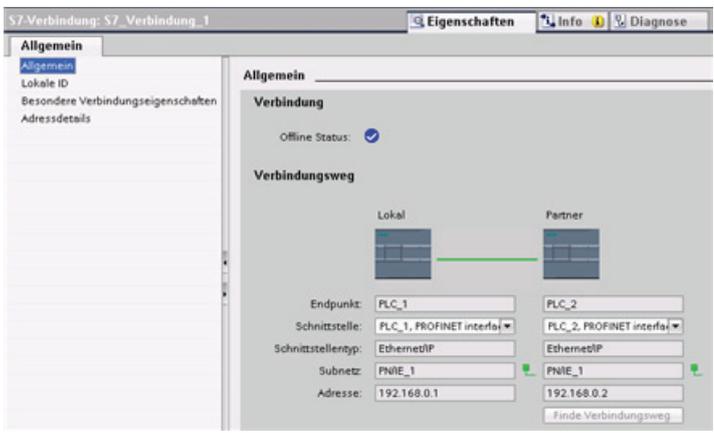
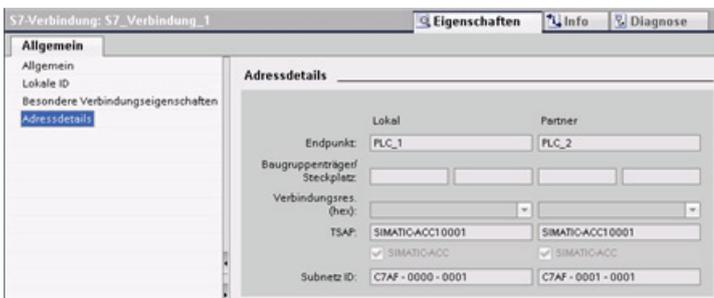
TCP, ISO-on-TCP und UDP	Verbindungseigenschaften
<p>Bei den TCP-, ISO-on-TCP- und UDP-Ethernet-Protokollen konfigurieren Sie die Verbindungen der lokalen und der Partner-CPU in den "Eigenschaften" der Anweisung (TSEND_C, TRCV_C oder TCON).</p> <p>Die Abbildung zeigt die "Verbindungseigenschaften" im 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 7-4 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 135) und im Abschnitt "S7-Kommunikation" unter "S7-Verbindung erstellen" (Seite 156) finden Sie weitere Informationen sowie eine Liste der verfügbaren Kommunikationsanweisungen.

Tabelle 7-5 Parameter für die CPU-Verbindung

Parameter	Definition																
Adresse	Zugewiesene IP-Adressen																
Allgemein	<table border="1"> <tr> <td data-bbox="331 1451 624 1489">Endpunkt</td> <td data-bbox="624 1451 1477 1489">Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td data-bbox="331 1489 624 1527">Schnittstelle</td> <td data-bbox="624 1489 1477 1527">Name der Schnittstellen</td> </tr> <tr> <td data-bbox="331 1527 624 1565">Subnetz</td> <td data-bbox="624 1527 1477 1565">Name der Subnetze</td> </tr> <tr> <td data-bbox="331 1565 624 1603">Schnittstellentyp</td> <td data-bbox="624 1565 1477 1603"><i>Nur S7-Kommunikation:</i> Typ der Schnittstelle</td> </tr> <tr> <td data-bbox="331 1603 624 1641">Verbindungstyp</td> <td data-bbox="624 1603 1477 1641">Typ des Ethernet-Protokolls</td> </tr> <tr> <td data-bbox="331 1641 624 1680">Verbindungs-ID</td> <td data-bbox="624 1641 1477 1680">ID-Nummer</td> </tr> <tr> <td data-bbox="331 1680 624 1718">Verbindungsdaten</td> <td data-bbox="624 1680 1477 1718">Datenspeicher für die lokale CPU und die Partner-CPU</td> </tr> <tr> <td data-bbox="331 1718 624 1794">Aktive Verbindung herstellen</td> <td data-bbox="624 1718 1477 1794">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 data-bbox="331 1794 624 1832">Endpunkt</td> <td data-bbox="624 1794 1477 1832"><i>Nur S7-Kommunikation:</i> Name der Partner-CPU (Empfänger)</td> </tr> <tr> <td data-bbox="331 1832 624 1899">Baugruppenträger/Steckplatz</td> <td data-bbox="624 1832 1477 1899"><i>Nur S7-Kommunikation:</i> Baugruppenträger- und Steckplatzadresse</td> </tr> <tr> <td data-bbox="331 1899 624 1964">Verbindungsressource</td> <td data-bbox="624 1899 1477 1964"><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.

7.5.4 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 TCP-, ISO-on-TCP- und UDP-Protokolle angegeben. 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.

Tabelle 7-6 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.

Byte	Parameter und Datentyp		Beschreibung
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.
12 ... 27	local_tsap_id	Array [1..16] of Byte	Komponente der lokalen Adresse der Verbindung: <ul style="list-style-type: none"> • TCP und ISO-on-TCP: lokale Port-Nr. (mögliche Werte: 1 bis 49151; empfohlene Werte: 2000...5000): <ul style="list-style-type: none"> – local_tsap_id[1] = High Byte der Portnummer in 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. Hinweis: Stellen Sie sicher, dass jeder Wert von local_tsap_id innerhalb der CPU eindeutig ist.

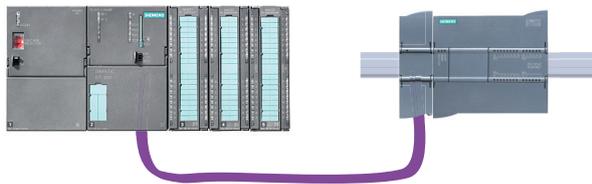
Byte	Parameter und Datentyp		Beschreibung
28 ... 33	rem_subnet_id	Array [1..6] of USInt	Dieser Parameter wird nicht verwendet.
34 ... 39	rem_staddr	Array [1..6] of USInt	Nur bei TCP und ISO-on-TCP: IP-Adresse des Partnerendpunkts. (Nicht relevant bei passiven Verbindungen.) Beispiel: Die IP-Adresse 192.168.002.003 wird in den folgenden Elementen des Arrays gespeichert: rem_staddr[1] = 192 rem_staddr[2] = 168 rem_staddr[3] = 002 rem_staddr[4] = 003 rem_staddr[5-6] = irrelevant
40 ... 55	rem_tsap_id	Array [1..16] of Byte	Komponente der Partneradresse der Verbindung: <ul style="list-style-type: none"> • TCP: Portnummer der Partner-CPU. Bereich: 1 bis 49151; empfohlene Werte: 2000 bis 5000): <ul style="list-style-type: none"> – rem_tsap_id[1] = High Byte der Portnummer in Hexadezimalnotierung; – rem_tsap_id[2] = Low Byte der Portnummer in Hexadezimalnotierung; – rem_tsap_id[3-16] = irrelevant • ISO-on-TCP: Partner-TSAP-ID: <ul style="list-style-type: none"> – rem_tsap_id[1] = B#16#E0 – rem_tsap_id[2] = Baugruppenträger und Steckplatz des Partnerendpunkts (Bits 0 bis 4: Steckplatznummer, Bits 5 bis 7: Nummer des Baugruppenträgers) – rem_tsap_id[3-16] = TSAP-Erweiterung, optional • UDP: Dieser Parameter wird nicht verwendet.
56 ... 61	next_staddr	Array [1..6] of Byte	Dieser Parameter wird nicht verwendet.
62 ... 63	spare	Word	Reserviert: W#16#0000

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. In der folgenden Abbildung ist die S7-1200 ein 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-Mastersystems fungieren.



7.6.1 Anweisungen für die dezentrale E/A

Die folgenden Anweisungen für die dezentrale Peripherie können mit PROFIBUS verwendet werden:

- Anweisung RDREC: Mit der Nummer INDEX aus einer Komponente können Sie einen Datensatz lesen.
- Anweisung WRREC: Sie können einen Datensatz mit der Nummer INDEX in eine von ID angegebene DP-Slave-Komponente übertragen.
- Anweisung RALRM: Sie können einen Alarm mit allen entsprechenden Informationen von einer DP-Slave-Komponente empfangen und diese Informationen an die Ausgangsparameter übergeben.
- Anweisung DPRD_DAT: Die CPU unterstützt bis zu 64 Byte konsistenter Daten. Konsistente Datenbereiche größer als 64 Byte müssen Sie über die Anweisung DPRD_DAT aus einem DP-Standardslave auslesen.
- Anweisung DPWR_DAT: Die CPU unterstützt bis zu 64 Byte konsistenter Daten. Konsistente Datenbereiche größer als 64 Byte müssen Sie über die Anweisung DPWR_DAT in einen DP-Standardslave schreiben.
- Anweisung DPNRM_DG: Die aktuellen Diagnosedaten eines DP-Slaves können Sie in dem Format auslesen, das in EN 50 170 Band 2, PROFIBUS, angegeben ist.

7.6.2 Konfigurationsbeispiele für PROFIBUS

Im Folgenden finden Sie Beispiele für Konfigurationen, in denen das CM 1242-5 als PROFIBUS-Slave und das CM 1243-5 als PROFIBUS-Master eingesetzt wird.

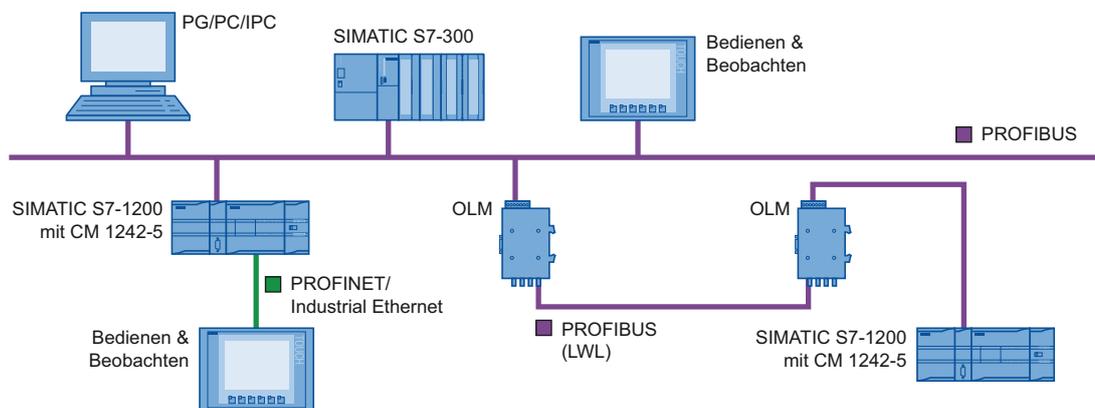


Bild 7-1 Konfigurationsbeispiel mit CM 1242-5 als PROFIBUS-Slave

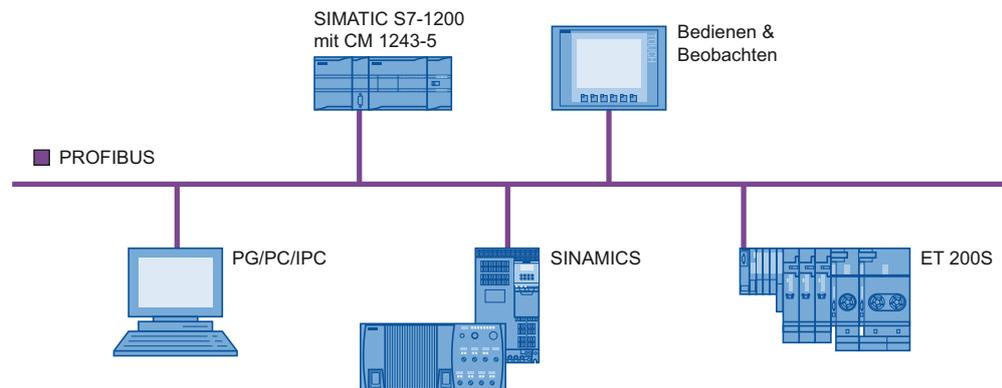


Bild 7-2 Konfigurationsbeispiel mit CM 1243-5 als PROFIBUS-Master

Anbindung der S7-1200 an PROFIBUS DP

Mithilfe der folgenden Kommunikationsmodule lässt sich die S7-1200 an ein PROFIBUS-Feldbusssystem anschließen:

- CM 1242-5
Funktion als DP-Slave
- CM 1243-5
Funktion als DP-Master Klasse 1

Bei Einbau eines CM 1242-5 und eines CM 1243-5 kann eine S7-1200 gleichzeitig folgende Funktionen ausfüllen:

- Slave eines übergeordneten DP-Mastersystems
und
- Master eines untergeordneten DP-Mastersystems

Busprotokoll

Die PROFIBUS-CMs verwenden das Protokoll PROFIBUS DP-V1.

PROFIBUS-Kommunikationspartner der S7-1200

Mit den beiden PROFIBUS-CMs wird der S7-1200 die Datenübertragung zu folgenden Kommunikationspartnern ermöglicht.

- CM 1242-5

Das CM 1242-5 (DP-Slave) kann Kommunikationspartner folgender DP-V0/V1-Master sein:

- SIMATIC S7-1200, S7-300, S7-400, S7-Modular Embedded Controller
- DP-Master-Baugruppen der dezentralen Peripherie SIMATIC ET200
- SIMATIC-PC-Stationen
- SIMATIC NET IE/PB Link
- Automatisierungsgeräte verschiedener Hersteller

- CM 1243-5

Das CM 1243-5 (DP-Master) kann Kommunikationspartner folgender DP-V0/V1-Slaves sein:

- Dezentralen Peripherie SIMATIC ET200
- S7-1200-CPU mit CM 1242-5
- S7-200-CPU mit PROFIBUS DP-Modul EM 277
- SINAMICS-Umrichter
- Antriebe und Aktoren verschiedener Hersteller
- Sensoren verschiedener Hersteller
- S7-300/400-CPU mit PROFIBUS-Schnittstelle
- S7-300/400-CPU mit PROFIBUS-CP (z. B. CP 342-5)
- SIMATIC PC-Stationen mit PROFIBUS-CP

Kommunikationsarten unter DP-V1

Folgende Kommunikationsarten stehen unter DP-V1 zur Verfügung:

- Zyklische Kommunikation (CM 1242-5 und CM 1243-5)

Beide PROFIBUS-Baugruppen unterstützen zyklische Kommunikation zur Übertragung von Prozessdaten zwischen DP-Slave und DP-Master.

Die zyklische Kommunikation führt das Betriebssystem der CPU durch. Hierfür sind keine Software-Bausteine erforderlich. Die E/A-Daten werden direkt in das Prozessabbild der CPU gelesen bzw. geschrieben.

- Azyklische Kommunikation (nur CM 1243-5)

Die DP-Master-Baugruppe unterstützt zusätzlich azyklische Kommunikation mithilfe von Software-Bausteinen:

- Für die Alarmbehandlung steht die Anweisung "RALRM" zur Verfügung.
- Für die Übertragung von Projektierungs- und Diagnosedaten stehen die Anweisungen "RDREC" und "WRREC" zur Verfügung.

Nicht unterstützte Funktionen des CM 1243-5:
SYNC/FREEZE
Get_Master_Diag

Weitere Kommunikationsdienste des CM 1243-5

Die DP-Master-Baugruppe CM 1243-5 unterstützt folgende weitere Kommunikationsdienste:

- S7-Kommunikation

- PUT-/GET-Dienste

Der DP-Master fungiert als Client und Server für Anfragen anderer S7-Steuerungen oder PCs über PROFIBUS.

- PG/OP-Kommunikation

Die PG-Funktionen ermöglichen das Laden von Projektierungsdaten und Anwenderprogrammen aus einem PG und die Übertragung von Diagnosedaten an ein PG.

Mögliche Kommunikationspartner für die OP-Kommunikation sind HMI-Panels, SIMATIC Panel-PCs mit WinCC flexible oder SCADA-Systeme, welche S7-Kommunikation unterstützen.

Projektierung und Baugruppentausch

Die Projektierung der Baugruppen, Netze und Verbindungen führen Sie in STEP 7 ab Version V11.0 durch.

Für die Projektierung in Fremdsystemen steht für das CM 1242-5 (DP-Slave) eine GSD-Datei auf der CD, welche zusammen mit der Baugruppe geliefert wird, und auf den Siemens Automation Customer Support-Seiten im Internet zur Verfügung.

Die Projektierungsdaten der PROFIBUS-CMs werden auf der jeweils lokalen CPU gespeichert. Dies ermöglicht im Ersatzteillfall einen einfachen Austausch dieser Kommunikationsbaugruppen.

Sie können maximal drei PROFIBUS-CMs pro Station projektieren, davon maximal 1 DP-Master.

Elektrische Anschlüsse

- Spannungsversorgung

- Das CM 1242-5 wird über den Rückwandbus der SIMATIC-Station gespeist.

- Das CM 1243-5 besitzt einen eigenen Anschluss für die Spannungsversorgung DC 24 V.

- PROFIBUS

Die RS485-Schnittstelle des PROFIBUS-Anschlusses ist eine 9-polige Sub-D-Buchse.

Optische PROFIBUS-Netze können Sie optional über ein Optical Bus Terminal OBT oder ein Optical Link Module OLM anschließen.

7.6.3 CM 1243-5 (DP-Master) und DP-Slave hinzufügen

Im Hardwarekatalog können Sie PROFIBUS-Module zur CPU hinzufügen. 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

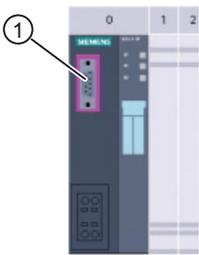
DP-Slave einfügen	Ergebnis

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.

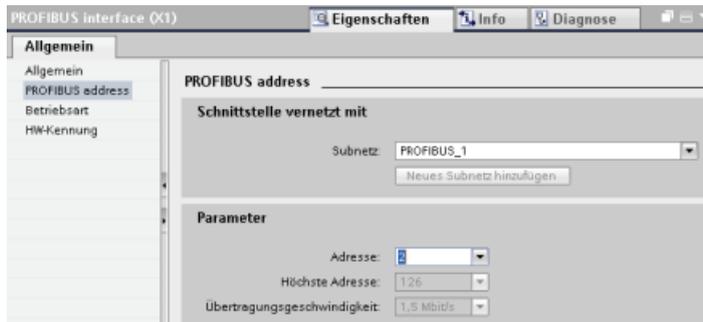


Tabelle 7- 10 Parameter für die PROFIBUS-Adresse

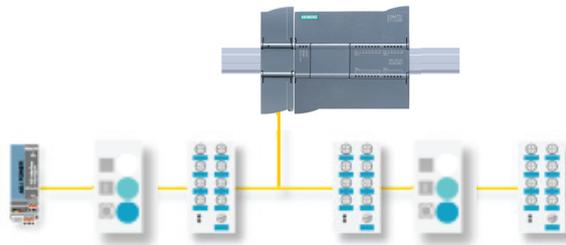
Parameter	Beschreibung	
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist. Um ein neues Subnetz anzulegen, klicken Sie auf Schaltfläche "Neues Subnetz hinzufügen". Die Voreinstellung ist "nicht verbunden". Zwei Arten von Verbindungen sind möglich: <ul style="list-style-type: none"> • Die Voreinstellung "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.
	Übertragungsgeschwindigkeit	Übertragungsgeschwindigkeit im konfigurierten PROFIBUS-Netzwerk: Die PROFIBUS-Übertragungsgeschwindigkeiten liegen zwischen 9,6 kBit/s und 12 MBit/s. Die Übertragungsgeschwindigkeit ist von den Eigenschaften der verwendeten PROFIBUS-Teilnehmer abhängig. Die Übertragungsgeschwindigkeit darf nicht größer sein als die vom langsamsten Teilnehmer unterstützte Geschwindigkeit. Die Übertragungsgeschwindigkeit wird normalerweise für den Master im PROFIBUS-Netzwerk festgelegt, wobei alle DP-Slaves automatisch die gleiche Übertragungsgeschwindigkeit nutzen (Auto-Baud).

7.7 ASi

Der S7-1200 CM 1243-2 AS-i Master ermöglicht die Anbindung eines AS-i-Netzwerks an eine S7-1200 CPU.

Die Aktor-/Sensorschnittstelle bzw. AS-i ist ein Netzwerkverbindingssystem für einen Master auf der niedrigsten Stufe in einem Automatisierungssystem. Das CM 1243-2 dient als AS-i-Master im Netzwerk. Mit nur einem AS-i-Kabel können Sensoren und Aktoren (AS-i-Slavegeräte) über das CM 1243-2 mit der CPU verbunden werden. Das CM 1243-2 übernimmt sämtliche Koordinationsaufgaben im AS-i-Netzwerk und gibt die Daten und Statusinformationen von den Aktoren und Sensoren über die E/A-Adressen, die dem CM 1243-2 zugewiesen sind, an die CPU weiter. Je nach Art des Slaves können Sie auf binäre oder analoge Werte zugreifen. Die AS-i-Slaves stellen die Eingangs- und Ausgangskanäle des AS-i-Systems dar und sind erst nach Aufruf durch das CM 1243-2 aktiv.

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 CM 1243-2 AS-i-Mastermodul und AS-i-Slave hinzufügen

Im Hardwarekatalog können Sie CM1243-2 AS-i-Mastermodule zur CPU hinzufügen. 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- 11 Ein AS-i CM 1243-2 AS-i-Mastermodul in die Gerätekonfiguration einfügen

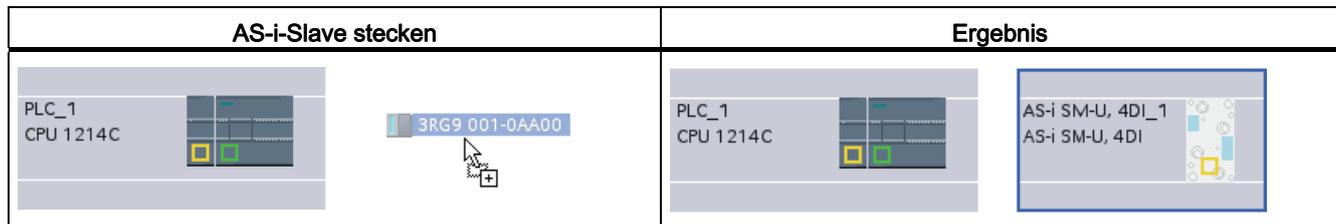
Modul	Modul auswählen	Modul einsetzen	Ergebnis
CM 1243-2 AS-i- Master	<ul style="list-style-type: none"> ▼ Kommunikationsmodule ▶ PROFIBUS ▶ FunItcp-FunIt ▼ AS-Interface ▼ CM 1243-2 3PP-7243-2AA30-0XB0 		

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

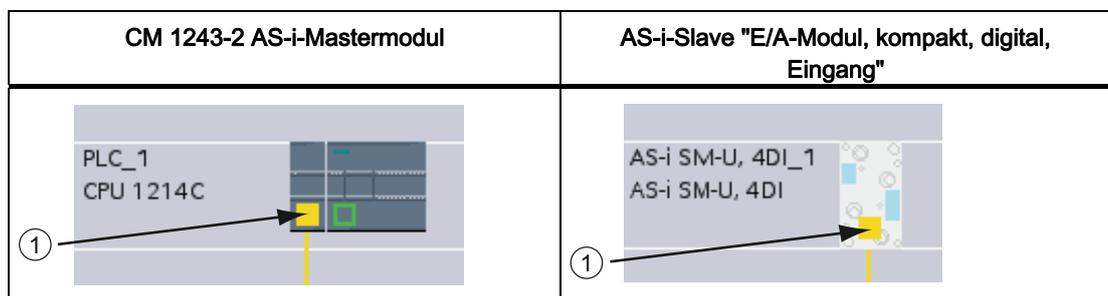


7.7.2 Dem CM 1243-2 AS-i-Mastermodul und dem AS-i-Slave AS-i-Adressen zuweisen

AS-i-Schnittstelle konfigurieren

Um Parameter für die AS-i-Schnittstellen zu konfigurieren, klicken Sie auf das gelbe AS-i-Feld am CM 1243-2 AS-i-Mastermodul. Daraufhin wird die AS-i-Schnittstelle im Register "Eigenschaften" im Inspektorfenster angezeigt. Die AS-i-Slaveschnittstelle wird auf dieselbe Weise konfiguriert.

Tabelle 7- 13 CM 1243-2 AS-i-Mastermodul und AS-i-Slaveschnittstelle mit Merkmalen "E/A-Modul, kompakt, digital, Eingang" konfigurieren



① AS-i-Port

AS-i-Adressen zuweisen

In einem AS-i-Netzwerk wird jedem Gerät eine AS-i-Adresse 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. 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.

Ein neuer Slave, dem noch keine Adresse zugewiesen wurde, hat immer die Adresse 0. Er wird vom Master als neuer Slave ohne Adresszuweisung erkannt und wird erst nach Zuweisung einer Adresse in die normale Kommunikation einbezogen.

Im Fenster "Eigenschaften" wählen Sie den Konfigurationseintrag "AS-i-Adresse". STEP 7 zeigt den Konfigurationsdialog für die AS-i-Adresse an, in dem Sie die AS-i-Adresse des Geräts zuweisen.

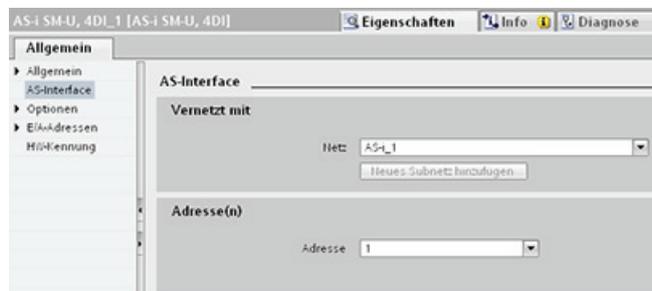


Tabelle 7- 14 Parameter für die AS-i-Adresse

Parameter	Beschreibung	
Subnetz	Name des Subnetzes, mit dem das Gerät verbunden ist.	
Parameter	Adresse	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
	Übertragungsgeschwindigkeit	Die Übertragungsgeschwindigkeit im konfigurierten AS-i-Netzwerk beträgt 10 ms.

7.8 S7-Kommunikation

7.8.1 Anweisungen GET und PUT

Mit den Anweisungen GET und PUT können Sie mit S7-CPU's über PROFINET- und PROFIBUS-Verbindungen kommunizieren.

- Zugriff auf Daten in einer S7-300/400-CPU: Eine S7-1200-CPU kann entweder absolute Adressen oder symbolische Namen verwenden, um Variablen in einer S7-300/400-CPU anzusprechen. Auf Datentypen des entfernten Kommunikationspartners, die von der aufrufenden S7-1200-CPU nicht unterstützt werden, kann nur als Byte-Arrays zugegriffen werden. Auf den S7-300 Datentyp-DT beispielsweise wird als ein Array aus 8 Bytes zugegriffen.
- Zugriff auf Daten in einem Standard-DB: Eine S7-1200-CPU kann entweder absolute Adressen oder symbolische Namen 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 nur symbolische Namen verwenden, um DB-Variablen in einem optimierten DB einer dezentralen S7-CPU anzusprechen. Nur Variablen der ersten Schachtelungsebene werden unterstützt. Dies beinhaltet Variablen, die in einem optimierten globalen DB auf DB-Ebene deklariert sind. Komponenten von Strukturen optimierter DBs oder Elemente aus Arrays können nicht angesprochen werden.

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

Hinweis

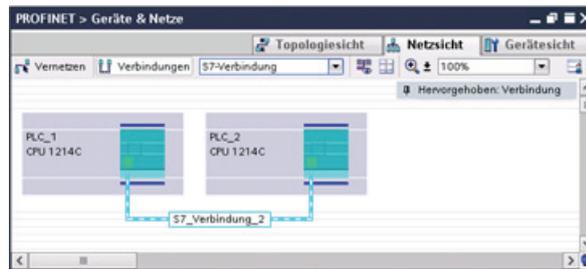
Um Datenkonsistenz sicherzustellen, prüfen Sie stets, ob die Anweisung beendet wurde (NDR = 1 bei GET bzw. DONE = 1 bei PUT), bevor Sie auf die Daten zugreifen oder eine weitere Lese- oder Schreibweisung ausführen.

7.8.2 S7-Verbindung erstellen

Die ausgewählte Verbindungsart baut eine Kommunikationsverbindung zu einer Partnerstation auf. Die Verbindung wird eingerichtet, aufgebaut und automatisch überwacht.

Im Portal "Geräte & Netze" können Sie in der "Netzansicht" die Geräte in Ihrem Projekt vernetzen. Klicken Sie zunächst auf das Register "Verbindungen" und wählen Sie dann über die Klappliste rechts den Verbindungstyp aus (z. B. eine S7-Verbindung). Klicken Sie auf das grüne Feld (PROFINET) auf dem ersten Gerät und ziehen Sie eine Linie zum PROFINET-Feld auf dem zweiten Gerät. Lassen Sie die Maustaste los. Damit ist Ihre PROFINET-Verbindung hergestellt.

Ausführliche Informationen finden Sie unter "Erstellen einer Netzwerkverbindung" (Seite 132).

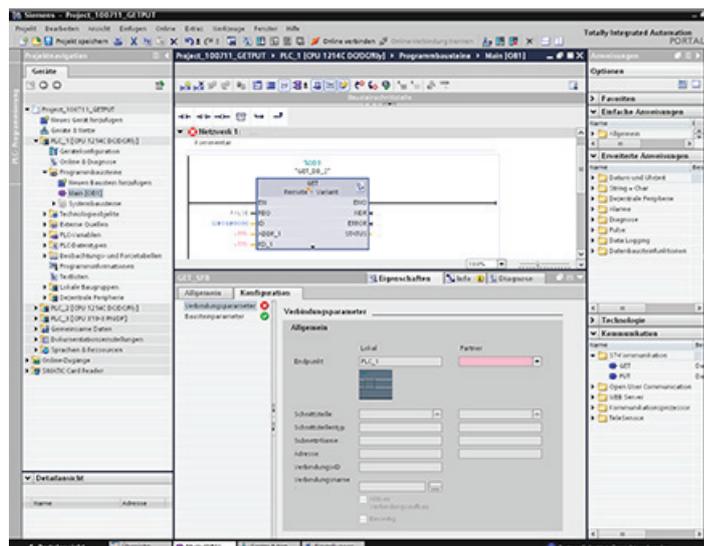


Klicken Sie auf die Schaltfläche "Hervorgehoben: Verbindung", um den Konfigurationsdialog "Eigenschaften" der Kommunikationsanweisung aufzurufen.

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.

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

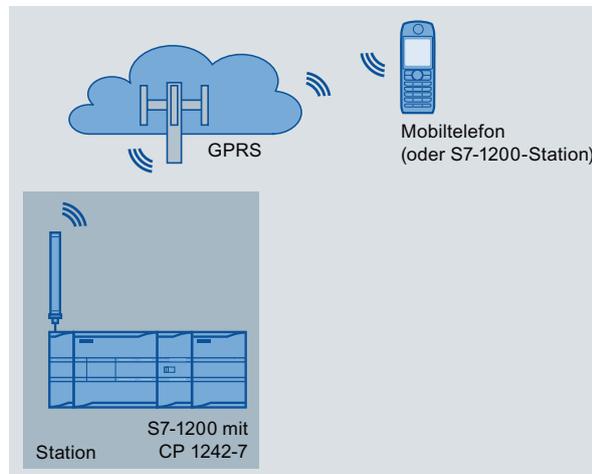


Bild 7-3 SMS-Versand einer S7-1200-Station

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

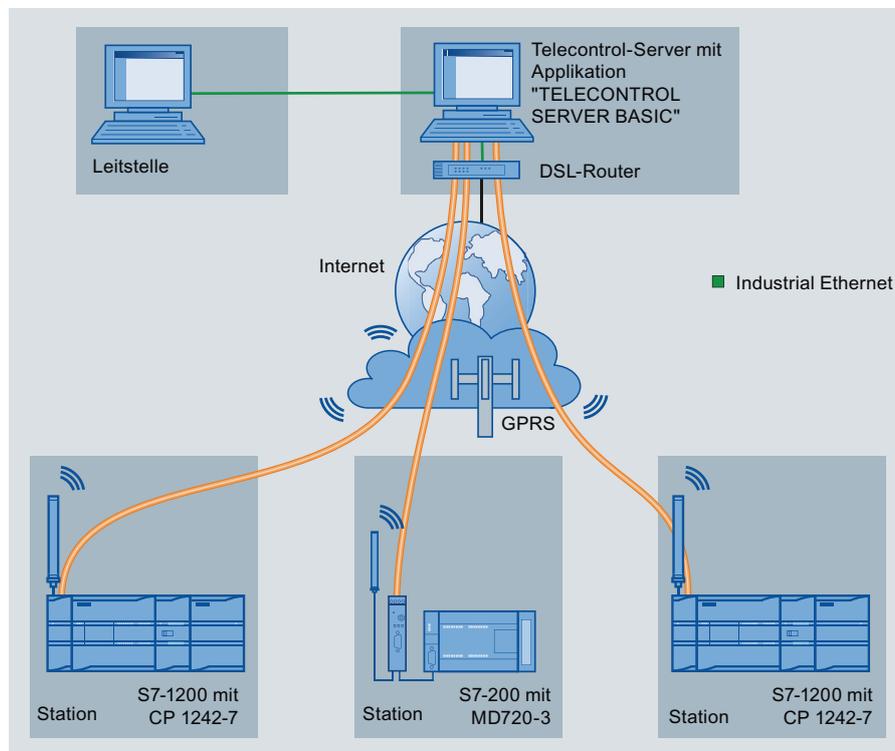


Bild 7-4 Kommunikation von S7-1200-Stationen mit einer Zentrale

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

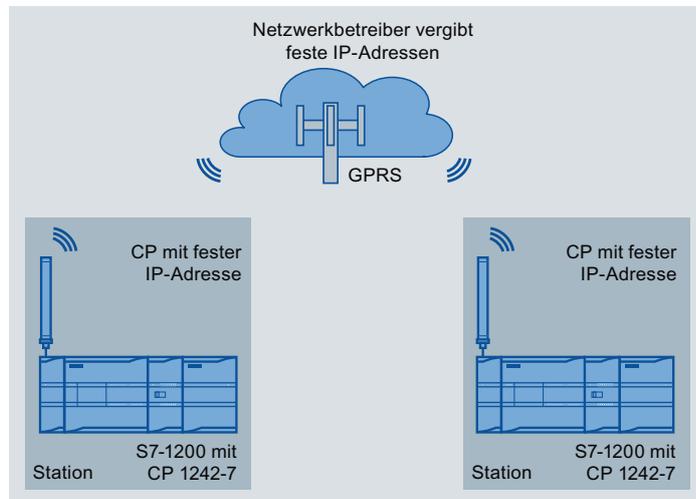


Bild 7-5 Direkte Querkommunikation von zwei S7-1200-Stationen

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

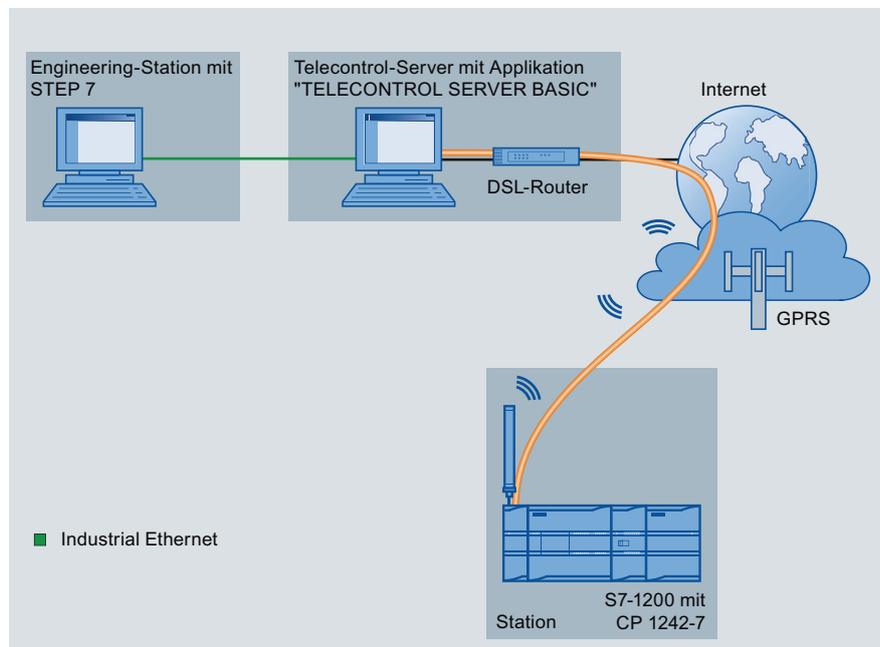


Bild 7-6 TeleService über GPRS

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 Netzbetreibers 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 7-7 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

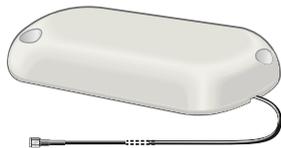


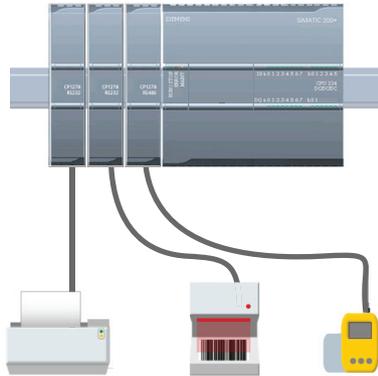
Bild 7-8 Flachantenne ANT794-3M

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

Die CPU unterstützt das PtP-Protokoll für die zeichenbasierte serielle Kommunikation, bei dem die Benutzeranwendung das bevorzugte Protokoll definiert und implementiert.



PtP bietet eine Vielzahl an Möglichkeiten:

- Direktes Senden von Informationen an ein externes Gerät, z. B. einen Drucker
- Empfangen von Informationen von Geräten wie z. B. Strichcodelesern, RFID-Lesern, Kamera- oder Überwachungssystemen anderer Hersteller und vielen anderen Geräteearten
- Senden und Empfangen von Daten an Geräte bzw. von Geräten wie z. B. GPS-Geräten, Kamera- oder Überwachungssystemen anderer Hersteller oder Funkmodems

PtP ist eine Form serieller Kommunikation, die verschiedene Baudraten und Paritäten unterstützt. STEP 7 bietet Anweisungen für das USS-Protokoll für Antriebe (nur RS485-Schnittstelle) sowie für das Modbus-RTU-Master- und Modbus-RTU-Slave-Protokoll.

7.10.1 Arbeiten mit den RS232- und RS485-Kommunikationsschnittstellen

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

- CM 1241 RS232
- CM 1241 RS485
- 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".

Die RS232- und RS485-Kommunikationsschnittstellen haben die folgenden Eigenschaften:

- Potentialgetrennter Anschluss
- Unterstützung von Punkt-zu-Punkt-Protokollen
- Konfiguration und Programmierung über erweiterte Anweisungen und Bibliotheksfunktionen
- 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 ein CB oder CMs und adressiert diese. Die Diagnose-LED beginnt, grün zu blinken. Das bedeutet, dass die CPU das CM oder CB adressiert, jedoch noch keine Konfiguration zugewiesen hat. Die CPU lädt die Konfiguration in die konfigurierten CMs und das CB, wenn das Programm in die CPU geladen wird. Nach dem Laden in die CPU muss die Diagnose-LED am Kommunikationsmodul oder Kommunikationsboard 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.

7.10.2 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.

7.10.3 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. 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.

7.10.4 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.

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.

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 universalen 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. Sie passen auch die Parameter für die Selbsteinstellung an ("Erstoptimierung" während des Anlaufs oder manuelle "Feineinstellung"), um den PID-Regler in Betrieb zu nehmen.

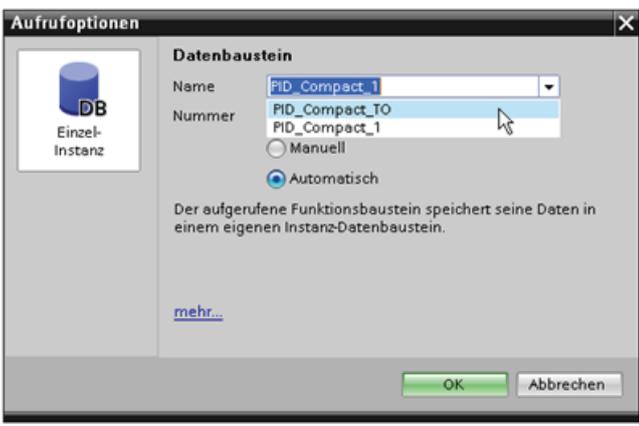
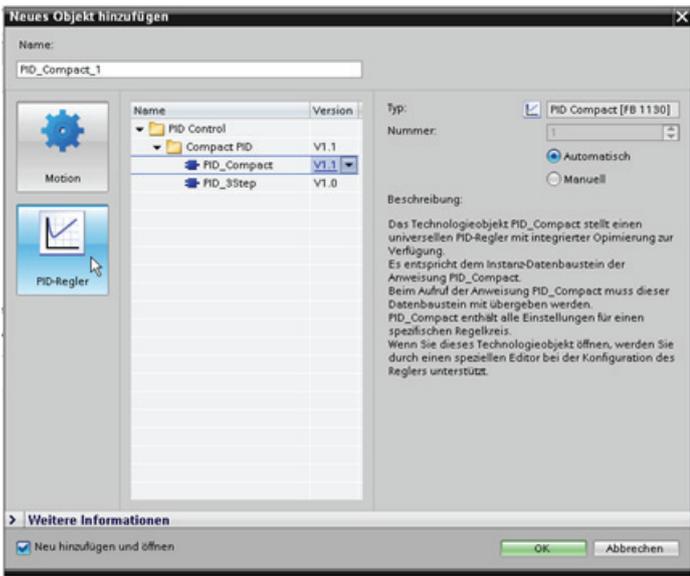
Tabelle 8- 1 Anweisung PID und Technologieobjekt einfügen

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.

Nach dem Einfügen der PID-Anweisung und dem Erstellen des Technologieobjekts sowie des Instanz-DBs konfigurieren Sie die Parameter für das Technologieobjekt.



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>	

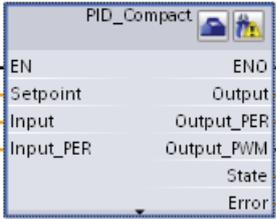
8.2 Operation PID_Compact

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)

Tabelle 8-3 Anweisung PID_Compact

KOP/FUP	SCL	Beschreibung
	<pre>"PID_Compact_1" (Setpoint:=_real_in_, Input:=_real_in_, Input_PER:=_word_in_, ManualEnable:=_bool_in_, ManualValue:=_real_in_, Reset:=_bool_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=>_dword_out_);</pre>	<p>PID_Compact bietet einen PID-Regler mit Selbsteinstellung für den Automatik- und Handbetrieb. PID_Compact ist ein PID1-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	Beschreibung
Setpoint	IN	Real
Input	IN	Real

Sollwert des PID-Reglers im Automatikbetrieb. Standardwert: 0.0
 Prozesswert. Standardwert: 0.0
 Sie müssen außerdem sPid_Cmpt.b_Input_PER_On = FALSCH setzen.

Parameter und Datentyp		Datentyp	Beschreibung
Input_PER	IN	Word	Analoger Prozesswert (optional). Standardwert: W#16#0 Sie müssen außerdem sPid_Cmpt.b_Input_PER_On = WAHR setzen.
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. Standardwert: FALSCH <ul style="list-style-type: none"> Bei der Flanke von FALSCH nach WAHR wechselt der PID-Regler in den Handbetrieb. State = 4 und sRet.i_Mode ändern sich nicht. Bei der Flanke von WAHR nach FALSCH wechselt der PID-Regler in die letzte aktive Betriebsart und nach State = sRet.i_Mode.
ManualValue	IN	Real	Prozesswert für Handbetrieb. Standardwert: 0.0
Reset	IN	Bool	Startet den Regler neu. Standardwert: FALSCH Wenn Reset = WAHR, gilt Folgendes: <ul style="list-style-type: none"> Inaktiver Betriebszustand Eingangswert = 0 Integralanteil des Prozesswerts = 0 Zwischenwerte des Systems werden zurückgesetzt (PIDParameter wird gespeichert)
ScaledInput	OUT	Real	Skalierter Prozesswert. Standardwert: 0.0
Output ¹	OUT	Real	Ausgangswert. Standardwert: 0.0
Output_PER ¹	OUT	Word	Analoger Ausgangswert. Standardwert: W#16#0
Output_PWM ¹	OUT	Bool	Ausgangswert für die Impulsdauermodulation. Standardwert: FALSCH
SetpointLimit_H	OUT	Bool	Sollwert oberer Grenzwert. Standardwert: FALSCH Wenn SetpointLimit_H = WAHR, ist der absolute obere Grenzwert des Sollwerts erreicht. Standardwert: FALSCH
SetpointLimit_L	OUT	Bool	Sollwert unterer Grenzwert. Standardwert: FALSCH Wenn SetpointLimit_L = WAHR, ist der absolute untere Grenzwert des Sollwerts erreicht. Standardwert: FALSCH
InputWarning_H	OUT	Bool	Wenn InputWarning_H = WAHR, hat der Prozesswert die obere Warngrenze erreicht oder überschritten. Standardwert: FALSCH
InputWarning_L	OUT	Bool	Wenn InputWarning_L = WAHR, hat der Prozesswert die untere Warngrenze erreicht. Standardwert: FALSCH
State	OUT	Int	Aktuelle Betriebsart des PID-Reglers. Standardwert: 0 Mit sRet.i_Mode wechseln Sie die Betriebsart. <ul style="list-style-type: none"> State = 0: Inaktiv State = 1: Erstopptimierung State = 2: Manuelle Feineinstellung State = 3: Automatikbetrieb State = 4: Handbetrieb
Error	OUT	DWord	Fehlermeldung. Standardwert: DW#16#0000 (kein Fehler)

¹ Die Ausgänge der Parameter Output, Output_PER und Output_PWM können parallel verwendet werden.

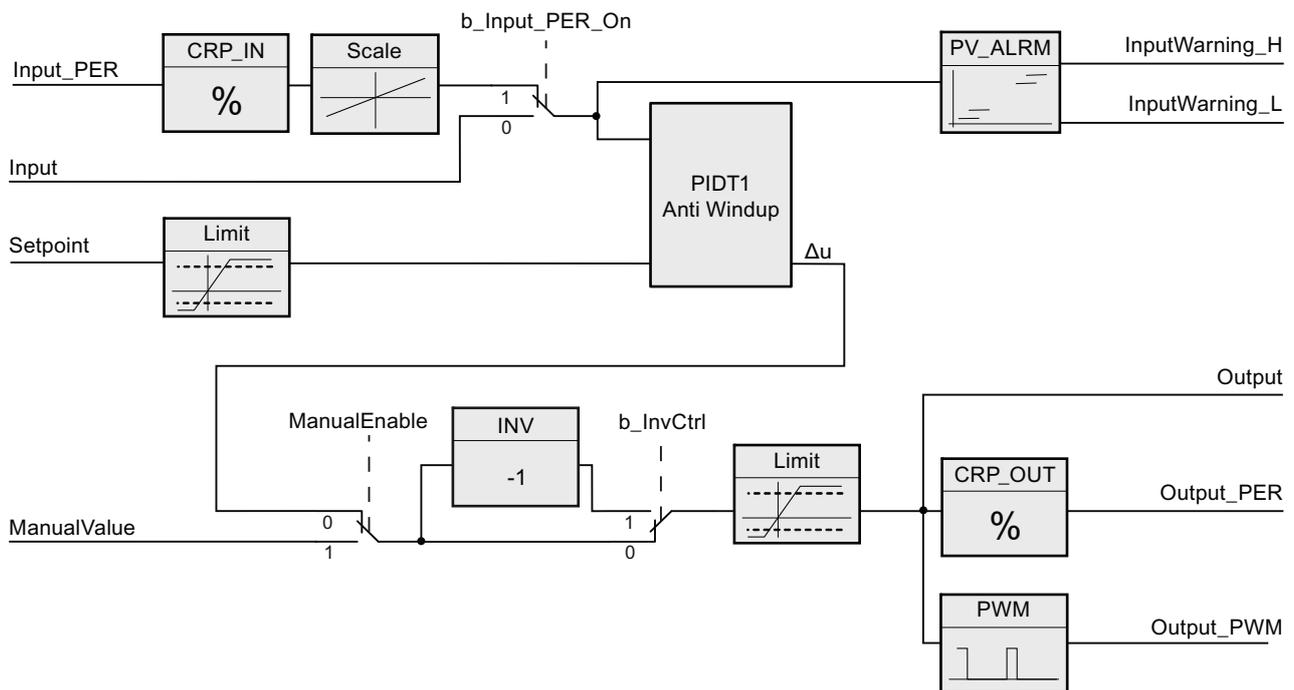


Bild 8-1 Funktionsweise des Reglers PID_Compact

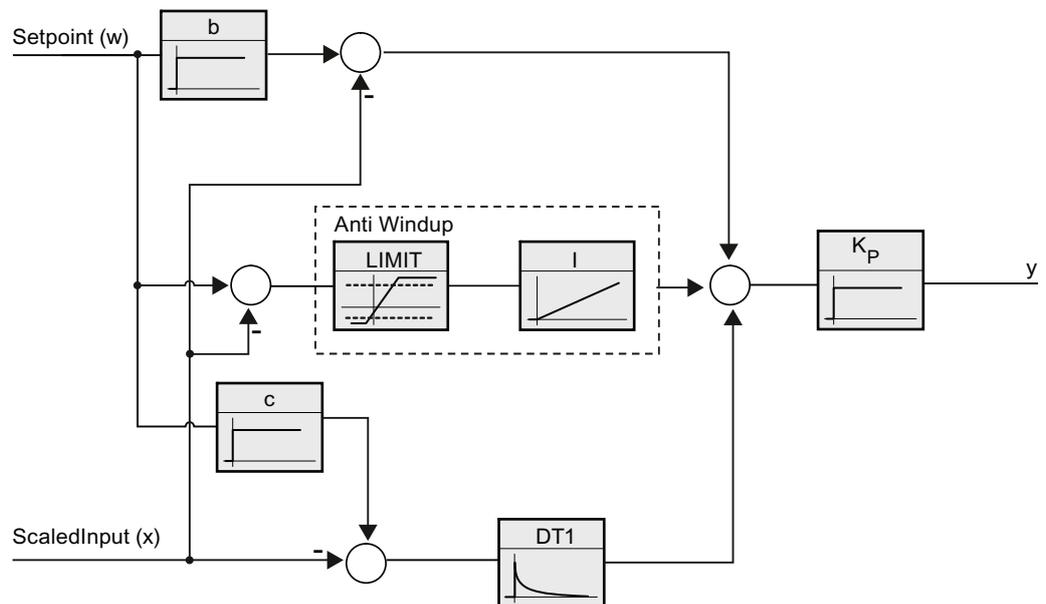


Bild 8-2 Funktionsweise des Reglers PID_Compact als PIDT1-Regler mit Anti-Windup

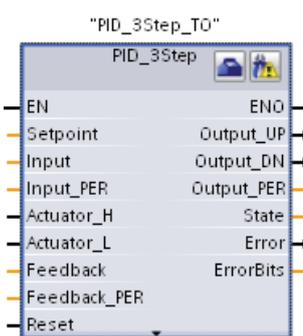
8.3 Anweisung PID_3STEP

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)

Tabelle 8-5 Anweisung PID_3Step

KOP/FUP	SCL	Beschreibung
 <p>The screenshot shows a control block titled "PID_3Step" with the following ports: EN, ENO, Setpoint, Output_UP, Input, Output_DN, Input_PER, Output_PER, Actuator_H, State, Actuator_L, Error, Feedback, ErrorBits, Feedback_PER, and Reset.</p>	<pre>"PID_3Step_1" (SetpoInt:=_real_in_, Input:=_real_in_, ManualValue:=_real_in_, Feedback:=_real_in_, InputPer:=_word_in_, FeedbackPer:=_word_in_, ManualEnable:=_bool_in_, ManualUP:=_bool_in_, ManualDN:=_bool_in_, ActuatorH:=_bool_in_, ActuatorL:=_bool_in_, Reset:=_bool_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_);</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 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_3Step_1" der Name des Instanz-DBs.

Tabelle 8-6 Datentypen für die Parameter

Parameter und Datentyp		Datentyp	Beschreibung
Setpoint	IN	Real	Sollwert des PID-Reglers im Automatikbetrieb. Standardwert: 0.0
Input	IN	Real	Prozesswert. Standardwert: 0.0 Sie müssen außerdem Config.InputPEROn = FALSCH setzen.
Input_PER	IN	Word	Analoger Prozesswert (optional). Standardwert: W#16#0 Sie müssen außerdem Config.InputPEROn = WAHR setzen.
ManualEnable	IN	Bool	Aktiviert oder deaktiviert den Handbetrieb. Standardwert: FALSCH <ul style="list-style-type: none"> Bei der Flanke von FALSCH nach WAHR wechselt der PID-Regler in den Handbetrieb, State = 4 und Retain.Mode ändern sich nicht. Bei der Flanke von WAHR nach FALSCH wechselt der PID-Regler in die letzte aktive Betriebsart und nach State = Retain.Mode.
ManualUP	IN	Bool	Im Handbetrieb wird das Ventil durch jede steigende Flanke um 5% des Gesamtstellbereichs geöffnet, bzw. für die Dauer der minimalen Motorbetätigungszeit. ManualUP wird nur ausgewertet, wenn Sie OutputPer verwenden und wenn eine zurückgemeldete Position verfügbar ist. Standardwert: FALSCH <ul style="list-style-type: none"> Wenn Output_PER FALSCH ist, schaltet der manuelle Eingang Output_UP für den Zeitraum ein, der einer Bewegung von 5 % des Geräts entspricht. Wenn Config.ActuatorEndStopOn WAHR ist, dann wird Output_UP nicht eingeschaltet, wenn Actuator_H WAHR ist.
ManualDN	IN	Bool	Im Handbetrieb wird das Ventil durch jede steigende Flanke um 5 % des Gesamtstellbereichs geschlossen, bzw. für die Dauer der minimalen Motorbetätigungszeit. ManualDN wird nur ausgewertet, wenn Sie OutputPer verwenden und wenn eine zurückgemeldete Position verfügbar ist. Standardwert: FALSCH <ul style="list-style-type: none"> Wenn Output_PER FALSCH ist, schaltet der manuelle Eingang Output_DN für den Zeitraum ein, der einer Bewegung von 5 % des Geräts entspricht. Wenn Config.ActuatorEndStopOn WAHR ist, dann wird Output_DN nicht eingeschaltet, wenn Actuator_L WAHR ist.
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. Standardwert: 0.0
Feedback	IN	Real	Zurückgemeldete Position des Ventils. Standardwert: 0.0 Um Feedback zu verwenden, setzen Sie Config.FeedbackPerOn = FALSE.

Parameter und Datentyp		Datentyp	Beschreibung
Feedback_PER	IN	Word	Zurückgemeldeter Analogwert der Ventilposition. Standardwert: W#16#0 Um Feedback_PER zu verwenden, setzen Sie Config.FeedbackPerOn = TRUE. Feedback_PER wird mit folgenden Parametern skaliert: <ul style="list-style-type: none"> • Config.FeedbackScaling.LowerPointIn • Config.FeedbackScaling.UpperPointIn • Config.FeedbackScaling.LowerPointOut • Config.FeedbackScaling.UpperPointOut
Actuator_H	IN	Bool	Wenn Actuator_H = TRUE, befindet sich das Ventil am oberen Endpunkt und wird nicht weiter in diese Richtung bewegt. Standardwert: FALSCH
Actuator_L	IN	Bool	Wenn Actuator_L = TRUE, befindet sich das Ventil am unteren Endpunkt und wird nicht weiter in diese Richtung bewegt. Standardwert: FALSCH
Reset	IN	Bool	Startet den PID-Regler neu. Standardwert: FALSCH Wenn Reset = WAHR: <ul style="list-style-type: none"> • Betriebszustand "Inaktiv" • Eingangswert = 0 • Zwischenwerte des Reglers werden zurückgesetzt. (PID-Parameter werden gespeichert.)
ScaledInput	OUT	Real	Skalierter Prozesswert
ScaledFeedback	OUT	Real	Skalierte Ventilposition
Output_PER	OUT	Word	Analoger Ausgangswert. Wenn Config.OutputPerOn = WAHR, dann wird Output_PER ausgewertet.
Output_UP	OUT	Bool	Digitaler Ausgangswert zum Öffnen des Ventils. Standardwert: FALSCH Wenn Config.OutputPerOn = FALSCH, dann wird der Parameter Output_UP ausgewertet.
Output_DN	OUT	Bool	Digitaler Ausgangswert zum Schließen des Ventils. Standardwert: FALSCH Wenn Config.OutputPerOn = FALSCH, dann wird der Parameter Output_DN ausgewertet.
SetpointLimitH	OUT	Bool	Sollwert oberer Grenzwert. Standardwert: FALSCH Wenn SetpointLimitH = WAHR, ist der absolute obere Grenzwert des Sollwerts erreicht. In der CPU ist der Sollwert auf den konfigurierten absoluten oberen Grenzwert des Istwerts begrenzt.
SetpointLimitL	OUT	Bool	Sollwert unterer Grenzwert. Standardwert: FALSCH Wenn SetpointLimitL = WAHR, ist der absolute untere Grenzwert des Sollwerts erreicht. In der CPU ist der Sollwert auf den konfigurierten absoluten unteren Grenzwert des Istwerts begrenzt.
InputWarningH	OUT	Bool	Wenn InputWarningH = WAHR, hat der Eingangswert die obere Warngrenze erreicht oder überschritten. Standardwert: FALSCH
InputWarningL	OUT	Bool	Wenn InputWarningL = WAHR, hat der Eingangswert die untere Warngrenze erreicht oder überschritten. Standardwert: FALSCH

Parameter und Datentyp	Datentyp	Beschreibung
State	OUT	Int Aktuelle Betriebsart des PID-Reglers. Standardwert: 0 Mit Retain.Mode wechseln Sie die Betriebsart: <ul style="list-style-type: none"> • State = 0: Inaktiv • State = 1: Erstopptimierung • State = 2: Manuelle Feineinstellung • State = 3: Automatikbetrieb • State = 4: Handbetrieb • State = 5: Sicherheitsbetrieb • State = 6: Ausgangswertmessung • State = 7: Überwachung im Sicherheitsmodus mit aktivem Auslöser • State = 8: Überwachung im inaktiven Modus mit aktivem Auslöser
Error	OUT	Bool Wenn Error = WAHR, steht mindestens eine Fehlermeldung an. Standardwert: FALSCH
ErrorBits	OUT	DWord Fehlermeldung. Standardwert: DW#16#0000 (kein Fehler)

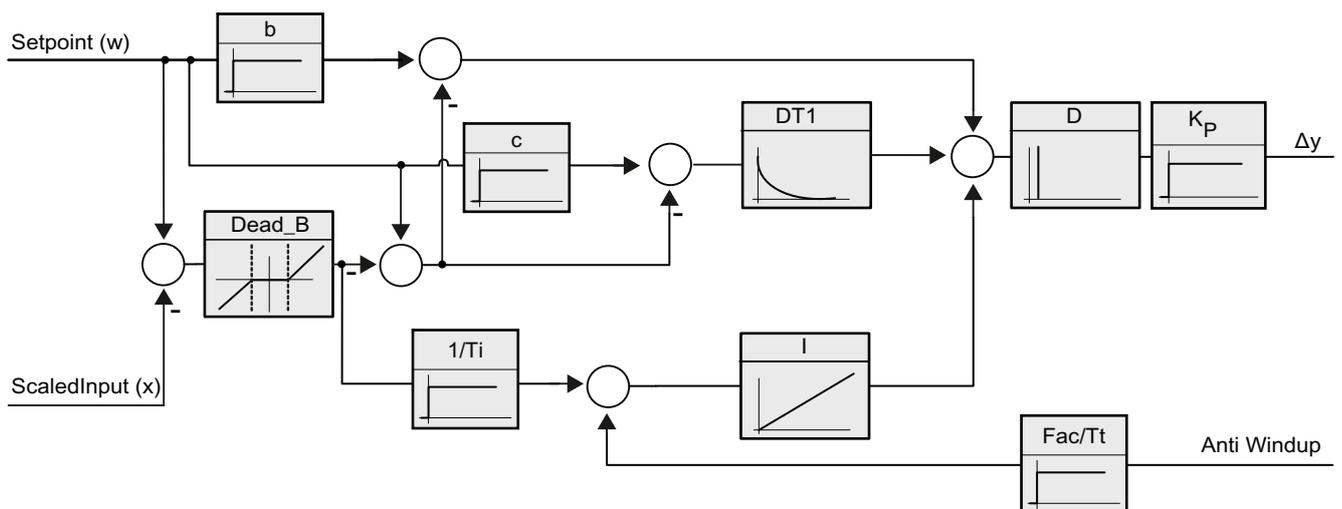


Bild 8-3 Funktionsweise des Reglers PID_3Step als PIDT1-Regler mit Anti-Windup

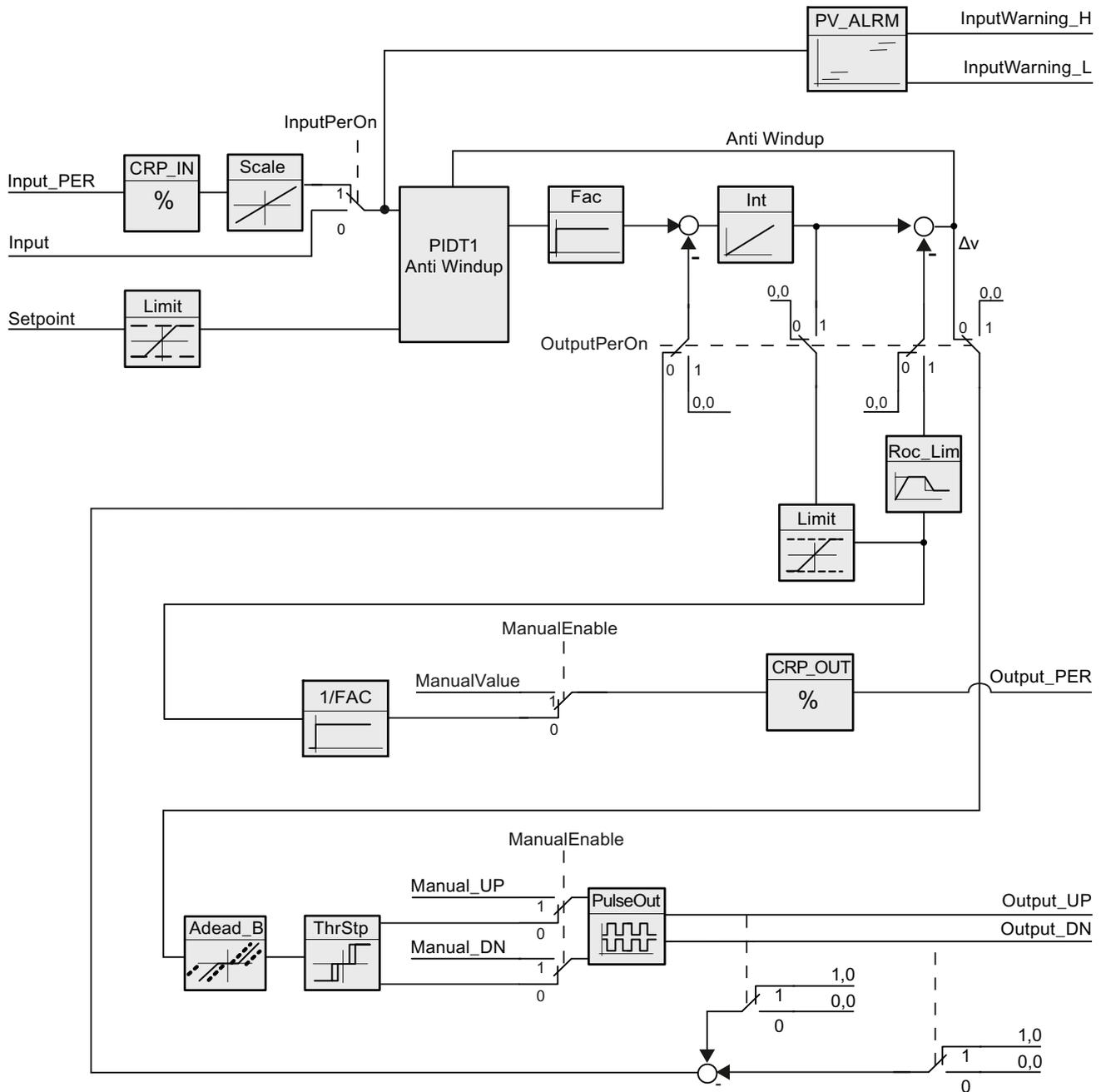


Bild 8-4 Funktionsweise des Reglers PID_3Step ohne Positionsrückmeldung

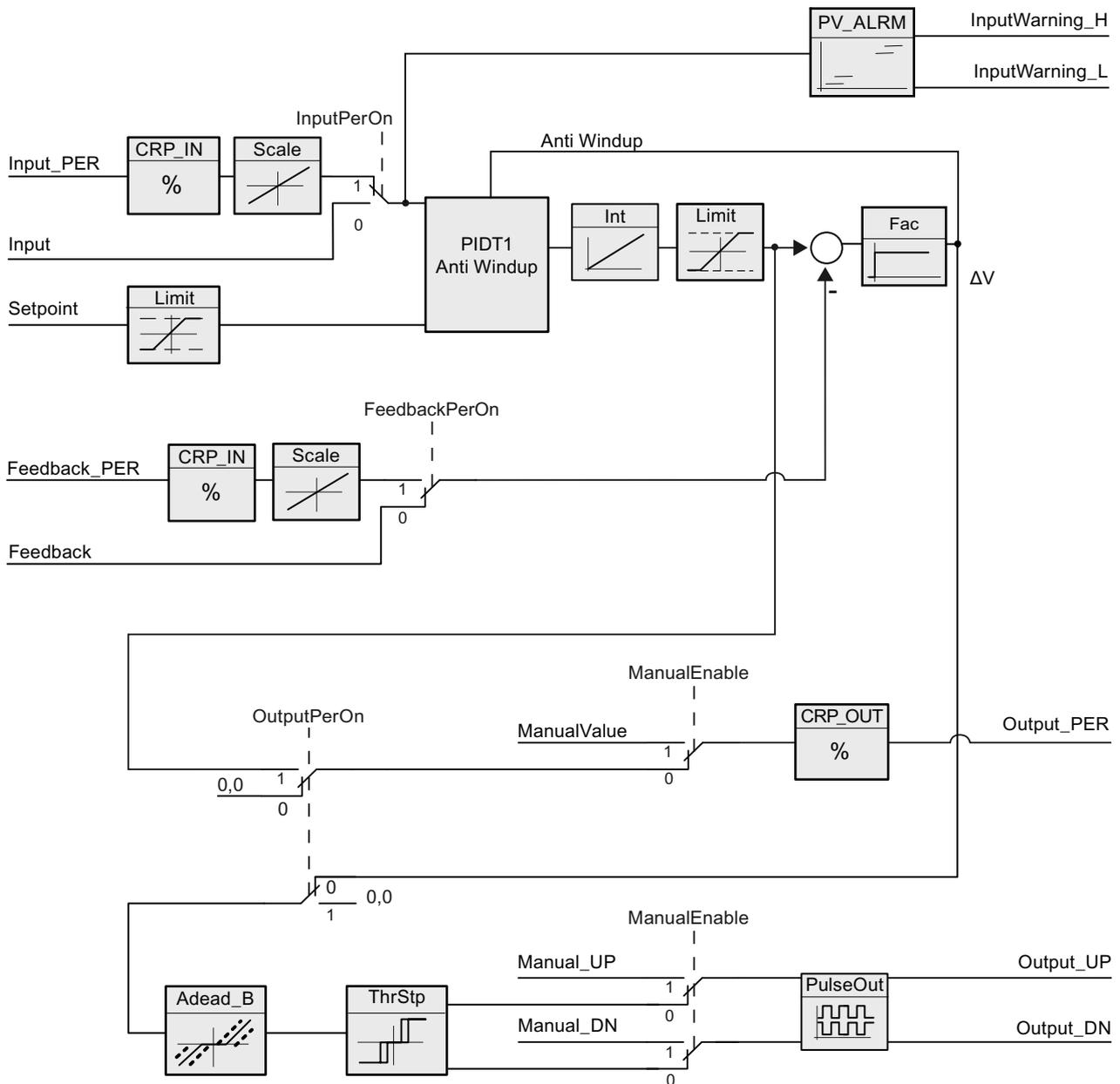


Bild 8-5 Funktionsweise des Reglers PID_3Step mit aktivierter Positionsrückmeldung

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- 7 ErrorBit-Parameter

ErrorBit (DW#16#...)	Beschreibung
0000	Kein Fehler
0001	Der Parameter Input liegt außerhalb der Grenzen des Prozesswerts. <ul style="list-style-type: none"> • Input > Config.InputUpperLimit oder • Input < Config.InputLowerLimit
0002	Ungültiger Wert für den Parameter Input_PER. Ermitteln Sie, ob am Analogeingang ein Fehler vorliegt.
0004	Feineinstellung: Schwingung des Prozesswerts (Eingang) konnte nicht erhalten werden.
0008	Erstoptimierung: Der Prozesswert (Eingang) ist zu nah am Sollwert. Starten Sie die Feineinstellung.
0010	Der Sollwert darf während der Erstoptimierung am Betriebspunkt nicht geändert werden.
0020	Für die Erstoptimierung ist Automatikbetrieb eingestellt, dies ist während der Feineinstellung nicht zulässig.
0040	Erstoptimierung: Der Sollwert ist zu nah an den Grenzen des Ausgangswerts.
0080	Erstoptimierung: Inkorrekte Konfiguration der Grenzen des Ausgangswerts.
0100	Fehler während der Feineinstellung: führte zu ungültigen Parametern.
0200	Ungültiger Wert für den Parameter Input: <ul style="list-style-type: none"> • Wert liegt außerhalb des Zahlenbereichs (kleiner als $-1e^{12}$ oder größer als $1e^{12}$) • Wert mit ungültigem Zahlenformat
0400	Ungültiger Wert für den Parameter Output: <ul style="list-style-type: none"> • Wert liegt außerhalb des Zahlenbereichs (kleiner als $-1e^{12}$ oder größer als $1e^{12}$) • Wert mit ungültigem Zahlenformat
800	Abtastzeitfehler: Die Anweisung PID_3STEP wird in einem Programmzyklus-OB (wie OB 1) aufgerufen oder die Einstellungen für den Weckalarm-OB wurden geändert.
1000	Ungültiger Wert für den Parameter Setpoint: <ul style="list-style-type: none"> • Wert liegt außerhalb des Zahlenbereichs (kleiner als $-1e^{12}$ oder größer als $1e^{12}$) • Wert mit ungültigem Zahlenformat

8.4 PID-Regler konfigurieren

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

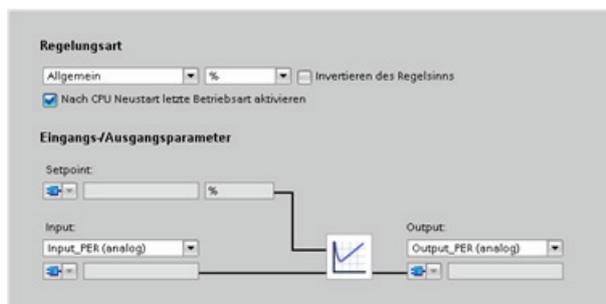


Bild 8-6 Konfigurationseditor für PID_Compact (Grundlegende Einstellungen)

Tabelle 8- 8 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.	

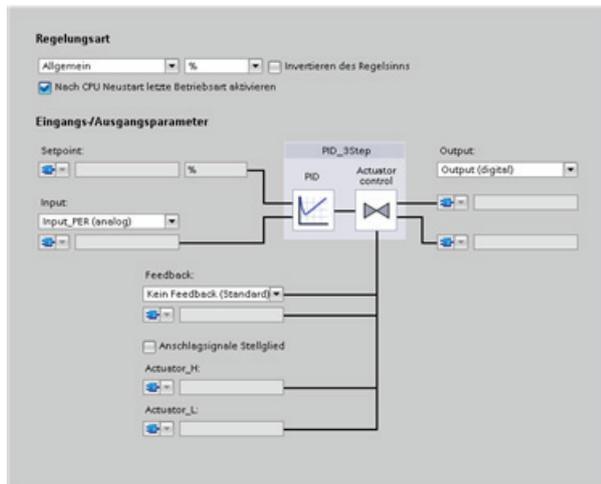


Bild 8-7 Konfigurationseditor für PID_3Step (Grundlegende Einstellungen)

Tabelle 8-9 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.
	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	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.	
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.)

Einstellungen		Beschreibung
	Kleinste AUS-Zeit	Legt die minimale Pausenzeit für das Ventil fest. (Ermitteln Sie diesen Wert auf dem Datenblatt oder anhand des Ventil-Typenschildes.)
	Fehlerverhalten	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"> • "Stopp oben" und "Stopp untere Grenze" definieren die maximale positive Position (vollständig geöffnet) und die maximale negative Position (vollständig geschlossen). "Stopp oben" muss größer als "Stopp untere Grenze" 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.

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

8.5 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- 10 Beispiel für einen Konfigurationsbildschirm (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) 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>
--	--

Webserver für einfachen Internetanschluss

Der Webserver bietet Webseitenzugriff auf Daten über Ihre CPU und auf Prozessdaten in der CPU. Ein Satz Standard-Webseiten ist in die Firmware der CPU integriert. Mit diesen Webseiten können Sie über den Webbrowser Ihres PCs auf die CPU zugreifen. Die Standard-Webseiten ermöglichen Ihnen die Durchführung 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 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.

Sie können jeden Webbrowser nutzen, der HTTP Version 1.1 unterstützt.

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.

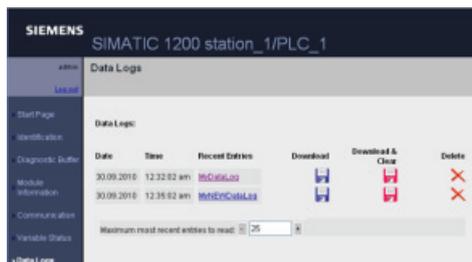
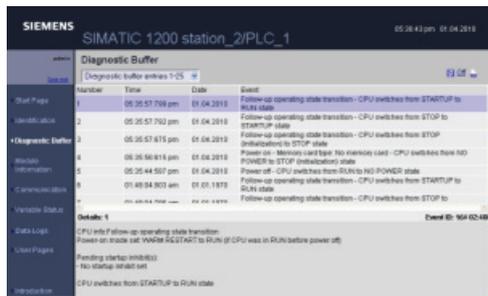


Die Startseite zeigt eine Darstellung der CPU an, mit der Sie verbunden sind, und führt allgemeine Informationen über die CPU auf.

Wenn Sie sich als Benutzer "admin" anmelden, können Sie den Betriebszustand der CPU (STOP bzw. RUN) wechseln und die LEDs blinken lassen.



Auf der Variablenzustandsseite können Sie alle E/A oder Speicherdaten in Ihrer CPU überwachen oder ändern. 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. 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 Datenprotokollseite können Sie eine angegebene Anzahl von Datenprotokolleinträgen anzeigen oder herunterladen. Der Webserver lädt die Datenprotokolle im amerikanischen CSV-Format (durch Komma getrennte Werte) auf Ihren PC. Beachten Sie für weitere Informationen den Abschnitt zu Datenprotokollen (Seite 116).

Andere Standard-Webseiten zeigen Informationen über die CPU an (z. B. Seriennummer, Version und Bestellnummer), über die Kommunikationsparameter (z. B. Netzwerkadressen, physikalische Eigenschaften der Kommunikationsschnittstellen und Kommunikationsstatistik) und über die Module im lokalen Baugruppenträger.

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

Die folgenden IT-Aspekte können sich auf Ihre Nutzung des Webserver auswirken:

- Üblicherweise müssen Sie die IP-Adresse der CPU eingeben, um auf die Standard-Webseiten bzw. die benutzerdefinierten Webseiten zuzugreifen. 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 JavaScripts und Cookies. Wenn JavaScripts oder Cookies in Ihrem Webbrowser deaktiviert sind, aktivieren Sie sie. Wenn Sie sie nicht aktivieren können, sind einige Funktionen eingeschränkt. Die Verwendung von JavaScripts 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 Webserverns über die `https://`-Adresse eine Sicherheitsabfrage angezeigt.

9.2.1 Einschränkungen bei deaktiviertem JavaScript

Bei deaktiviertem JavaScript sind einige Funktionen eingeschränkt

Die Standard-Webseiten sind über HTML, JavaScripts und Cookies implementiert. Sofern an Ihrem Standort die Verwendung von JavaScripts und Cookies nicht eingeschränkt ist, aktivieren Sie sie, damit die Webseiten einwandfrei funktionieren. Wenn Sie für Ihren Webbrowser keine JavaScripts aktivieren können, können die von JavaScripts gesteuerten Funktionen nicht ausgeführt werden.

Tabelle 9- 1 Bei deaktiviertem JavaScript betroffene Webseiten

Standard-Webseite	Auswirkungen
Allgemeines	<ul style="list-style-type: none"> • Die Daten auf den Seiten werden nicht dynamisch aktualisiert. Sie müssen die Seiten manuell über das Symbol zum Aktualisieren aktualisieren, um die aktuellen Daten anzuzeigen. • Sie können sich nicht als Benutzer "admin" anmelden.
Modulinformationen	<ul style="list-style-type: none"> • Sie können keine Daten filtern. • Sie können keine Felder sortieren.
Diagnose	<ul style="list-style-type: none"> • Einzelheiten zu Ereignissen anzeigen: Ohne JavaScript müssen Sie auf den Hyperlink des Ereignisfelds eines Diagnosepuffereintrags klicken, um die Ereignisdaten im unteren Bereich anzuzeigen. • Bereich der anzuzeigenden Diagnosepuffereinträge ändern: Ohne JavaScript können Sie über die Klappliste oben auf der Seite den Bereich der anzuzeigenden Diagnosepuffereinträge auswählen, doch Sie müssen dann auf den Link "Los" klicken, um die Diagnosepufferseite entsprechend zu aktualisieren.

Standard-Webseite	Auswirkungen
Variable	<ul style="list-style-type: none">• Nachdem Sie eine Variable eingegeben haben, müssen Sie den Fokus manuell in die Zeile "Neue 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 "Wert überwachen" klicken, um die Anzeige mit dem neuen Format zu aktualisieren.
Datenprotokolle	<ul style="list-style-type: none">• Sie können nicht auf einen Dateinamen unter "Letzte Einträge" klicken, um eine Protokolldatei zu öffnen oder zu speichern. Sie können für die gleiche Funktionalität jedoch das Symbol zum Herunterladen nutzen.• Die Datenprotokollseite wird nicht aktualisiert.• Die Schaltflächen "+" und "-" zum Inkrementieren und Dekrementieren der Anzahl der Einträge sind wirkungslos.• Beachten Sie, dass Sie die Datenprotokollseite verlassen und wieder aufrufen können, um die neuesten 25 Einträge anzuzeigen.

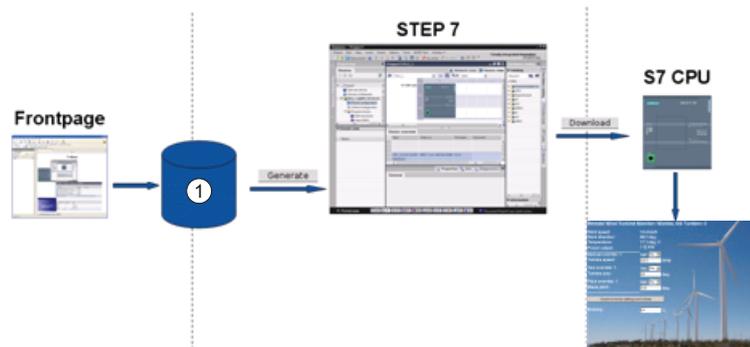
9.2.2 Eingeschränkte Funktionen, wenn keine Cookies zugelassen sind

Wenn Ihr Webbrowser keine Cookies zulässt, können Sie sich nicht als Benutzer "admin" anmelden.

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 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">
```

Und achten Sie 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.

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.

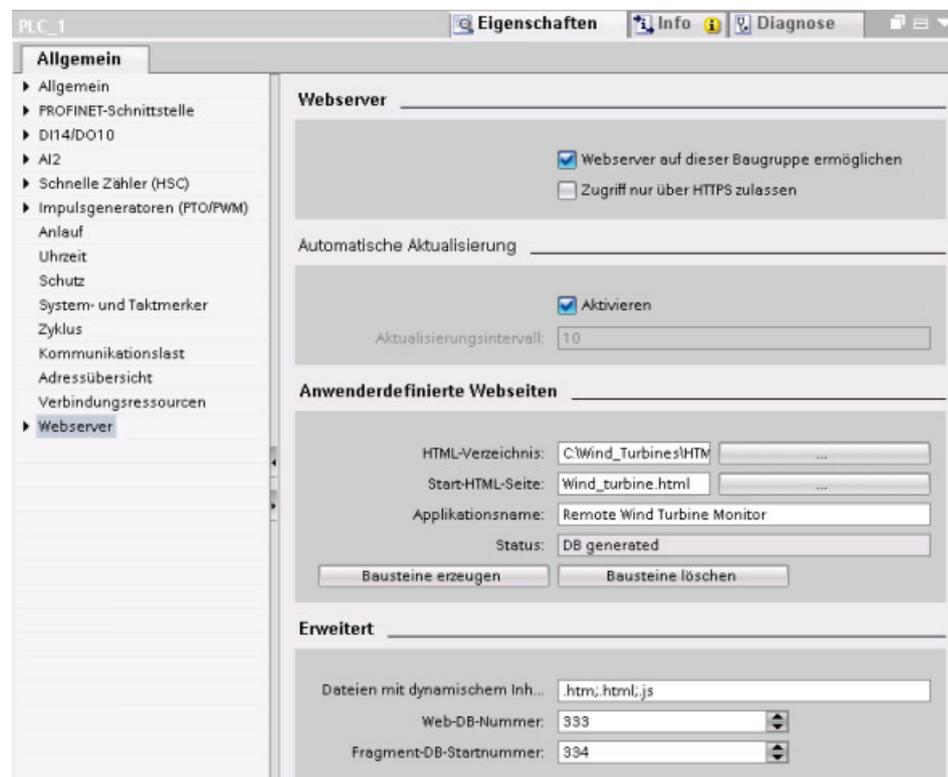
Mit den Online- und Diagnosefunktionen in STEP 7 können Sie den belegten und den freien Platz im Ladespeicher abfragen. Sie können zudem in den Eigenschaften der einzelnen Bausteine, die STEP 7 für Ihre benutzerdefinierten Webseiten generiert, den benötigten Platz im Ladespeicher prüfen.

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- 2 Anweisung WWW

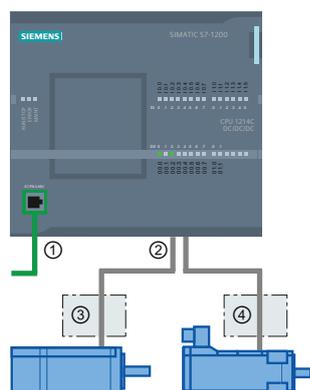
KOP/FUP	SCL	Beschreibung
	<pre>ret_val := #WWW(ctrl_db:=_uint_in_);</pre>	<p>Gibt den Steuer-DB für die benutzerdefinierten Webseiten an.</p> <p>Der Steuerdatenbaustein ist der Eingangsparameter für die WWW-Anweisung und gibt den Inhalt der Seiten wie in den Datenbausteinfragmenten dargestellt sowie 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.



- 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 2 Digitalausgängen kann als Impuls- und Richtungsausgang zur Steuerung eines Motors verwendet werden. Ein SB mit 4 Digitalausgängen kann für Impuls- und Richtungsausgänge zur Steuerung zweier Motoren verwendet werden. Integrierte Relaisausgänge können nicht als Impulsausgänge zur Motorsteuerung verwendet 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 grundlegende Bewegungssteuerungsanweisungen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0 bis A0.3, A4.0 bis A4.3) aus dem Speicher der Ausgänge entfernt und können in Ihrem Anwenderprogramm nicht für andere Zwecke 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.

Tabelle 10- 1 Maximale Anzahl von steuerbaren Antrieben

Typ der CPU		Kein SB installiert	Mit einem SB (2 x DC-Ausgänge)	Mit einem SB (4 x DC-Ausgänge)
CPU 1211C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2
CPU 1212C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2
CPU 1214C	DC/DC/DC	2	2	2
	AC/DC/RLS	0	1	2
	DC/DC/RLS	0	1	2

Tabelle 10- 2 Grenzfrequenzen von Impulsausgängen

Impulsausgang	Frequenz
Integriert	$2 \text{ Hz} \leq f \leq 100 \text{ kHz}$
Standard-SB	$2 \text{ Hz} \leq f \leq 20 \text{ kHz}$
Schnelle SBs (200 kHz)	MC-Anweisungen V2: $2 \text{ Hz} \leq f \leq 200 \text{ kHz}$ MC-Anweisungen V1: $2 \text{ Hz} \leq f \leq 100 \text{ kHz}$ ¹

¹ MC-Anweisungen V1 unterstützen eine maximale Frequenz von 100 kHz.

ACHTUNG

Die maximale Impulsfrequenz der Impulsgeneratoren beträgt 100 kHz für die digitalen Ausgänge der CPU, 20 kHz für die digitalen Ausgänge des Standard-Signalboards und 200 kHz für die digitalen Ausgänge der schnellen SBs (bzw. 100 kHz für MC-Anweisungen V1). Sie werden von STEP 7 jedoch **nicht** gewarnt, 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.

1. Konfigurieren eines Impulsgenerators: Wählen Sie die Eigenschaften "Impulsgeneratoren (PTO/PWM)" für eine CPU (in der Gerätekonfiguration) und aktivieren Sie einen Impulsgenerator. Bei jeder S7-1200 CPU stehen zwei Impulsgeneratoren zur Verfügung. In demselben Konfigurationsbereich wählen Sie unter "Impulsoptionen" für den Impulsgenerator die Verwendung "PTO" aus.
2. 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 konfigurierten PTO aus. Beachten Sie die zwei Ausgänge Q für Impuls und Richtung.
 - Konfigurieren Sie die restlichen Grund- und erweiterten Parameter.
3. 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.

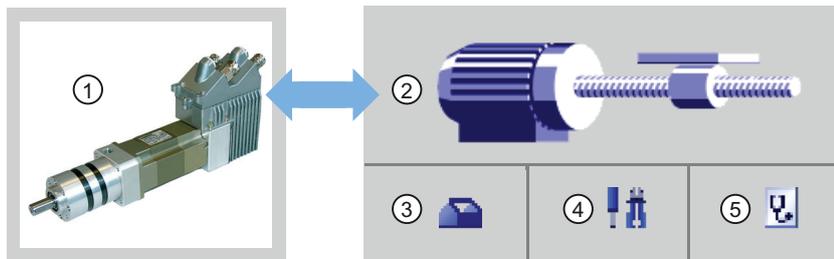
4. Fügen Sie die anderen Bewegungssteuerungsanweisungen ein, um die erforderliche Bewegung zu erzeugen.

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).

10.1 Konfiguration der Achse

STEP 7 stellt die Konfigurationswerkzeuge, Inbetriebnahmewerkzeuge und Diagnosewerkzeuge für das Technologieobjekt "Achse" zur Verfügung.



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

Hinweis

Die PTO benötigt die interne Funktionalität eines schnellen Zählers (HSC). Das bedeutet, dass der jeweilige schnelle Zähler 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 ID 1000) nicht überwachen, wenn Impulse auftreten.

Tabelle 10- 3 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.

ACHTUNG

Wenn Sie das Bemaßungssystem zu einem späteren Zeitpunkt ändern, werden die Werte möglicherweise nicht in allen Konfigurationsfenstern des Technologieobjekts korrekt umgewandelt. Prüfen Sie in diesem Fall die Konfiguration aller Achsenparameter.

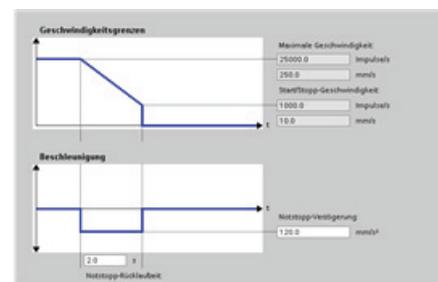
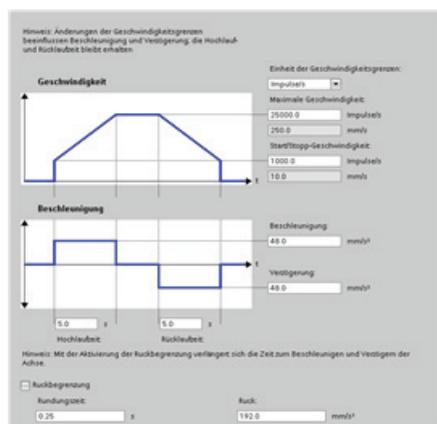
Sie müssen möglicherweise die Werte der Eingangsparameter der Bewegungssteuerungsanweisungen im Anwenderprogramm an die neue Einheit anpassen.



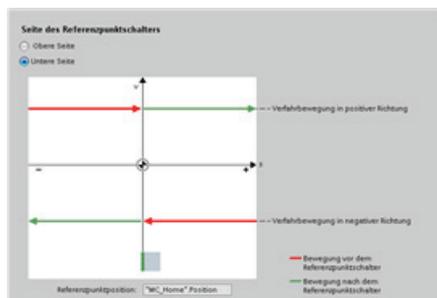
Konfigurieren Sie die Eigenschaften für die Antriebssignale, Antriebsmechanik und Positionsüberwachung (Hardware- und Software-Endschalter).

Deaktivieren Sie die Optionen für die Konfiguration eines Hardwaregrenzwerts oder eines Referenzpunkts nur dann, wenn der Eingang nicht mehr als Hardwaregrenzwert oder Referenzpunkt zugewiesen ist.

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.

10.2 Anweisung MC_Power

ACHTUNG

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- 4 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>

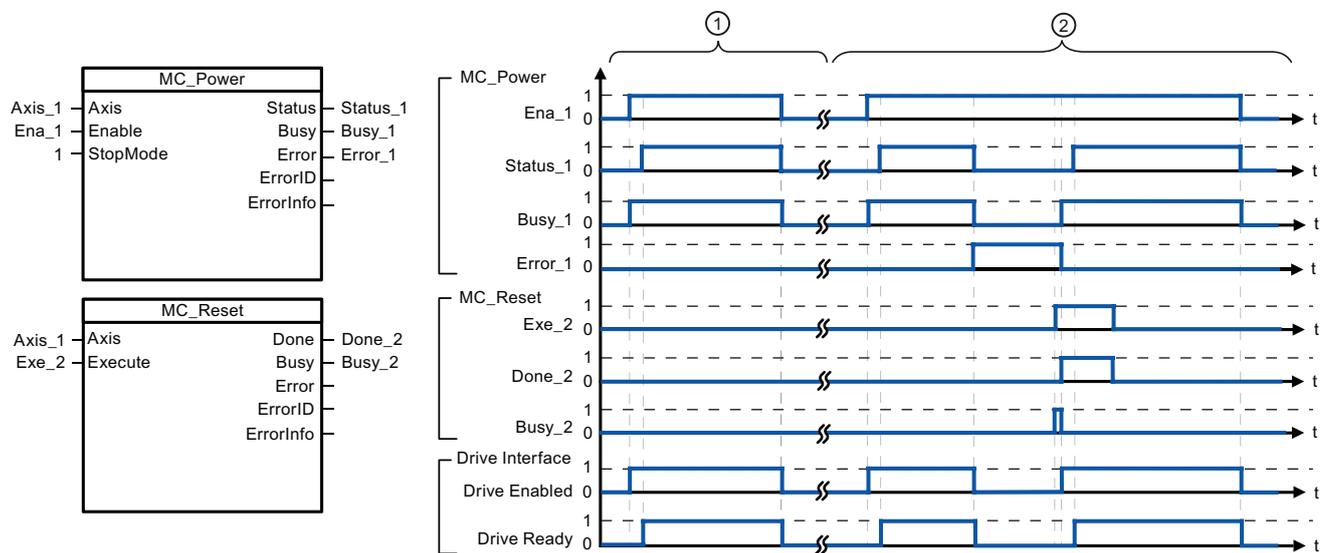
¹ STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.

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

Tabelle 10- 5 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"> • FALSCH (Standard): Alle aktiven Aufgaben werden entsprechend dem parametrisierten "StopMode" abgebrochen und die Achse wird gestoppt. • WAHR: Die Bewegungssteuerung versucht, die Achse zu aktivieren.
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.

Parameter und Datentyp		Datentyp	Beschreibung
Status	OUT	Bool	Status der Achsenfreigabe: <ul style="list-style-type: none"> FALSCH: Die Achse ist deaktiviert. <ul style="list-style-type: none"> Die Achse führt keine Bewegungssteuerungsaufgaben aus und nimmt keine neuen Aufgaben an (Ausnahme: MC_Reset). Die Achse ist nicht am Referenzpunkt. Beim Deaktivieren wechselt der Zustand erst dann nach FALSCH, wenn die Achse den Stillstand erreicht. WAHR: 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	Falsch: MC_Power ist nicht aktiv. WAHR: MC_Power ist aktiv.
Error	OUT	Bool	FALSCH: Kein Fehler WAHR: 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:

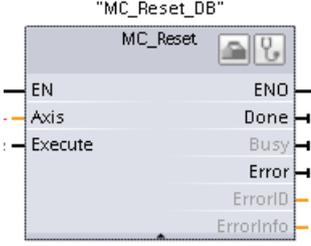
1. Prüfen Sie die oben angegebenen Voraussetzungen.
2. Initialisieren Sie den Eingangsparameter "StopMode" mit dem gewünschten Wert. Setzen Sie den Eingangsparameter "Enable" auf WAHR. Die Achse ist aktiviert. Der Ausgangsparameter "Status" und die Technologieobjektvariable <Achsenname>.StatusBits.Enable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

10.3 Anweisung MC_Reset

Tabelle 10- 6 Anweisung MC_Reset

KOP/FUP	SCL	Beschreibung
	<pre>"MC_Reset_DB" (Axis:=_multi_fb_in_, Execute:=_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 ungültigen Beschleunigungswert im Technologieobjekt "Achse" in einen gültigen Wert ändern).</p>

- STEP 7 erstellt den DB automatisch, wenn Sie die Anweisung einfügen.
- 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- 7 Parameter der Anweisung MC_Reset

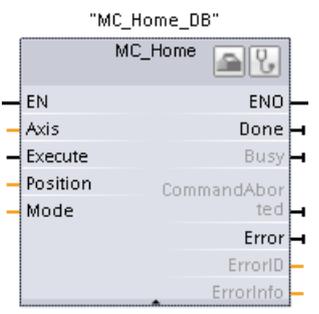
Parameter und Datentyp	Datentyp	Beschreibung
Axis	IN	Technologieobjekt "Achse"
Execute	IN	Starten der Aufgabe bei einer positiven Flanke
Done	OUT	WAHR = Fehler wurde quittiert.
Busy	OUT	WAHR = Die Aufgabe wird ausgeführt.
Error	OUT	WAHR = Während der Ausführung der Aufgabe trat ein Fehler auf. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt.
ErrorID	OUTP	Fehler-ID für Parameter "Error"
ErrorInfo	OUT	Fehlerinfo-ID für Parameter ErrorID

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

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

10.4 Anweisung MC_Home

Tabelle 10- 8 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- 9 Parameter für die MC_Home-Anweisung

Parameter und Datentyp	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: $-1,0e^{12} \leq \text{Position} \leq 1,0e^{12}$

Parameter und Datentyp		Datentyp	Beschreibung
Mode	IN	Int	<p>Art der Referenzpunktfahrt</p> <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- 10 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

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 Positionseingangs der Anweisung für die Referenzpunktfahrt zum Hardware-Referenzpunkt addiert.

Bei der aktiven Referenzpunktfahrt führt die Anweisung MC_Home die erforderliche Referenzpunktfahrtbewegung 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 Referenzpunktschalter kann im Arbeitsbereich der Achse oder außerhalb des normalen Arbeitsbereichs der Achse, jedoch innerhalb des Bewegungsbereichs platziert werden.

10.5 Anweisung MC_Halt

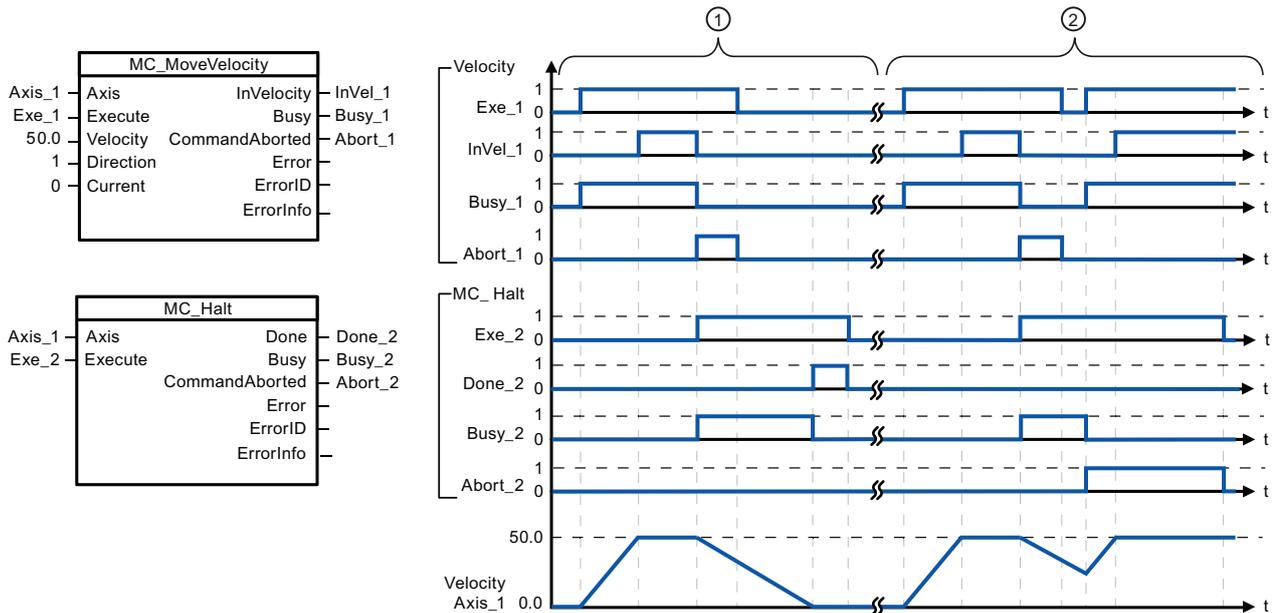
Tabelle 10- 11 Anweisung MC_Halt

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 jegliche Bewegung 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- 12 Parameter für die MC_Halt-Anweisung

Parameter und Datentyp	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:

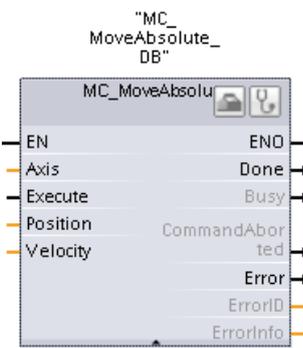
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

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

10.6 Anweisung MC_MoveAbsolute

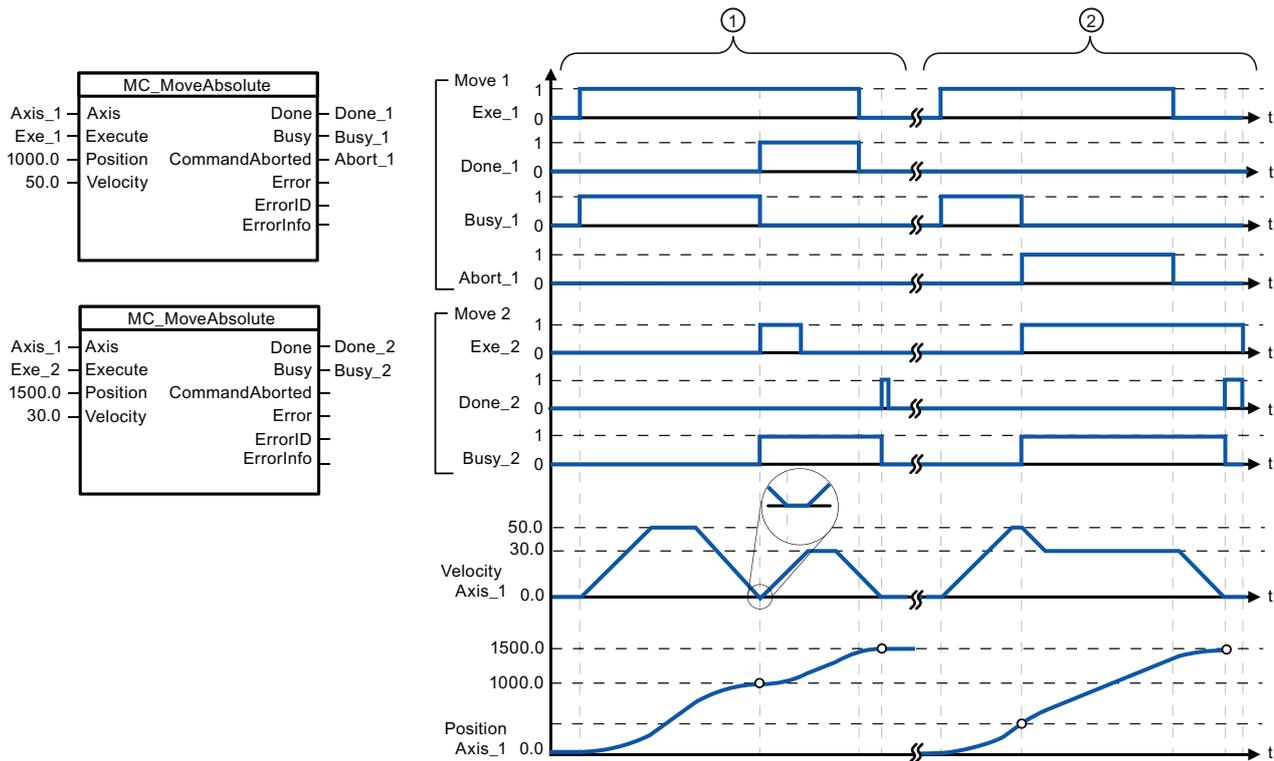
Tabelle 10- 13 Anweisung MC_MoveAbsolute

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>

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

Tabelle 10- 14 Parameter für die MC_MoveAbsolute-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)
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:

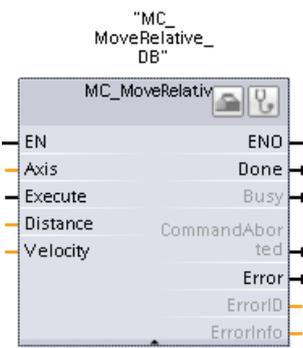
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

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

10.7 Anweisung MC_MoveRelative

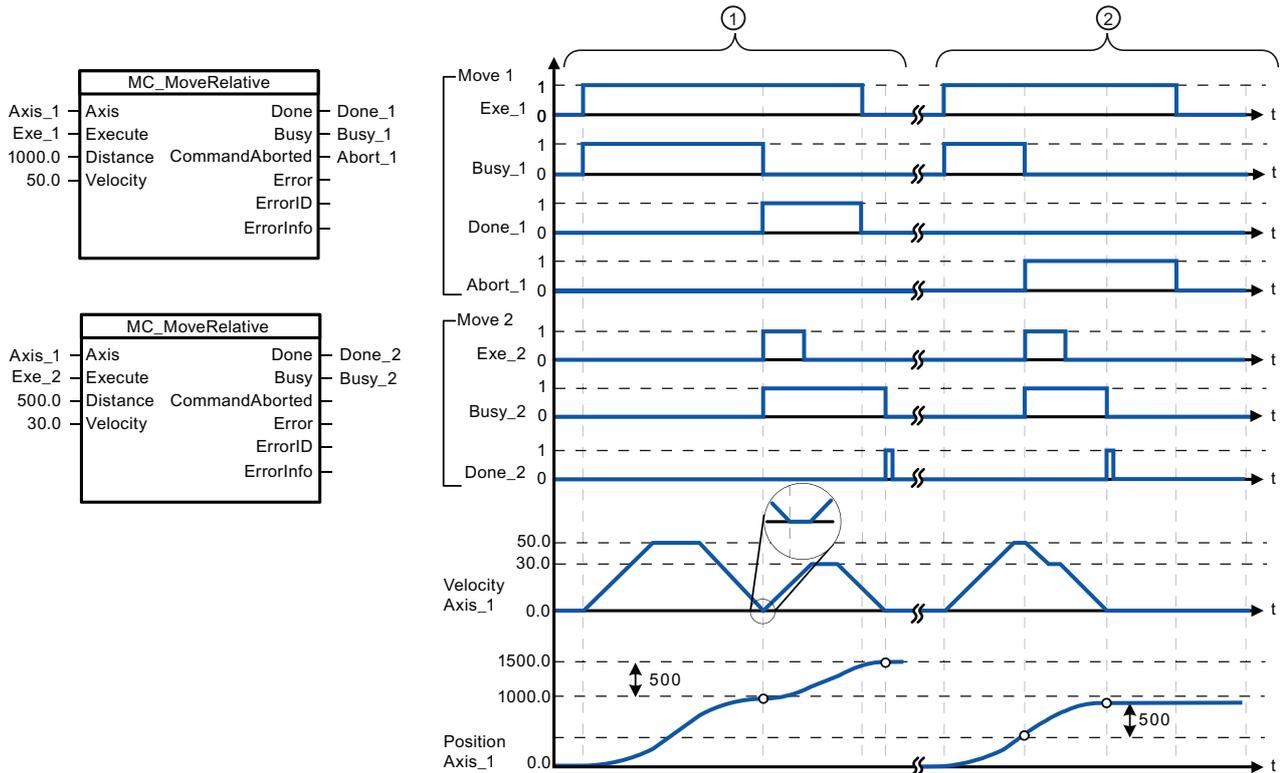
Tabelle 10- 15 Anweisung MC_MoveRelative

KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveRelative_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Distance:=_real_in_, Velocity:=_real_in_, Done=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Mit der Anweisung MC_MoveRelative starten Sie eine Positionierbewegung relativ zur Startposition.</p> <p>Um die Anweisung MC_MoveRelative zu verwenden, muss die Achse zunächst freigegeben werden.</p>

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

Tabelle 10- 16 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:

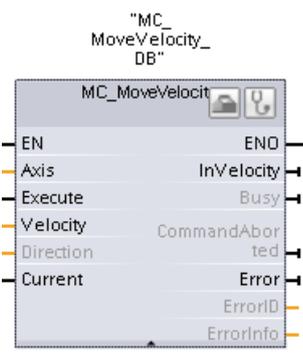
- MC_Home Mode = 3
- MC_Halt
- MC_MoveAbsolute
- MC_MoveRelative
- MC_MoveVelocity
- MC_MoveJog

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

10.8 Anweisung MC_MoveVelocity

Tabelle 10- 17 Anweisung MC_MoveVelocity

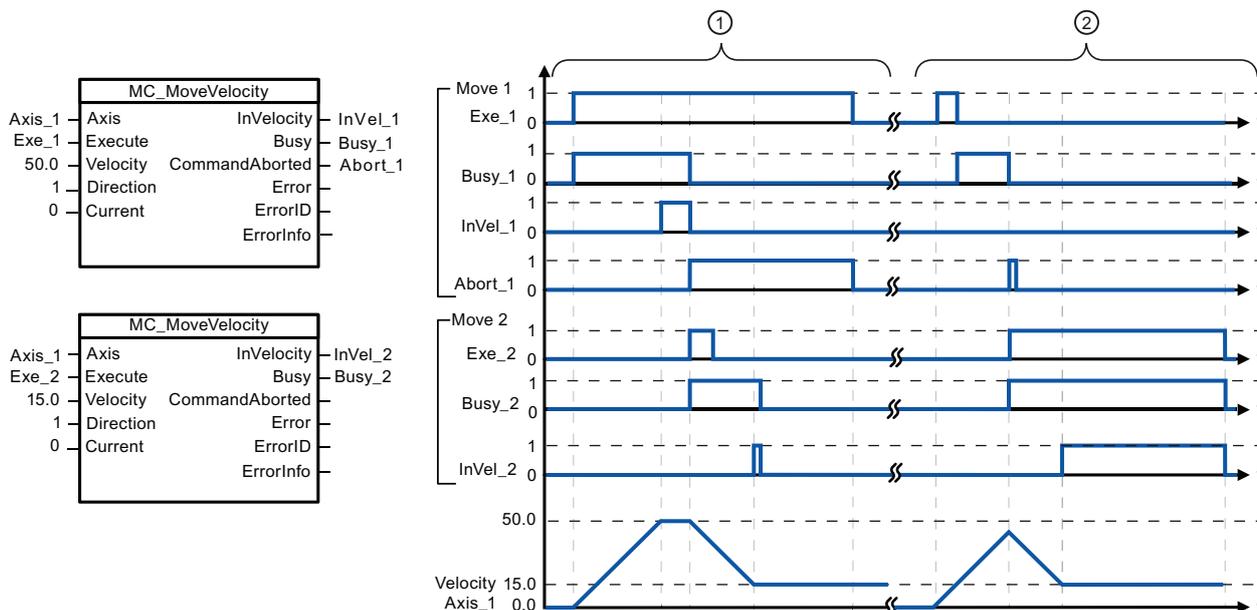
KOP/FUP	SCL	Beschreibung
	<pre>"MC_MoveVelocity_DB" (Axis:=_multi_fb_in_, Execute:=_bool_in_, Velocity:=_real_in_, Direction:=_int_in_, Current:=_bool_in_, InVelocity=>_bool_out_, Busy=>_bool_out_, CommandAborted=>_bool_out_, Error=>_bool_out_, ErrorID=>_word_out_, ErrorInfo=>_word_out_);</pre>	<p>Mit 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- 18 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 wieder aufnimmt, 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)

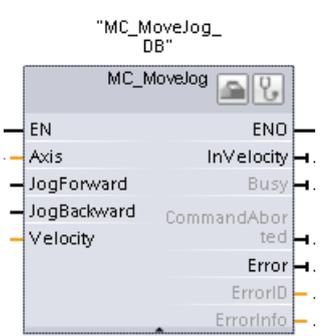
Eine Aufgabe MC_MoveVelocity mit "Velocity" = 0,0 (wie eine Aufgabe MC_Halt) bricht aktive Bewegungssteuerungsaufgaben ab und stoppt die Achse mit der konfigurierten Verzögerung. Wenn die Achse zum Stillstand kommt, gibt der Ausgangsparameter "InVelocity" mindestens einen Programmzyklus lang WAHR an.

"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.

10.9 Anweisung MC_MoveJog

Tabelle 10- 19 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>

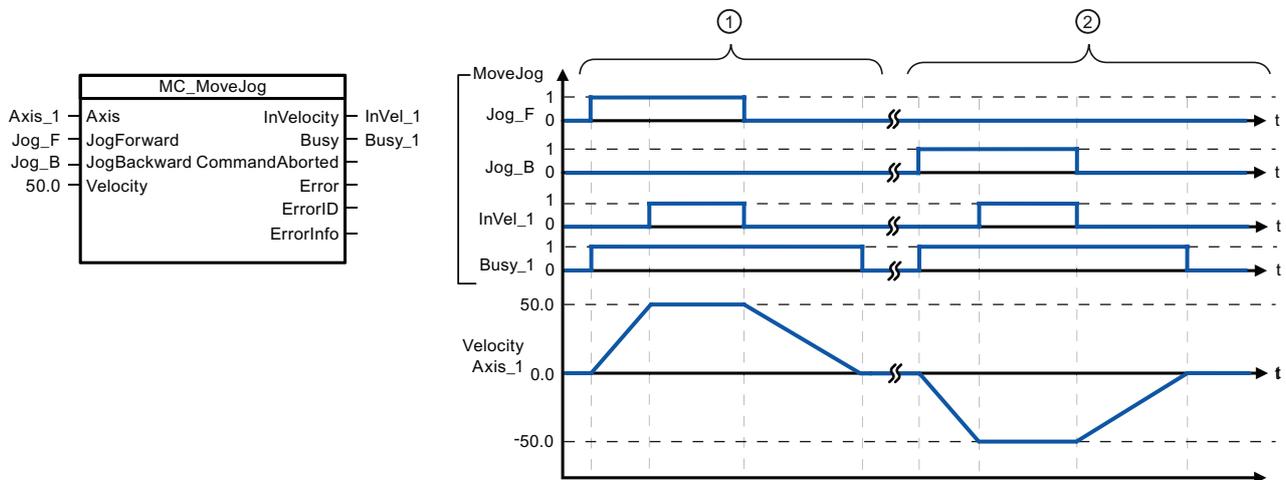
- STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- Im SCL-Beispiel ist "MC_MoveJog_DB" der Name des Instanz-DBs.

Tabelle 10- 20 Parameter für die MC_MoveJog-Anweisung

Parameter und 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)

Parameter und Datentyp	Datentyp	Beschreibung
JogBackward ¹	IN	Bool
Velocity	IN	Real
InVelocity	OUT	Bool
Busy	OUT	Bool
CommandAborted	OUT	Bool
Error	OUT	Bool
ErrorID	OUT	Word
ErrorInfo	OUT	Word

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

10.10 Anweisung MC_CommandTable

Tabelle 10- 21 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>

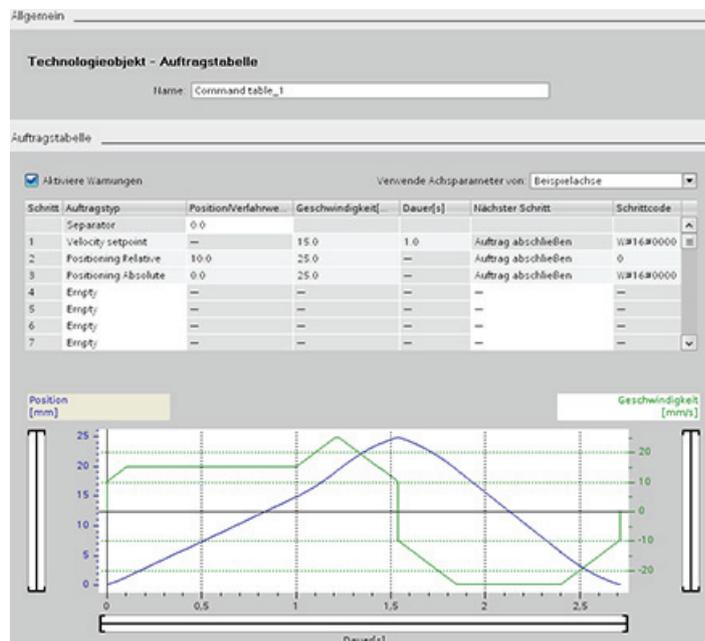
- 1 STEP 7 erstellt automatisch den DB, wenn Sie die Anweisung einfügen.
- 2 Im SCL-Beispiel ist "MC_CommandTable_DB" der Name des Instanz-DBs.

Tabelle 10- 22 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	Booll	FALSCH	Auftrag bei steigender Flanke starten
StartIndex	IN	Int	1	Verarbeitung der Befehlstabelle mit diesem Schritt starten Grenzwerte: $1 \leq \text{StartIndex} \leq \text{EndIndex}$

Parameter und Datentyp		Datentyp	Anfangswert	Beschreibung
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	Fehlerkennung
ErrorInfo	OUT	Word	16#0000	Fehlerinformation
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- 23 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 um die angegebene Lage und Geschwindigkeit.
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 fungiert als Bereichsgrenze für die grafische Anzeige des Kurvendiagramms.

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

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"

10.11 MC_ChangeDynamic

Tabelle 10- 24 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 für die Beschleunigungsänderung • Wert für die Verzögerungsänderung • Wert für die Not-Aus-Verzögerungsänderung • Ruckbegrenzung aktivieren/deaktivieren • Wert für die Ruckbegrenzungsänderung

- 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- 25 Parameter für die MC_ChangeDynamic-Anweisung

Parameter und Datentyp	Datentyp	Datentyp	Beschreibung
Axis	IN	TO_Axis_1	Technologieobjekt "Achse"
Execute	IN	Bool	Ausführung bei steigender Flanke starten. Anfangswert: FALSCH
ChangeRampUp	IN	Bool	WAHR = Hochlaufzeit in den Wert ändern, der vom Parameter RampUp Time angegeben wird. Anfangswert: FALSCH
RampUpTime	IN	Real	Zeit (in Sekunden) ohne Ruckbegrenzung, während der vom Stillstand zur konfigurierten Maximalgeschwindigkeit beschleunigt werden soll. Anfangswert: 5.00 Der Status der betroffenen Variablen wird im Wert von <Achsenname> angezeigt. Config.DynamicDefaults.Acceleration.
ChangeRampDown	IN	Bool	WAHR Ändern Sie die Freigabezeit entsprechend dem Eingangsparameter RampDownTime. Anfangswert: FALSCH
RampDownTime	IN	Real	Zeit (in Sekunden) ohne Ruckbegrenzung, während der von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand verzögert werden soll. Anfangswert: 5.00 Der Status der betroffenen Variablen wird im Wert von <Achsenname> angezeigt. Config.DynamicDefaults.Deceleration .
ChangeEmergency	IN	Bool	WAHR Ändern Sie die Not-Aus-Zeit entsprechend dem Eingangsparameter EmergencyRampTime. Anfangswert: FALSCH

Parameter und Datentyp		Datentyp	Beschreibung
EmergencyRampTime	IN	Real	Zeit (in Sekunden) ohne Ruckbegrenzung, während der die Verzögerung des Not-Aus-Betriebs von der konfigurierten Maximalgeschwindigkeit bis zum Stillstand kommen soll. Anfangswert: 2.00 Der Status der betroffenen Variablen wird gespeichert in: <Achsenname>. Config.DynamicDefaults.EmergencyDeceleration
ChangeJerkEnable	IN	Bool	WAHR = Ändern Sie die Ruckbegrenzung entsprechend dem Eingangsparameter JerkEnable: Anfangswert: FALSCH
JerkEnable	IN	Bool	WAHR = Aktivieren Sie die Ruckbegrenzung. Anfangswert: FALSCH Der Status der betroffenen Variablen wird gespeichert in: <Achsenname>. Config.DynamicDefaults.JerkActive.
ChangeRoundingOff	IN	Bool	WAHR Ändern Sie die Rampe entsprechend dem Eingangsparameter RoundingOffTime: Anfangswert: FALSCH
RoundingOffTime	IN	Real	Rampe (in Sekunden), die auf die Beschleunigung und Verzögerung der Achse angewendet wird. Anfangswert: 0.20 Der Status der betroffenen Variablen wird gespeichert in: <Achsenname>. Config.DynamicDefaults.Jerk .
Done	OUT	Bool	WAHR = Die geänderten Werte werden in den Technologie-DB geschrieben. Anfangswert: FALSCH
Error	OUT	Bool	WAHR = Ein Fehler ist während der Bearbeitung aufgetreten. Die Fehlerursache ist in den Parametern ErrorID und ErrorInfo hinterlegt. Anfangswert: FALSCH
ErrorID	OUT	Word	Fehlerkennung. Anfangswert: 16#0000
ErrorInfo	IN	Word	Fehlerinformationen. Anfangswert: 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

Übersteuerungsantwort

Die Aufgabe MC_ChangeDynamic kann von jeder anderen Bewegungssteuerungsaufgabe gestoppt werden.

Eine neue Aufgabe MC_ChangeDynamic bricht alle aktuellen 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.

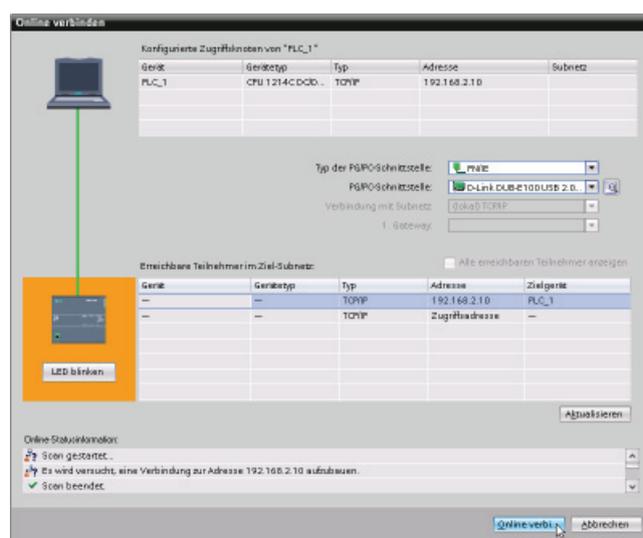
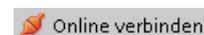
Einfaches Arbeiten mit den Online-Tools

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 230)
- Anzeigen und Einstellen von Datum und Uhrzeit der CPU (Seite 241)
- Anzeigen der Modulinformationen
- Vergleichen und Synchronisieren (Seite 239) von Offline- und Online-Programmbausteinen
- Laden von Programmbausteinen in die und aus der CPU
- Anzeigen von Diagnose und Diagnosepuffer (Seite 240)
- Mit einer Beobachtungstabelle (Seite 232) das Anwenderprogramm durch Beobachten und Steuern von Werten testen
- Mit einer Forcetabelle Werte in der CPU forcen (Seite 233)

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":

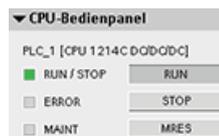


Wenn Sie zum ersten Mal mit dieser CPU online gehen, müssen Sie den Typ der PG/PC-Schnittstelle und die spezifische PG/PC-Schnittstelle im Dialog "Online verbinden" auswählen, bevor Sie eine Online-Verbindung zu einer CPU auf dieser Schnittstelle herstellen.

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.

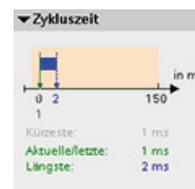
11.2 Interaktion mit der Online-CPU

Im Portal "Online & Diagnose" steht Ihnen ein Bedienpanel zur Verfügung, in dem Sie den Betriebszustand der Online-CPU ändern können. 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 Urlöschen des Speichers.

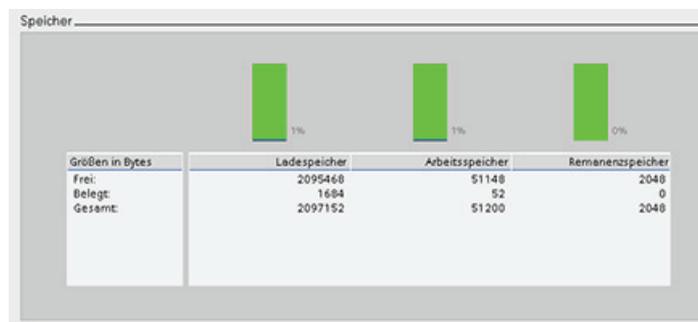


Der aktuelle Betriebszustand der CPU wird durch die Farbe der RUN/STOP-Anzeige angegeben: Gelb steht für den Betriebszustand STOP, Grün für RUN.

Um das Bedienpanel verwenden zu können, müssen Sie mit der CPU online verbunden sein. 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.



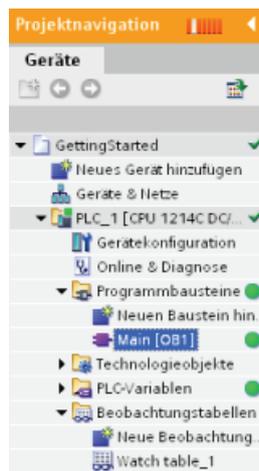
Sie können die Zykluszeit einer Online-CPU überwachen.



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 tabellen zeigen die Variablen. Beobachtungstabellen können auch die Variablen anzeigen, ebenso wie direkte Adressen.

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	"On"	%E0.0	Bool		
2	"Off"	%E0.1	Bool		
3	"Run"	%A0.0	Bool		



Um die Ausführung des Anwenderprogramms zu beobachten und die Werte der Variablen anzuzeigen, klicken Sie in der Funktionsleiste auf die Schaltfläche "Alle beobachten".

	Name	Adresse	Anzeigeformat	Beobachtungswert	Steuerwert
1	"On"	%E0.0	Bool	<input type="checkbox"/> FALSE	
2	"Off"	%E0.1	Bool	<input type="checkbox"/> FALSE	
3	"Run"	%A0.0	Bool	<input type="checkbox"/> FALSE	

Im Feld "Beobachtungswert" wird für jede Variable der Wert angezeigt.

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 Signalfuss 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.

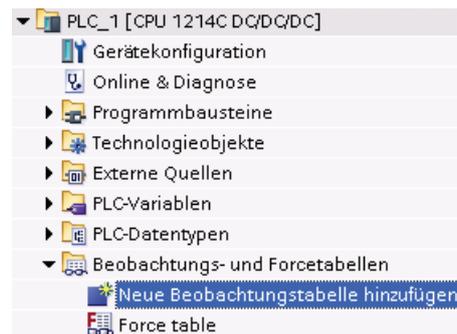
	Name	Adresse	Anzeigeformat	Beobachtungswert	Beobachten mit Trigger	Steuern mit Trigger	Steuwert
1	"Start"	%I0.0	Bool		Permanent	Permanent	<input type="checkbox"/>
2	"Stop"	%I0.1	Bool		Permanent	Permanent	<input type="checkbox"/>
3	"Running"	%M0.0	Bool		Permanent	Permanent	<input type="checkbox"/>

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	Forcewart
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"/>

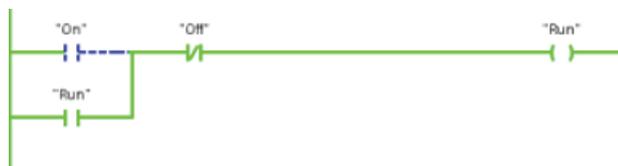
Geben Sie den Wert für den zu forcenden Eingang oder Ausgang in die Zelle "Forcewart" ein. Sie können dann das Kontrollkästchen in der Spalte "Forcen" aktivieren, um das Forcen des Eingangs oder Ausganges zu aktivieren.



Klicken Sie auf die Schaltfläche "Forcen starten oder ersetzen", um die Werte der Variablen in der Forcetabelle zu forcen. Klicken Sie auf die Schaltfläche "Forcen beenden", um die Werte der Variablen zurückzusetzen.

In der Forcetabelle können Sie den Status des geforcten Werts eines Eingangs beobachten. Den geforcten Wert eines Ausganges können Sie jedoch nicht beobachten.

Sie können den Zustand der geforcten Werte auch im Programmiereditor anzeigen.



ACHTUNG

Wenn ein Eingang oder Ausgang in einer Forcetabelle geforct wird, werden die Forceaktionen Teil der Projektconfiguration. 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 Beobachtungstabelle die physische Adresse der 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 geforcten Wert erscheint am physischen Ausgang und wird im Prozess verwendet.

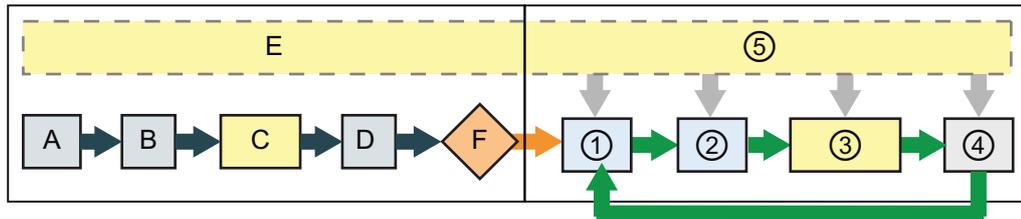
Wenn ein Eingang oder Ausgang in der Forcetabelle geforct wird, werden die Forceaktionen Teil des Anwenderprogramms. Auch wenn Sie die Programmiersoftware schließen, bleiben die geforcten Werte im ausgeführten CPU-Programm solange aktiv, bis Sie wieder in der Programmiersoftware online gehen und die 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 geforcten 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 Beobachtungstabelle geändert werden.



Anlauf

- A Das Löschen des Speicherbereichs E wird von der Forcetablelle nicht beeinflusst.
- B Die Initialisierung der Ausgangswerte wird von der Forcetablelle nicht beeinflusst.
- C Während der Ausführung der Anlauf-OBs schaltet die CPU den Forcetablellewert 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 Forcetablellewert bei der Aktualisierung der Ausgänge auf.
- ② Beim Lesen der physischen Eingänge wendet die CPU die Forcetablellewerte an, kurz bevor die Eingänge in den Speicherbereich E kopiert werden.
- ③ Während der Ausführung der Anwenderprogramms (Programmzyklus-OBs) schaltet die CPU den Forcetablellewert 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 Forcetablellefunktion 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

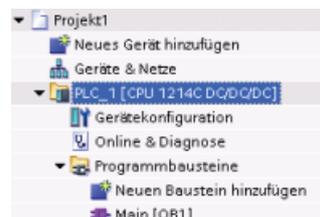
Die Werte, die in der Spalte "Beobachtungswert" angezeigt werden, sind immer aus der CPU kopiert. STEP 7 prüft nicht, ob alle Werte aus demselben Zyklus der CPU stammen.

11.8 Elemente des Projekts kopieren

Sie können die Programmbausteine einer Online-CPU oder einer an Ihr Programmiergerät angeschlossenen Memory Card auch kopieren.

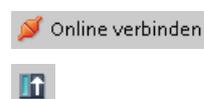
Bereiten Sie das Offline-Projekt für die kopierten Programmbausteine vor:

1. Fügen Sie eine CPU hinzu, die der Online-CPU entspricht.
2. Erweitern Sie den CPU-Knoten, so dass der Ordner "Programmbausteine" angezeigt wird.



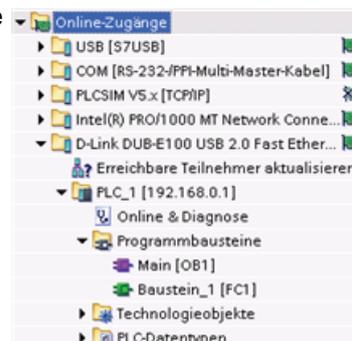
Um die Programmbausteine aus der Online-CPU in das Offline-Projekt zu laden, gehen Sie folgendermaßen vor:

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.

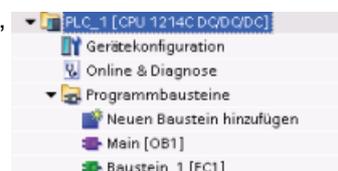


Als Alternative zur vorherigen Vorgehensweise gehen Sie folgendermaßen vor:

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.

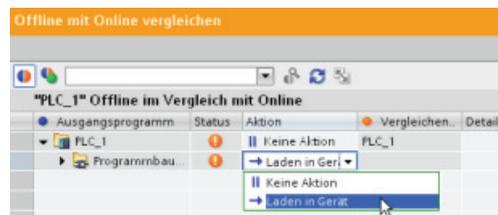
11.9 Vergleichen von Offline- und Online-CPU's

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.



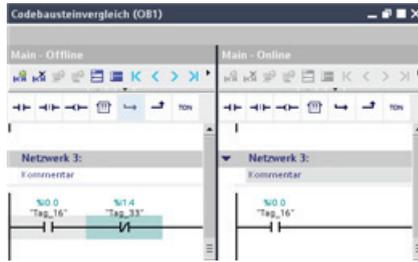
Wählen Sie die CPU in Ihrem Projekt aus.

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.



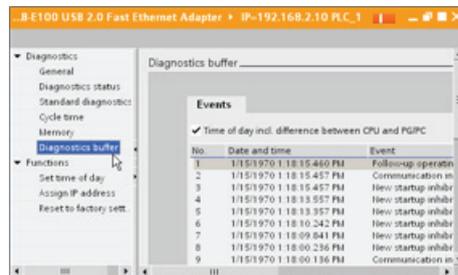
Klicken Sie in der Spalte "Vergleichen mit" mit der rechten Maustaste auf ein Objekt und wählen Sie "Detaillierten Vergleich starten", um die Codebausteine nebeneinander anzuzeigen.

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.

Weitere Informationen finden Sie im Abschnitt zur IP-Adresse.



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".

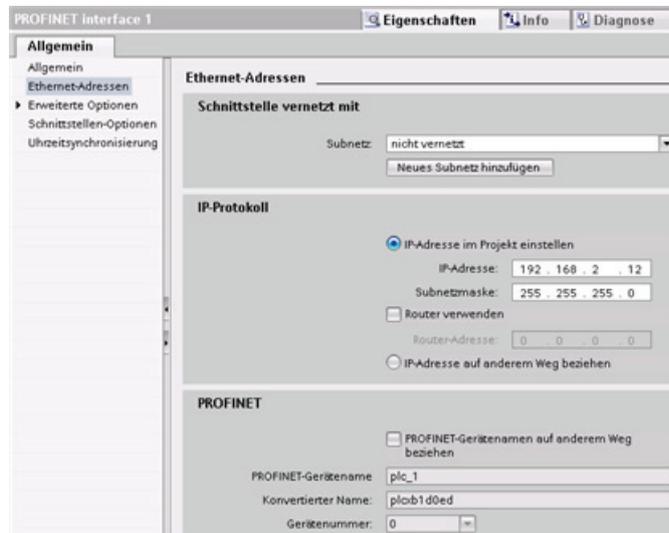
Ergebnis

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.)

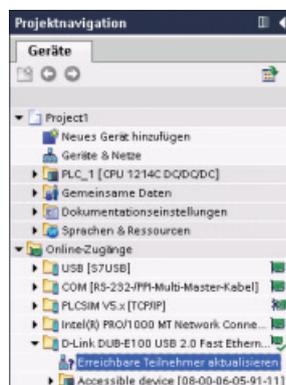
11.13 Laden einer permanenten IP-Adresse in eine Online-CPU

Um eine IP-Adresse zuzuweisen, gehen Sie folgendermaßen vor:



- Richten Sie die IP-Adresse für die CPU (Seite 84) 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 mit der Gerätekonfiguration zusammen geladen wird, geht beim Ausschalten des Zielsystems nicht verloren.

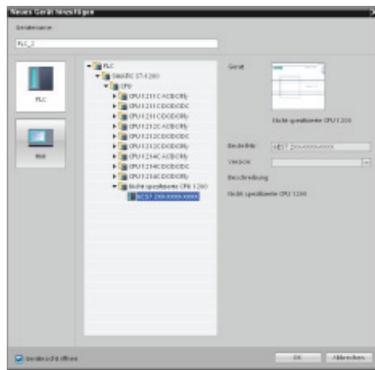
Nachdem Sie die Gerätekonfiguration einschließlich der IP-Adresse geladen haben, erscheint die IP-Adresse unter dem Ordner "Online-Zugänge".

11.14 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.

11.14 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.

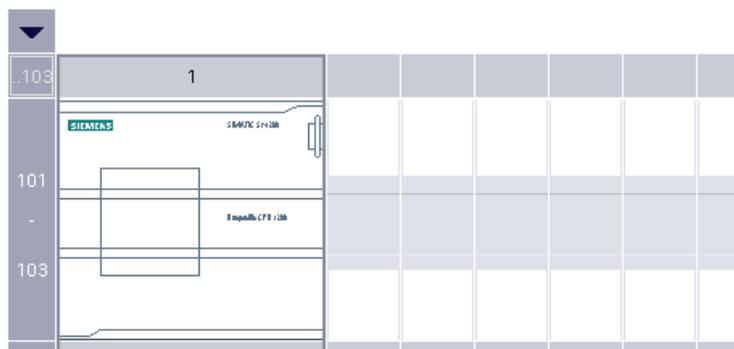


Fügen Sie in der Gerätekonfiguration (Projektansicht oder Portalansicht) ein neues Gerät hinzu, wählen Sie jedoch anstelle einer bestimmten CPU die "nicht spezifizierte CPU". STEP 7 legt eine nicht spezifizierte CPU an.



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.

11.15 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

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 Variablentabellen erstellen, überschreiben und löschen
- Datenbausteine (DBs) erstellen und löschen; DB-Strukturänderungen können jedoch nicht überschrieben werden. Ausgangswerte des DBs können überschrieben werden. Im Betriebszustand RUN können Sie keinen Webserver-DB (Steuerung oder Fragment) laden.
- Organisationsbausteine (OBs) überschreiben; es können jedoch keine OBs erstellt oder gelöscht werden.

Maximal zehn Bausteine können gleichzeitig im Betriebszustand RUN geladen werden. Werden mehr als zehn Bausteine geladen, muss die CPU in den Betriebszustand STOP versetzt werden.

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

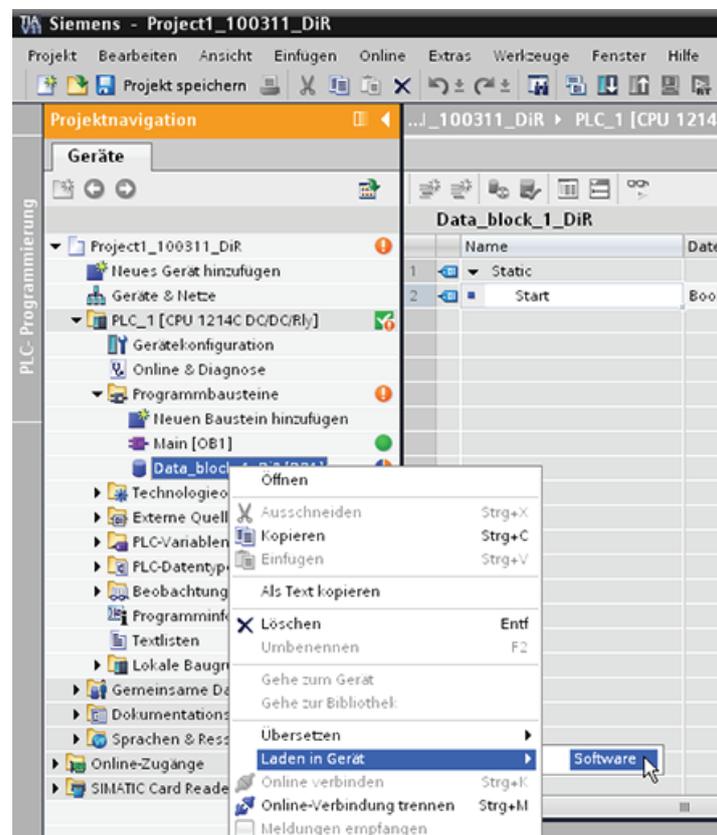
Befindet sich die CPU im Betriebszustand RUN und wurden Programmänderungen vorgenommen, versucht das TIA-Portal zunächst stets, die Änderungen im Betriebszustand RUN zu laden. Möchten Sie dies verhindern, müssen Sie die CPU in STOP versetzen.

Werden die vorgenommenen Änderungen nicht von der Funktion "Laden im Betriebszustand RUN" unterstützt, werden Sie vom TIA-Portal aufgefordert, die CPU in den Betriebszustand STOP zu versetzen.

11.15.1 Ändern des Programms im Betriebszustand RUN

Wenn Sie Ihr Programm im Betriebszustand RUN ändern möchten, müssen Sie zunächst sicherstellen, dass die CPU das "Laden im Betriebszustand RUN" unterstützt und sich außerdem in RUN befindet:

1. Um Ihr Programm im Betriebszustand RUN zu laden, gehen Sie auf eine der folgenden Arten vor:
 - Wählen Sie im Menü "Online" den Befehl "Laden in Gerät".
 - Wählen Sie in der Symbolleiste 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".



2. Wird das Programm erfolgreich übersetzt, lädt STEP 7 das Programm in die CPU.
3. STEP 7 fordert Sie auf, das Programm zu laden oder den Vorgang abubrechen.
4. Wenn Sie auf "Laden" klicken, lädt STEP 7 das Programm in die CPU.

Technische Daten

A.1 Allgemeine technische Daten

Erfüllte Normen

Das Automatisierungssystem S7-1200 erfüllt die folgenden Normen und Prüfvorschriften. Die Prüfkriterien für S7-1200 beruhen auf diesen Normen und Prüfvorschriften.

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)

ACHTUNG
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 T4A Ta = 40 °C

Class I, Zone 2, IIC, Temperature Class T4 Ta = 40 °C

Canadian Class I, Zone 2 Installation nach CEC 18-150

Hinweis

Das Signalmodul SM 1223 DI 8 x 120/230 VAC, DO 8 x Relais (6ES7 223-1QH30-0XB0) ist für die Verwendung in Umgebungen der Class 1, Division 2, Gas Group A, B, C, D, Temperature Class T4 Ta = 40 °C zugelassen.

ATEX-Zulassung



EN 60079-0:2006: Explosionsfähige Atmosphäre - Allgemeine Anforderungen

EN 60079-15:2005: Elektrische Betriebsmittel für gasexplosionsgefährdete Bereiche; Schutzart 'n'

II 3 G Ex nA II T4

Darüber hinaus müssen die folgenden Bedingungen für den sicheren Einsatz der S7-1200 erfüllt werden:

- Die Module in einem geeigneten Gehäuse mit einer Schutzklasse von mindestens IP54 nach EN 60529 einbauen und die Umgebungsbedingungen für den Betrieb der Geräte berücksichtigen.
- Werden bei Nennbedingungen Temperaturen von 70 °C am Kabeleintritt bzw. 80 °C am Abzweigpunkt der Leitungen überschritten, so muss der zulässige Temperaturbereich des ausgewählten Kabels für die tatsächlich gemessenen Temperaturen geeignet sein.
- Es sind Vorkehrungen zu treffen, um zu verhindern, dass die Nennspannung durch kurzzeitige Störungen um mehr als 40 % überschritten wird.

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)

Hinweis

Das CM 1242-5 (PROFIBUS-Slavemodul), das CM 1243-5 (PROFIBUS-Mastermodul) und der CP 1242-7 (GPRS-Modul) haben keine Zulassung für das Seewesen.

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, 1kV Gegentakt DC-Systeme - 2 kV Gleichtakt, 1kV Gegentakt Für DC-Systeme (E/A-Signale, DC-Stromversorgungen) ist ein externer Schutz erforderlich.
EN 61000-4-6 Leitungsgeführte Störungen	150 kHz bis 80 MHz, 10 V effektiv, 80% AM bei 1 kHz
EN 61000-4-11 Spannungseinbrüche	AC-Systeme 0% für 1 Zyklus, 40% für 12 Zyklen und 70% für 30 Zyklen bei 60 Hz

Tabelle A- 3 Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4

Elektromagnetische Verträglichkeit - Leitungsgeführte und abgestrahlte Störaussendungen nach EN 61000-6-4		
Leitungsgeführte Störaussendungen EN 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

Umgebungsbedingungen - Transport und Lagerung	
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 (Luftzufuhr 25 mm unterhalb des Geräts)	0° C bis 55° C horizontale Montage 0° C bis 45° C vertikale Montage 95% Luftfeuchtigkeit, nicht kondensierend
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

Tabelle A- 6 Hochspannungsisolationsprüfung

Hochspannungsisolationsprüfung	
Stromkreis mit 24V/5VNennspannung	520 V DC (Typprüfung der optischen Potentialtrennungsgrenzen)
115/230VStromkreis an Erde	1.500 V AC Routineprüfung/1950 V DC Typprüfung
115/230VStromkreis an 115/230VStromkreis	1.500 V AC Routineprüfung/1950 V DC Typprüfung
115/230VStromkreis an 24/5VStromkreis	1.500 V AC Routineprüfung/3250 V DC Typprüfung

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

ACHTUNG

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, der Leistungseingang für die Relaispule eines SM oder die Versorgung eines nicht potentialgetrennten Analogeingangs. Alle nicht potentialgetrennten M-Klemmen müssen an dasselbe externe Bezugspotential angeschlossen werden.

 **WARNUNG**

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 unvorhersehbares Betriebsverhalten verursachen, was zu Tod oder schwerer Körperverletzung und/oder Sachschaden führen kann.

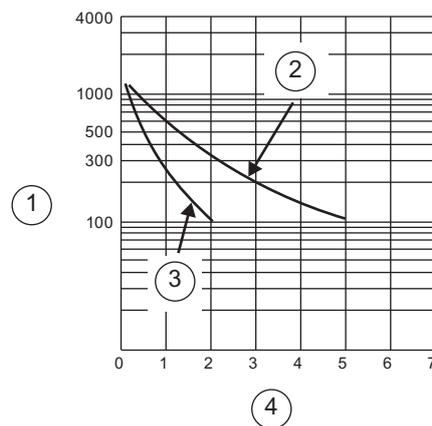
Schließen Sie stets alle nicht potentialgetrennten M-Klemmen in einem S7-1200 System an dasselbe Bezugspotential an.

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 von Relais-Herstellern zur Verfügung gestellt werden, sind nachstehend aufgeführt. Die tatsächliche Leistungsfähigkeit richtet sich nach der jeweiligen Verwendung. Ein externer Schutzkreis, der der Last angepasst ist, verlängert die Lebensdauer der Kontakte.



- ① Lebensdauer ($\times 10^3$ Operationen)
- ② Widerstandslast 250 V AC
Widerstandslast 30 V DC
- ③ Induktive Last 250 V AC (p.f.=0,4)
Induktive Last 30 V DC (L/R=7ms)
- ④ Nennbetriebsstrom (A)

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/support-request>).

Tabelle A- 8 Allgemeine technische Daten

Allgemeine technische Daten		CPU 1211C	CPU 1212C	CPU 1214C
Abmessungen (B x H x T)		90 x 100 x 75 (mm)	90 x 100 x 75 (mm)	110 x 100 x 75 (mm)
Gewicht	• AC/DC/Relais	• 420 Gramm	• 425 Gramm	• 475 Gramm
	• DC/DC/Relais	• 380 Gramm	• 385 Gramm	• 435 Gramm
	• DC/DC/DC	• 370 Gramm	• 370 Gramm	• 415 Gramm
Leistungsverlust	• AC/DC/Relais	• 10 W	• 11 W	• 14 W
	• DC/DC/Relais	• 8 W	• 9 W	• 12 W
	• DC/DC/DC	• 8 W	• 9 W	• 12 W
Verfügbarer Strom (5 VDC) für SM- und CM-Bus		max. 750 mA	max. 1000 mA	max. 1600 mA
Verfügbarer Strom (24 VDC) Geberspannung		max. 300 mA	max. 300 mA	max. 400 mA
Stromaufnahme digitaler Eingang (24 V DC)		4 mA/Eingang	4 mA/Eingang	4 mA/Eingang

Tabelle A-9 CPU-Merkmale

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C
Anwenderspeicher			
• Arbeitsspeicher	• 25 KB	• 25 KB	• 50 KB
• Ladespeicher	• 1 MB	• 1 MB	• 2 MB
• Remanenter Speicher	• 2 KB	• 2 KB	• 2 KB
Integrierte digitale E/A	6 Eingänge	8 Eingänge	14 Eingänge
Siehe technische Daten (Seite 266).	4 Ausgänge	6 Ausgänge	10 Ausgänge
Integrierte analoge E/A	2 Eingänge	2 Eingänge	2 Eingänge
Siehe technische Daten (Seite 274).			
Größe des Prozessabbilds			
• Eingänge	• 1024 Byte	• 1024 Byte	• 1024 Byte
• Ausgänge	• 1024 Byte	• 1024 Byte	• 1024 Byte
Merker (M)	4096 Byte	4096 Byte	8192 Byte
Temporärer (lokaler) Speicher	<ul style="list-style-type: none"> • 16 KB für Anlauf und Programmzyklus (einschließlich zugehöriger FBs und FCs). • 4 KB für Standardalarmereignisse einschließlich FBs und FCs • 4 KB für Fehleralarmereignisse einschließlich FBs und FCs 		
Zusätzliche SM-Module	Keine	max. 2 SMs	max. 8 SMs
Zusätzliche SB	max. 1 SB	max. 1 SB	max. 1 SB
Zusätzliche CM	max. 3 CM/CP	max. 3 CM/CP	max. 3 CM/CP
Schnelle Zähler	3 gesamt	4 gesamt	6 gesamt
• Einphasenzähler (Taktfrequenz)	• 3 bei 100 kHz	• 3 bei 100 kHz und 1 bei 30 kHz	• 3 bei 100 kHz und 3 bei 30 kHz
• A/B-Zähler (Taktfrequenz)	• 3 bei 80 kHz	• 3 bei 80 kHz und 1 bei 20 kHz	• 3 bei 80 kHz und 3 bei 20 kHz
Impulsausgänge ¹	2	2	2
Eingänge für Impulsabgriff	6	8	14
Verzögerungs-/Weckalarme	4 gesamt mit Auflösung von 1 ms	4 gesamt mit Auflösung von 1 ms	4 gesamt mit Auflösung von 1 ms
Flankenalarme	6 steigend und 6 fallend	8 steigend und 8 fallend	12 steigend und 12 fallend
Optional mit SB	10 steigend und 10 fallend	12 steigend und 12 fallend	16 steigend und 16 fallend

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C
Echtzeituhr			
<ul style="list-style-type: none"> Genauigkeit Pufferung (wartungsfreier Hochleistungskondensator) 	<ul style="list-style-type: none"> +/- 60 Sekunden/Monat Typ. 10 Tage/min. 6 Tage bei 40 °C 	<ul style="list-style-type: none"> +/- 60 Sekunden/Monat Typ. 10 Tage/min. 6 Tage bei 40 °C 	<ul style="list-style-type: none"> +/- 60 Sekunden/Monat Typ. 10 Tage/min. 6 Tage bei 40 °C
Ausführungsgeschwindigkeit			
<ul style="list-style-type: none"> Boolescher Wert Wort übertragen Realzahlenarithmetik 	<ul style="list-style-type: none"> 0,1 µs/Anweisung 12 µs/Anweisung 18 µs/Anweisung 	<ul style="list-style-type: none"> 0,1 µs/Anweisung 12 µs/Anweisung 18 µs/Anweisung 	<ul style="list-style-type: none"> 0,1 µs/Anweisung 12 µs/Anweisung 18 µs/Anweisung

¹ Bei CPU-Varianten mit Relaisausgängen müssen Sie ein digitales Signalboard (SB) installieren, um die Impulsausgänge zu verwenden.

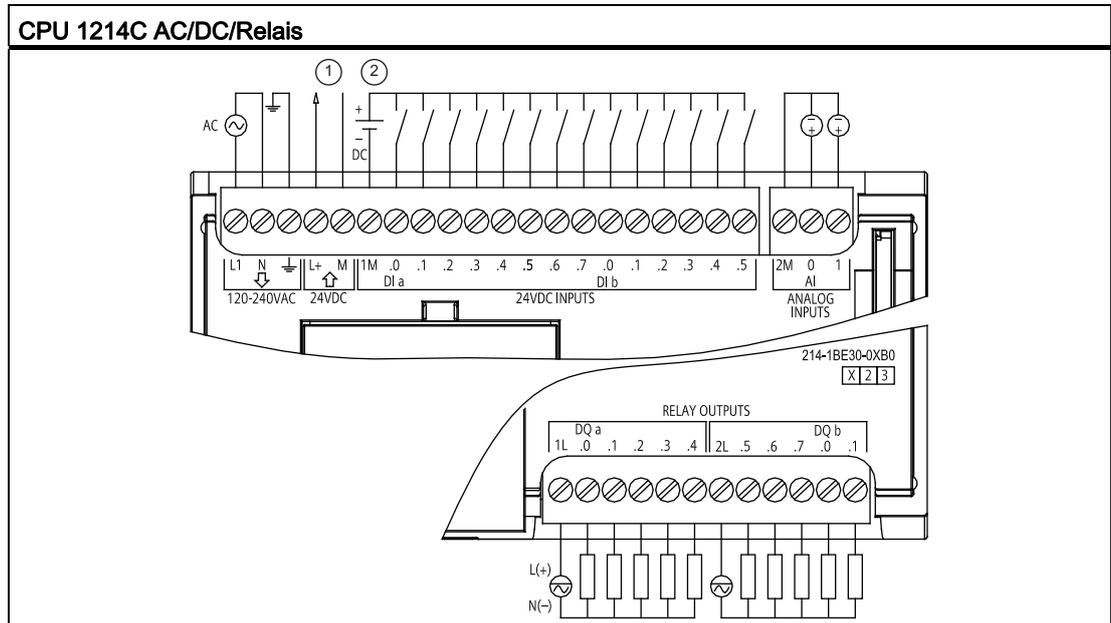
Tabelle A- 10 Kommunikation

Technische Daten	CPU 1211C	CPU 1212C	CPU 1214C
Kommunikation	1 Ethernet-Port	1 Ethernet-Port	1 Ethernet-Port
<ul style="list-style-type: none"> Datenraten Elektrische Trennung (externes Signal zu PLC-Logik) Kabelart 	<ul style="list-style-type: none"> 10/100 MBit/s Wandler potentialgetrennt, 1500 V DC CAT5e geschirmt 	<ul style="list-style-type: none"> 10/100 MBit/s Wandler potentialgetrennt, 1500 V DC CAT5e geschirmt 	<ul style="list-style-type: none"> 10/100 MBit/s Wandler potentialgetrennt, 1500 V DC CAT5e geschirmt
Geräte	<ul style="list-style-type: none"> 3 HMI¹ 1 PG 	<ul style="list-style-type: none"> 3 HMI¹ 1 PG 	<ul style="list-style-type: none"> 3 HMI¹ 1 PG
Ethernet-Verbindungen²	8 (aktiv oder passiv)	8 (aktiv oder passiv)	8 (aktiv oder passiv)
CPU-zu-CPU-Kommunikation über S7 (GET/PUT)	<ul style="list-style-type: none"> 8 (Client) 3 (Server) 	<ul style="list-style-type: none"> 8 (Client) 3 (Server) 	<ul style="list-style-type: none"> 8 (Client) 3 (Server)

¹ Die CPU bietet zweckbestimmte HMI-Verbindungen, um bis zu 3 HMI-Geräte zu unterstützen. (Sie können bis zu 2 SIMATIC Comfort Panels haben.) Die Gesamtzahl der HMI-Geräte wird von den Typen der HMI-Panels in Ihrer Konfiguration beeinflusst. Sie können beispielsweise bis zu drei SIMATIC Basic Panels an Ihre CPU anschließen, oder Sie können bis zu zwei SIMATIC Comfort Panels und ein zusätzliches Basic Panel anschließen.

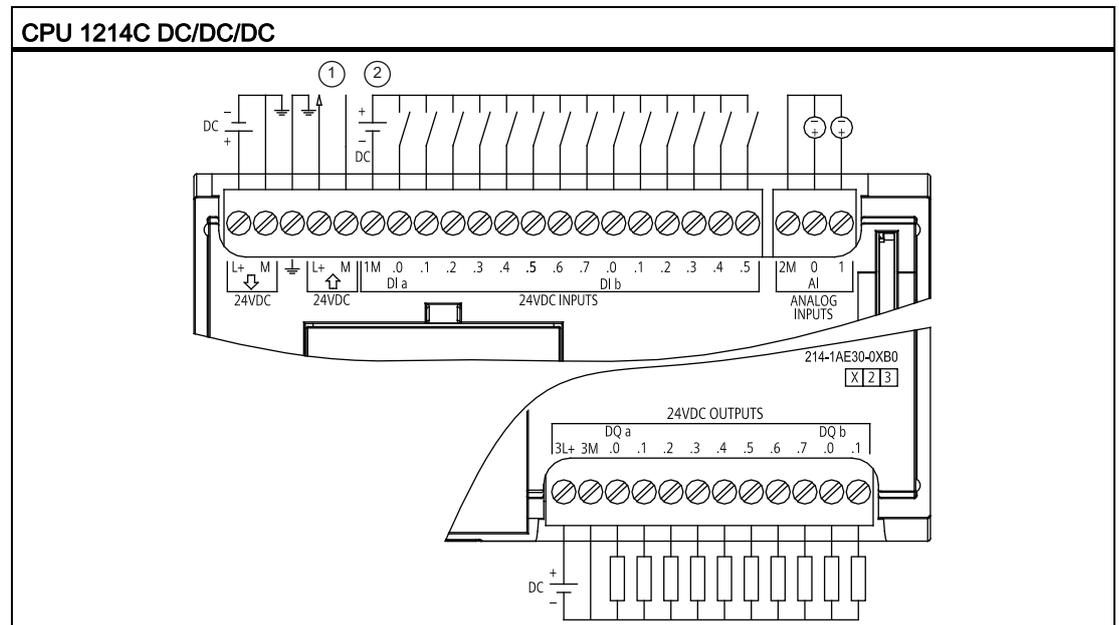
² Verbindungen für die offene Benutzerkommunikation (aktiv oder passiv): TSEND_C, TRCV_C, TCON, TDISCON, TSEND und TRCV.

Tabelle A- 11 Schaltplan der CPU 1214C AC/DC/Relais



- ① 24-V-DC-Geberversorgung. Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.
- ① Bei stromziehenden Eingängen "-" an "M" anschließen (gezeigt). Bei stromliefernden Eingängen "+" an "M" anschließen.

Tabelle A- 12 Schaltplan der CPU 1214C DC/DC/DC



- ① 24-V-DC-Geberverspannung. Um zusätzliche Störfestigkeit zu erreichen, schließen Sie "M" an Masse an, auch wenn Sie keine Geberversorgung verwenden.
- ① Bei stromziehenden Eingängen "-" an "M" anschließen (gezeigt). Bei stromliefernden Eingängen "+" an "M" anschließen.

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/support-request>).

A.3.1 SB 1221, SB 1222 und SB 1223 Digitaleingabe/-ausgabe (DI, DO und DI/DO)

Tabelle A- 13 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)	38 x 62 x 21 (mm)	38 x 62 x 21 (mm)
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

Allgemeines		SB 1221 4 DI (200 kHz)	SB 1222 4 DO (200 kHz)
Stromaufnahme	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- 14 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 	6ES7 223-0BD30-0XB0
Abmessungen (B x H x T)		38 x 62 x 21 (mm)	38 x 62 x 21 (mm)
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	35 mA	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 	4 mA / Eingang
Eingänge/Ausgänge		2 Eingänge (stromliefernd) 2 Ausgänge (elektronisch, MOSFET)	2 Eingänge (IEC Typ 1 stromziehend) 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 266).

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- 15 Schaltpläne der digitalen SBs

SB 1221 Eingabemodul	SB 1222 Ausgabemodul	SB 1223 Eingabe-/Ausgabemodul
<p>SB 1221 DI 4 (200 kHz)</p>	<p>SB 1222 DO 4 (200 kHz)</p>	<p>SB 1223 DI 2 / DO2 (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.</p>	<p>① Unterstützt nur stromliefernde Eingänge. ② Bei stromliefernden Ausgängen "Load" an "-" anschließen (s. Abbildung). Bei stromziehenden Ausgängen "Load" an "+" anschließen.</p>

Hinweis

Die schnellen (200 kHz) SBs (SB 1221 und SB 1223) unterstützen nur stromziehende Eingänge. Das Standard-SB 1223 unterstützt nur stromliefernde Eingänge.

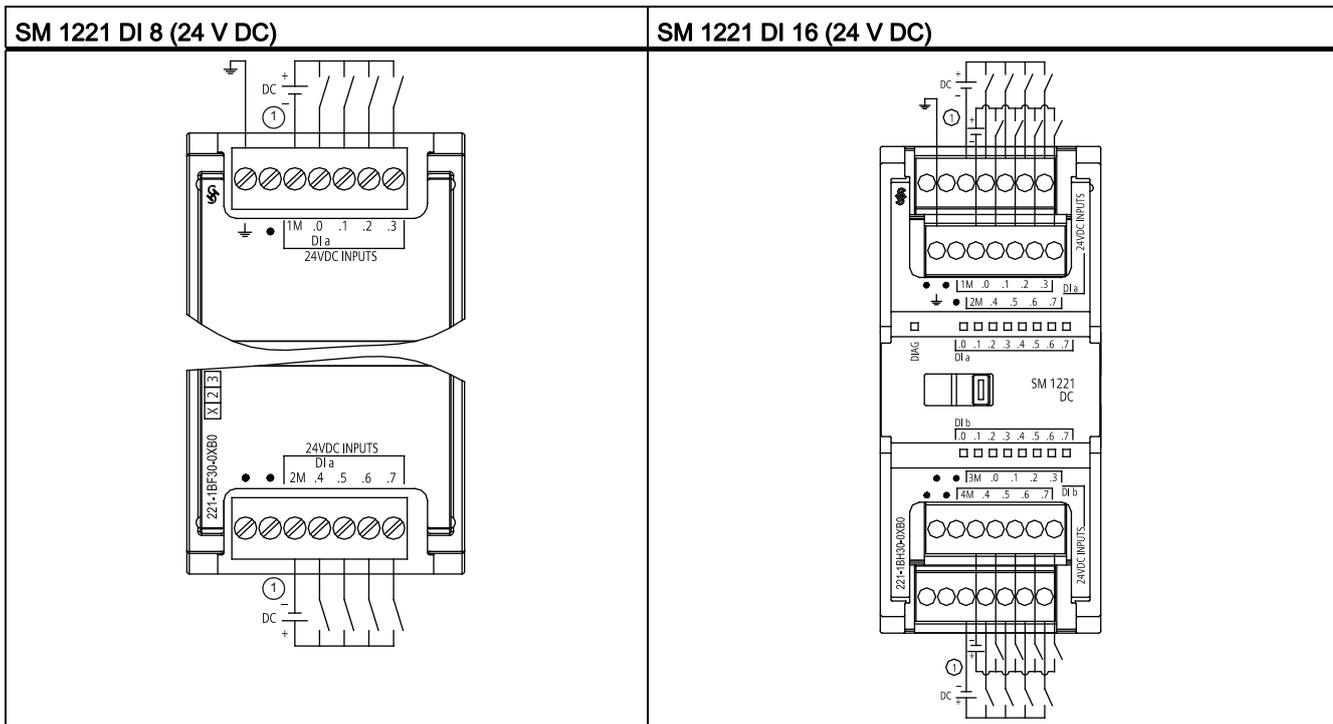
Die schnellen (200 kHz) Ausgänge (SB 1222 und SB 1223) können stromliefernd oder stromziehend sein. Bei stromliefernden Ausgängen "Load" an "-" anschließen (gezeigt). Bei stromziehenden Ausgängen "Load" an "+" anschließen. Weil sowohl stromliefernde als auch stromziehende Konfigurationen von denselben 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- 16 SM 1221 Digitaleingabe (DI)

Technische Daten	SM 1221 DI 8 (24 V DC)	SM 1221 DI 16 (24 V DC)
Bestellnummer	6ES7 221-1BF30-0XB0	6ES7 221-1BH30-0XB0
Anzahl der Eingänge (DI)	8	16
Siehe technische Daten (Seite 266).		
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- 17 Schaltplan der Digitaleingabemodule SM 1221 (DI)



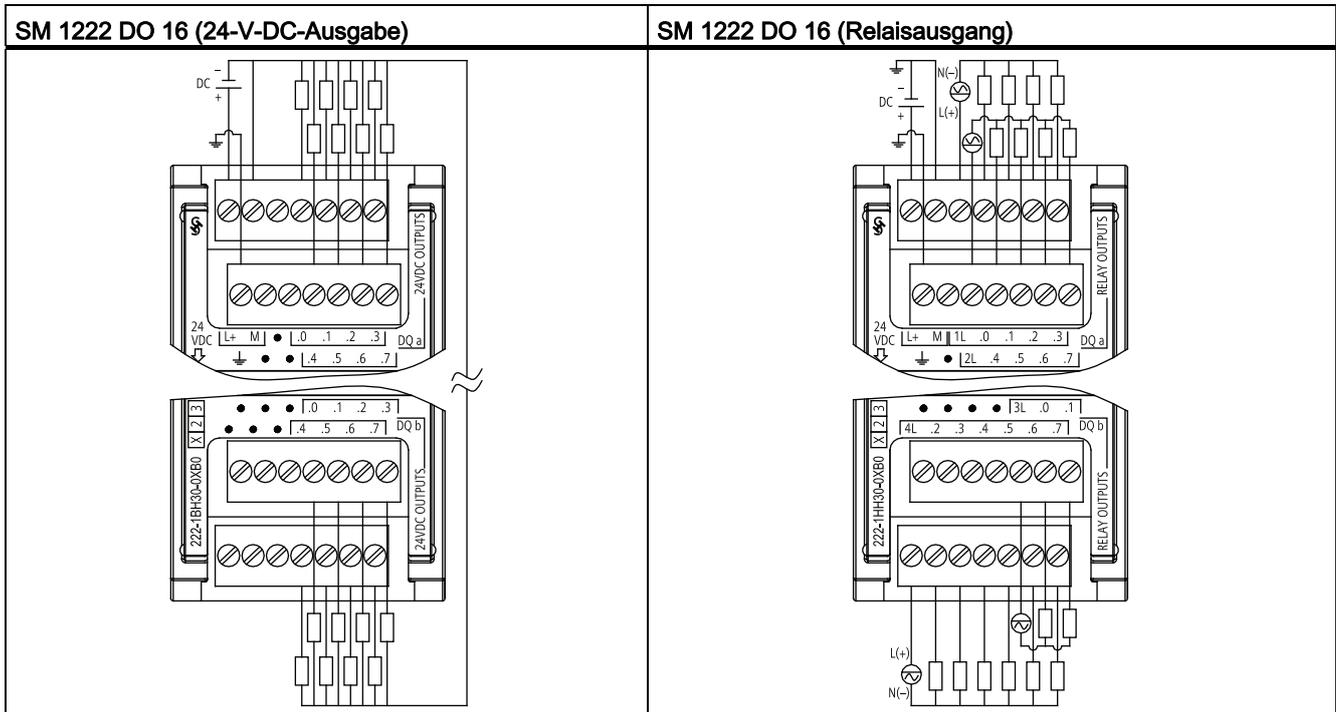
① Bei stromziehenden Eingängen "-" an "M" anschließen (s. Abbildung). Bei stromliefernden Eingängen "+" an "M" anschließen.

A.3.3 SM 1222 Digitalausgabe (DO)

Tabelle A- 18 SM 1222 Digitalausgabe (DO)

Technische Daten		SM 1222 DO (Relais)	SM1222 DO (24 V DC)
Bestellnummer		<ul style="list-style-type: none"> • DO 8: 6ES7 222-1HF30-0XB0 • DO 8 Umschaltung: 6ES7 222-1XF30-0XB0 • DO 16: 6ES7 222-1HH30-0XB0 	<ul style="list-style-type: none"> • DO 8: 6ES7 222-1BF30-0XB0 • DO 16: 6ES7 222-1BH30-0XB0
Anzahl der Ausgänge (DO) Siehe technische Daten (Seite 266).		<ul style="list-style-type: none"> • 8 (DO 8 und DO 8 Umschaltung) • 16 (DO 16) 	8 (DO 8) oder 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
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/Relaisspule • DO 8 Umschaltung: 16,7 mA/Relaisspule 	-/-

Tabelle A- 19 Schaltplan der Digitalausgabemodule SM 1222 (DO)

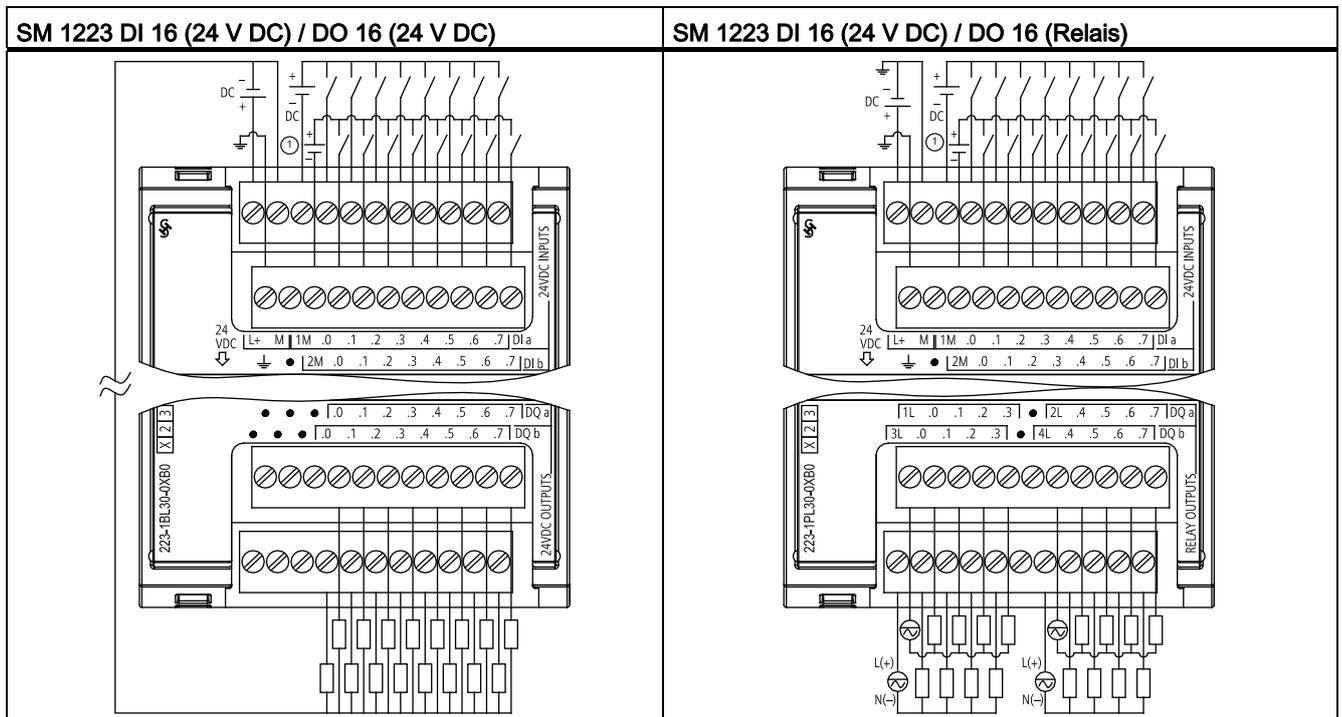


A.3.4 SM 1223 V-DC-Digitaleingabe/-ausgabe (DI/DO)

Tabelle A- 20 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 223-1PH30-0XB0 DI 16 / DO 16: 6ES7 223-1PL30-0XB0	DI 8 / DO 8: 6ES7 223-1BH30-0XB0 DI 8 / DO 8: 6ES7 223-1BL30-0XB0
Anzahl der Eingänge/Ausgänge (DI/DO) Siehe technische Daten (Seite 266).	<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	<p>SM-Bus</p> <ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 180 mA 	<ul style="list-style-type: none"> DI 8 / DO 8: 145 mA DI 16 / DO 16: 185 mA
	<p>24 V DC</p> <ul style="list-style-type: none"> 4 mA / Eingang 11 mA / Relaispule 	<ul style="list-style-type: none"> 4 mA / Eingang

Tabelle A- 21 Schaltpläne des SM 1223 Kombinationsmoduls DI/DO



① 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

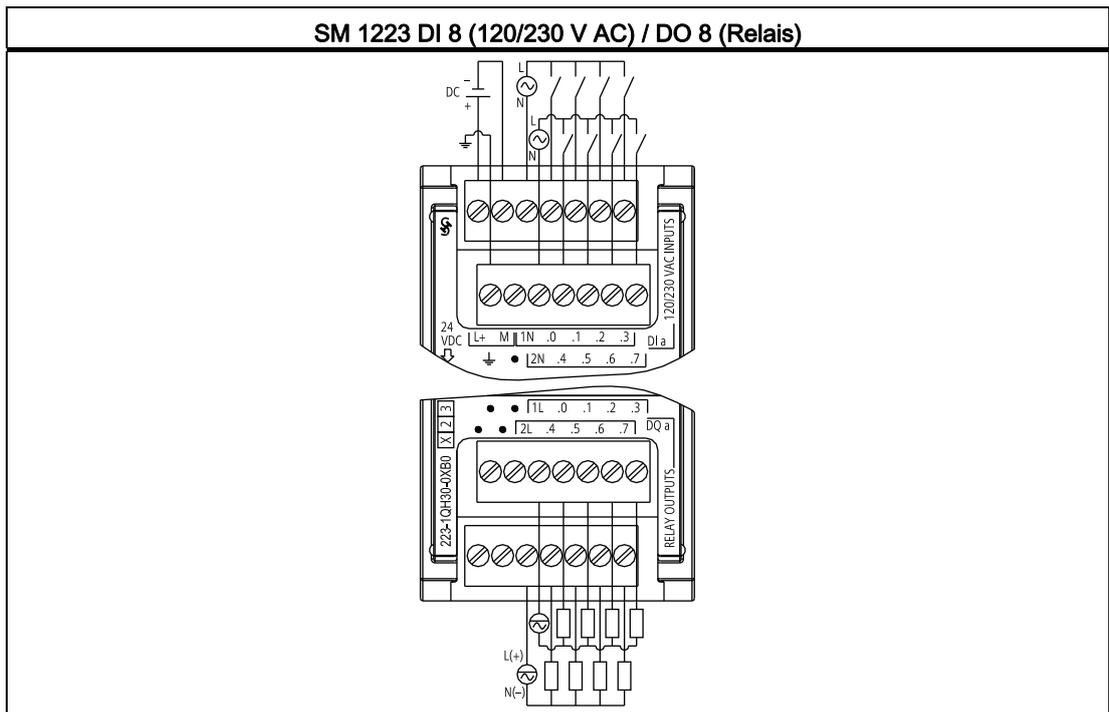
Tabelle A- 22 SM 1223 Kombinationsmodul V-AC-Digitaleingabe/-ausgabe (DI/DO)

Technische Daten	SM 1223 DI (120/230 V AC) / DO (Relais)
Bestellnummer	DI 8 / DO 8: 6ES7 223-1QH30-0XB0
Anzahl der Eingänge/Ausgänge (DI/DO)	Eingänge: 8 (120/230 V AC) Siehe technische Daten für 120/230 V AC-Eingabe (Seite 268). Ausgänge: 8 (Relais) Siehe technische Daten für die Digitalausgabe (Seite 268).
Abmessungen B x H x T (mm)	45 x 100 x 75
Gewicht	190 Gramm
Leistungsverlust	5,5 W
Stromaufnahme	SM-Bus 145 mA
	24 V DC 11 mA / Relaisspule

Hinweis

Das Signalmodul SM 1223 DI 8 x 120/230 V AC, DO 8 x Relais (6ES7 223-1QH30-0XB0) ist für den Einsatz in Umgebungen der Class 1, Division 2, Gas Group A, B, C, D, Temperature Class T4 Ta = 40 °C zugelassen.

Tabelle A- 23 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- 24 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 1223200 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	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

ACHTUNG

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- 25 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)

A.4 Technische Daten der digitalen Eingänge und Ausgänge

Technische Daten	Einphasenzähler	A/B-Zähler
CPU 1214C	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)
Schnelles SB (200 kHz)	200 kHz	160 kHz
Standard-SB (20 kHz)	30 kHz	20 kHz

¹ Pegel logisch 1 = 15 bis 26 V DC

A.4.2 120/230 V-AC-Digitaleingabe

Tabelle A- 26 120/230 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- 27 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

Technische Daten	Relais (CPU und SM)	24 V DC (CPU, SM und SB)	200 kHz 24 V DC (SB)
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 • 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 	⁶
Isolationswiderstand	min. 100 MΩ, wenn neu	-/-	-/-
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	-/-	-/-
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

A.4 Technische Daten der digitalen Eingänge und Ausgänge

Technische Daten	Relais (CPU und SM)	24 V DC (CPU, SM und SB)	200 kHz 24 V DC (SB)
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 µs von Aus nach Ein; max. 10 µs von Ein nach Aus SM: max. 50 µs von Aus nach Ein; max. 200 µs von Ein nach Aus	1,5 µs + 300 ns steigend ¹ 1,5 µs + 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 Je nach Impulsempfänger und Kabel kann ein zusätzlicher Lastwiderstand (bei mindestens 10% des Nennstroms) die Qualität der Impulssignale und die Störfestigkeit verbessern.

⁶ 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/support-request>).

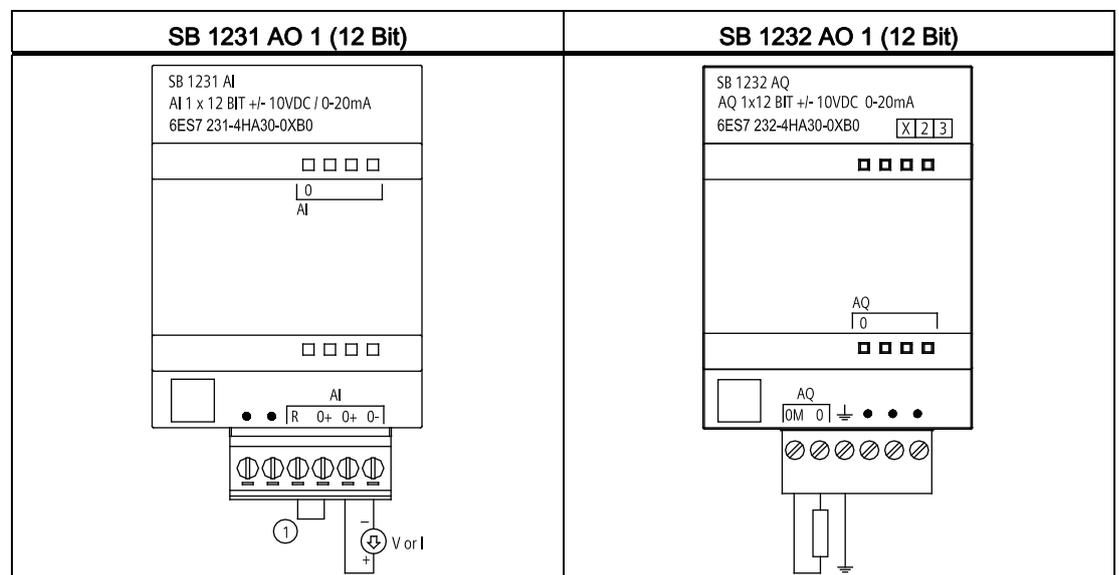
A.5.1 SB 1231 und SB 1232 Analogeingabe (AI) und Analogausgabe (AO)

Tabelle A- 28 Allgemeine technische Daten

Technische Daten	SB 1231 AI 1 (12 Bit) ¹	SB 1232 AO 1 (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- 29 Schaltpläne der analogen SBs



① "R" und "0+" für Strom anschließen.

A.5.2 SM 1231 Analogeingabe (AI)

Tabelle A- 30 SM 1231 Analogeingabe (AI)

Technische Daten	SM 1231 AI 4 (13 Bit)	SM 1231 AI 8 (13 Bit)
Bestellnummer (MLFB)	6ES7 231-4HD30-0XB0	6ES7 231-4HF30-0XB0
Anzahl der Eingänge	4 Eingänge (AI)	8 Eingänge (AI)
Typ	Spannung oder Strom (differential), wählbar in Gruppen zu je 2	Spannung oder Strom (differential), wählbar in Gruppen zu je 2
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	90 mA
Stromaufnahme (24 V DC)	45 mA	45 mA

A.5.3 SM 1232 Analogausgabe (AO)

Tabelle A- 31 SM 1232 Analogausgabe (AO)

Technische Daten	SM 1232 AO 2 (14 Bit)	SM 1232 AO 4 (14 Bit)
Bestellnummer (MLFB)	6ES7 232-4HB30-0XB0	6ES7 232-4HD30-0XB0
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)

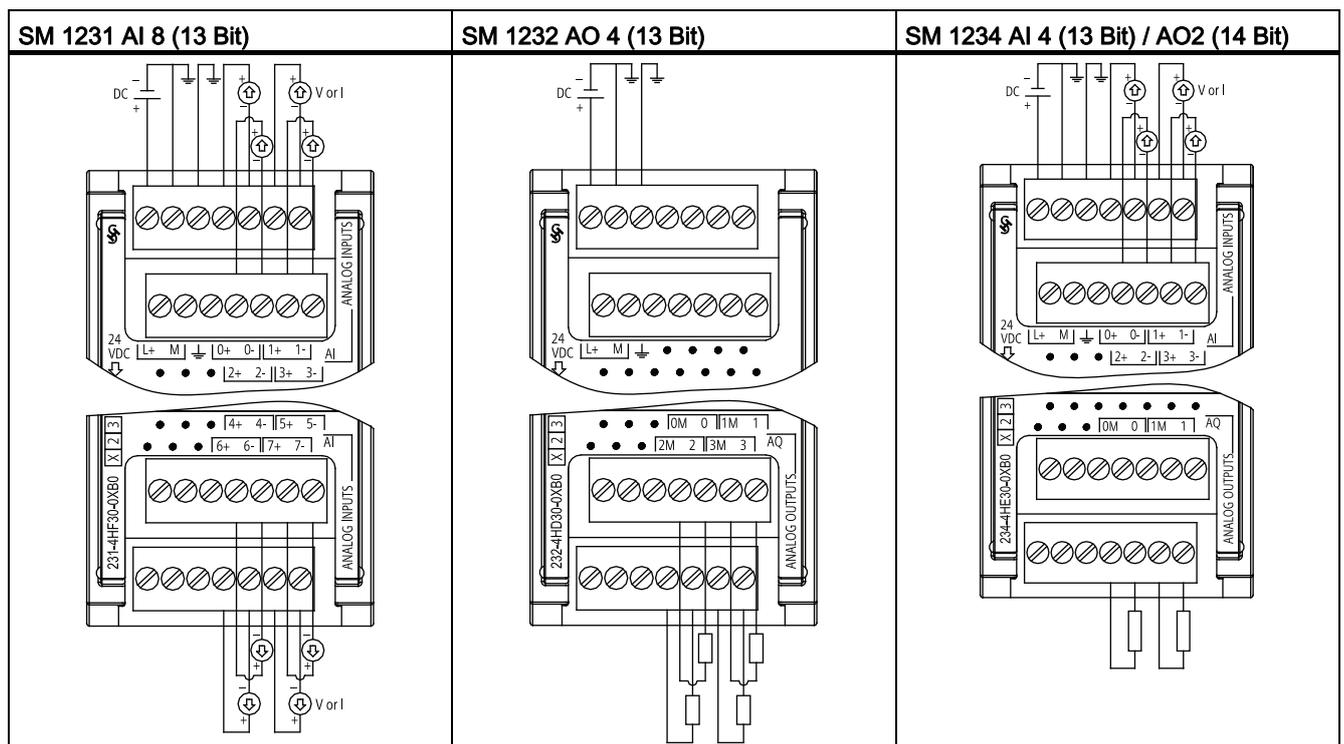
Tabelle A- 32 SM 1234 Kombinationsmodul Analogeingabe/-ausgabe (AI/AO)

Technische Daten	SM 1234 AI 4 (13 Bit) / AO 2 (14 Bit)
Bestellnummer (MLFB)	6ES7 234-4HE30-0XB0
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

Technische Daten	SM 1234 AI 4 (13 Bit) / AO 2 (14 Bit)
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- 33 Schaltpläne der analogen SMs



Hinweis

Nicht verwendete analoge Eingänge sollten eingefügt werden.

Wenn für die Eingänge der Modus "Strom" konfiguriert sind, fließt kein Strom durch den Eingang, es sei denn, Sie versorgen das Modul mit externer Spannung.

A.6 Technische Daten der analogen E/A

A.6.1 Technische Daten der analogen Eingänge (CPU, SM und SB)

Tabelle A- 34 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, $\pm 2,5$ oder 0 bis 20 mA	± 10 V, ± 5 V, $\pm 2,5$ V oder 0 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 / 0 bis 55 °C)	3,0 % / 3,5 % des Vollausschlags	$\pm 0,3$ % / $\pm 0,6$ % des Vollausschlags	$\pm 0,1$ % / $\pm 0,2$ % 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 (Signal-plus Gleichtaktspannung)	Kleiner als +12 V und größer als -12 V	Kleiner als +35 V und größer als -35 V	Kleiner als +12 V und größer als -12 V
Lastimpedanz	Differenz: ≥ 100 k Ω	Differenz: 220 k Ω (Spannung), 250 Ω (Strom) Gleichtakt: 55 k Ω (Spannung), 55 k Ω (Strom)	Differenz: 9 M Ω (Spannung), 250 Ω (Strom) Gleichtakt: 4,5 M Ω (Spannung), 4,5 M Ω (Strom)

Technische Daten	CPU	SB	SM
Elektrische Trennung (Feld zu Logik)	Keine	Keine	Keine
Leitungslänge (Meter)	100 m, geschirmtes, verdrilltes Leiterpaar	100 m, verdrillt und geschirmt	100 m, verdrillt und geschirmt
Diagnose	Überlauf/Unterlauf	Überlauf/Unterlauf	Überlauf/Unterlauf (siehe Hinweis 3) 24 V DC Niederspannung

Hinweis 1: Die Bereiche für Überspringen/Unterspringen und Überlauf/Unterlauf ermitteln Sie anhand der Messbereiche der analogen Ausgänge für Spannung und Strom (Seite 275).

Hinweis 2: Die Werte für Glättung und Rauschunterdrückung ermitteln Sie anhand der Schrittantwortzeiten (Seite 276).

Hinweis 3: Beim SM 1231 AI 4 (13 Bit) gilt: Wird am Eingang eine Spannung größer als +30 V DC oder kleiner als -15 V DC angelegt, ist der resultierende Wert unbekannt und der entsprechende Überlauf oder Unterlauf ist möglicherweise nicht aktiv.

A.6.2 Eingangsmessbereiche (AI) für Spannung und Strom

Tabelle A- 35 Darstellung Analogeingang für Spannung

System		Messbereich Spannung					
Dezimal	Hexadezimal	±10 V	±5 V	±2,5 V		0 bis 10 V	
32767	7FFF	11,851 V	5,926 V	2,963 V	Überlauf	11,851 V	Überlauf
32512	7F00						
32511	7EFF	11,759 V	5,879 V	2,940 V	Überschwingbereich	11,759 V	Überschwingbereich
27649	6C01						
27648	6C00	10 V	5 V	2,5 V	Bemessungsbereich	10 V	Bemessungsbereich
20736	5100	7,5 V	3,75 V	1,875 V		7,5 V	
1	1	361,7 µV	180,8 µV	90,4 µV		361,7 µV	
0	0	0 V	0 V	0 V		0 V	
-1	FFFF				Unterschwingbereich	Negative Werte werden nicht unterstützt	
-20736	AF00	-7,5 V	-3,75 V	-1,875 V			
-27648	9400	-10 V	-5 V	-2,5 V			
-27649	93FF						
-32512	8100	-11,759 V	-5,879 V	-2,940 V	Unterlauf		
-32513	80FF						
-32768	8000	-11,851 V	-5,926 V	-2,963 V			

A.6.3 Schrittantwort der analogen Eingänge (AI)

Die folgende Tabelle zeigt die Schrittantwortzeiten der analogen Eingänge (AI) von CPU, SB und SM.

Tabelle A- 36 Schrittantwort (ms) der analogen Eingänge

Auswahl der Glättung (Mittelwertbildung aus Abtastwerten)		Auswahl Integrationszeit ¹			
		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
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
	SM				
	• (4 Kanäle)	• 0.625	• 4.17	• 5	• 25
	• (8 Kanäle)	• 1.25	• 4.17	• 5	• 25

¹ 0 V bis 10 V, gemessen bei 95 % (CPU und SB), 0 bis Vollausschlag, gemessen bei 95 % (SM),

A.6.4 Abtastzeit und Aktualisierungszeiten der Analogeingänge

Tabelle A- 37 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- 38 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.6.5 Technische Daten der analogen Ausgänge (SB und SM)

Tabelle A- 39 Technische Daten der analogen Ausgänge (AO)

Technische Daten	SB	SM
Typ	Spannung oder Strom	Spannung oder Strom
Bereich	± 10 V oder 0 bis 20 mA	± 10 V oder 0 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 / 0 bis 55 °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 278) im Vollausschlagsbereich.		

A.6.6 Ausgangsmessbereiche (AO) für Spannung und Strom

Tabelle A- 40 Darstellung Analogausgang für Strom

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	
1	1	723,4 nA	
0	0	0 mA	

¹ Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

Tabelle A- 41 Darstellung Analogausgang für Spannung

System		Spannungsausgangsbereich	
Dezimal	Hexadezimal	±10 V	
32767	7FFF	Siehe Hinweis 1	Überlauf
32512	7F00	Siehe Hinweis 1	
32511	7EFF	11,76 V	Überschwingbereich
27649	6C01		
27648	6C00	10 V	Bemessungsbereich
20736	5100	7,5 V	
1	1	361,7 µV	
0	0	0 V	
-1	FFFF	-361,7 µV	
-20736	AF00	-7,5 V	
-27648	9400	-10 V	
-27649	93FF		Unterschwingbereich
-32512	8100	-11,76 V	
-32513	80FF	Siehe Hinweis 1	Unterlauf
-32768	8000	Siehe Hinweis 1	

¹ Bei Überlauf oder Unterlauf verhalten sich die Analogausgänge entsprechend den Eigenschaften der Gerätekonfiguration des Analogsignalmoduls. Wählen Sie für den Parameter "Reaktion auf CPU-STOP" entweder: Ersatzwert aufschalten oder Letzten Wert halten.

A.7 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.

Wenn RTD-Bereiche von 500 Ω und 1000 Ω mit anderen Widerständen niederen Werts verwendet werden, kann sich der Fehler auf den zweifachen spezifizierten Fehler erhöhen. Optimale Genauigkeit für die 10 Ω -RTD-Bereiche ermöglichen 4-Leiter-Anschlüsse.

Der Widerstand der Anschlussleitungen im 2-Leiter-Modus verursacht einen Fehler der Gebermessung. Die Messgenauigkeit ist daher nicht mehr gewährleistet.

ACHTUNG
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.7.1 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- 42 Allgemeine technische Daten

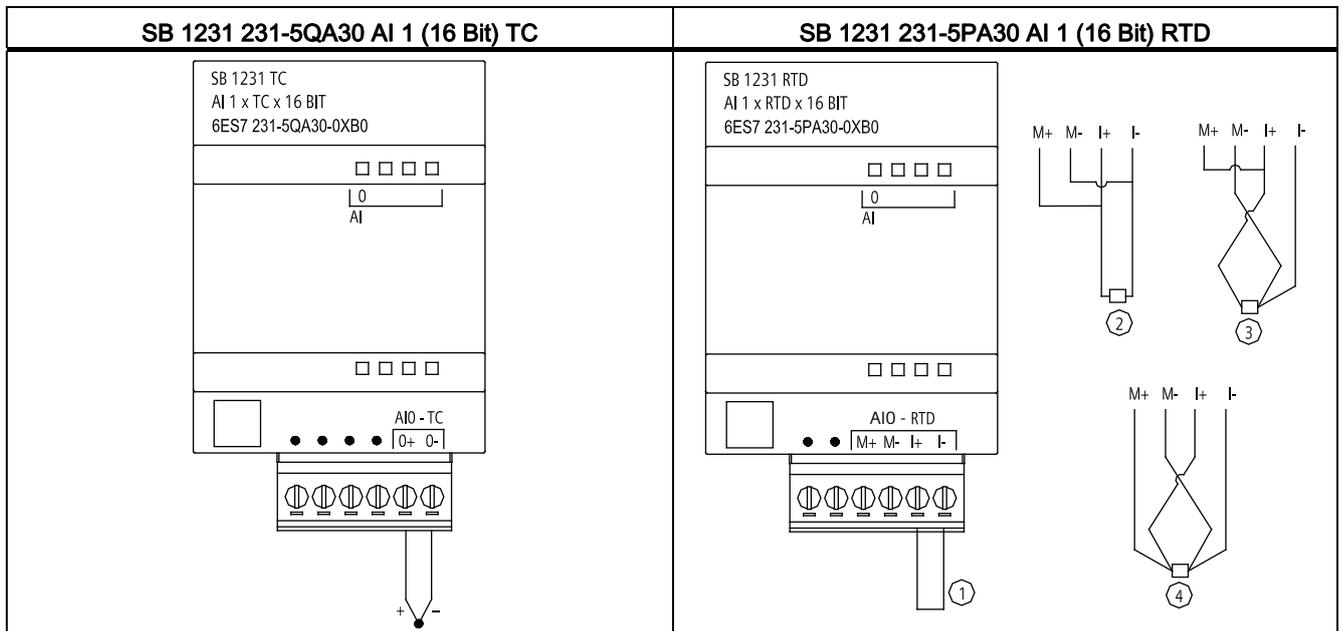
Technische Daten	SB 1231 AI 1 (16 Bit) TC	SB 1231 AI 1 (16 Bit) RTD
Bestellnummer	6ES7 231-5QA30-0XB0	6ES7 231-5PA30-0XB0
Abmessungen B x H x T (mm)	38 x 62 x 21 mm	38 x 62 x 21 mm
Gewicht	35 Gramm	35 Gramm
Leistungsverlust	0,5 W	0,7 W
Stromaufnahme (SM-Bus)	5 mA	5 mA
Stromaufnahme (24 V DC)	20 mA	25 mA
Anzahl der Eingänge (Seite 284)	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- 43 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.7.2 Technische Daten SM 1231 RTD

Tabelle A- 44 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 231-5PD30-0XB0	6ES7 231-5PF30-0XB0
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

Technische Daten	SM 1231 AI 4 x RTD x 16 Bit	SM 1231 AI 8 x RTD x 16 Bit
Anzahl der Eingänge (Seite 284)	4	8
Typ	Modulreferenz RTD und Ω	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- 45 Schaltpläne der RTD-SMs

SM 1231 RTD 4 (16 Bit)	SM 1231 RTD 8 (16 Bit)	Referenzen
		<p>① Nicht belegte RTD-Eingänge zurückschleifen</p> <p>② 2Leiter-RTD</p> <p>③ 3Leiter-RTD</p> <p>④ 4Leiter-RTD</p>

A.7.3 Technische Daten SM 1231 TC

Tabelle A- 46 Allgemeine technische Daten

Modell	SM 1231 AI4 x 16 Bit TC	SM 1231 AI8 x 16 Bit TC
Bestellnummer	6ES7 231-5QD30-0XB0	6ES7 231-5QF30-0XB0
Abmessungen B x H x T (mm)	45 x 100 x 75	45 x 100 x 75
Gewicht	180 Gramm	xxx Gramm

Modell	SM 1231 AI4 x 16 Bit TC	SM 1231 AI8 x 16 Bit TC
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 284)	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- 47 Schaltpläne der TC-SMs

SM 1231 AI 4 TC (16 Bit)	SM 1231 AI 8 TC (16 Bit)	Hinweise
		<p>① SM 1231 AI 8 TC: Die Anschlüsse von TC 2, 3, 4 und 5 werden zur besseren Übersichtlichkeit weggelassen.</p>

A.7.4 Technische Daten der analogen Eingänge für RTD und TC (SM und SB)

Tabelle A- 48 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 286) • TC (Seite 285)
Auflösung	Temperaturbereich	0,1 °C/0,1 °F
	Widerstand/Spannung	15 Bit plus Vorzeichen
Max. Stehspannung		± 35 V
Rauschunterdrückung		85 dB für die gewählte Filtereinstellung (10 Hz, 50 Hz, 60 Hz oder 400 Hz)
Gleichtaktunterdrückung		> 120 dB bei 120 V AC
Impedanz		≥ 10 M Ω
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 / 0 bis 55 °C)		Siehe RTD/TC-Typtabellen:
		<ul style="list-style-type: none"> • RTD (Seite 286) • TC (Seite 285)
Wiederholgenauigkeit		$\pm 0,05$ % 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 287) • TC (Seite 285)
Fehler kalte Verbindungsstelle		<ul style="list-style-type: none"> • RTD: -/- • TC: $\pm 1,5$ °C

Technische Daten	RTD und Thermoelement (TC)
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.7.5 Thermoelementtyp

Tabelle A- 49 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 0 °C bis 55 °C
J	-210,0 °C	-150,0 °C	1200,0 °C	1450,0 °C	$\pm 0,3$ °C	$\pm 0,6$ °C
K	-270,0 °C	-200,0 °C	1372,0 °C	1622,0 °C	$\pm 0,4$ °C	$\pm 1,0$ °C
T	-270,0 °C	-200,0 °C	400,0 °C	540,0 °C	$\pm 0,5$ °C	$\pm 1,0$ °C
E	-270,0 °C	-200,0 °C	1000,0 °C	1200,0 °C	$\pm 0,3$ °C	$\pm 0,6$ °C
R & S	-50,0 °C	100,0 °C	1768,0 °C	2019,0 °C	$\pm 1,0$ °C	$\pm 2,5$ °C
N	-270,0 °C	-200,0 °C	1300,0 °C	1550,0 °C	$\pm 1,0$ °C	$\pm 1,6$ °C
C	0,0 °C	100,0 °C	2315,0 °C	2500,0 °C	$\pm 0,7$ °C	$\pm 2,7$ °C
TXK/XK (L)	-200,0 °C	-150,0 °C	800,0 °C	1050,0 °C	$\pm 0,6$ °C	$\pm 1,2$ °C
Spannung	-32512	-27648 -80 mV	27648 80 mV	32511	$\pm 0,05\%$	$\pm 0,1\%$

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 $\pm 1,5$ °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.

A.7.6 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- 50 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

Unterdrückungsfrequenz (Hz)	Integrationszeit (ms)	Aktualisierungszeit (Sekunden)		
		1-kanaliges SB	4-kanaliges SM	8-kanaliges SM
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.7.7 Auswahltabelle RTD-Gebertyp

Tabelle A- 51 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 0 °C bis 55 °C
Pt 0,003850 ITS90 DIN EN 60751	Pt 10	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±1,0 °C	±2,0 °C
	Pt 50	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 100						
	Pt 200						
	Pt 500						
	Pt 1000						
Pt 0,003902 Pt 0,003916 Pt 0,003920	Pt 100	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 200	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 500	-243,0 °C	-200,0 °C	850,0 °C	1000,0 °C	±0,5 °C	±1,0 °C
	Pt 1000						
Pt 0,003910	Pt 10	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±1,0 °C	±2,0 °C
	Pt 50	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	Pt 100	-273,2 °C	-240,0 °C	1100,0 °C	1295 °C	±0,8 °C	±1,6 °C
	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

Temperaturkoeffizient	RTD-Typ	Minimum unterer Bereich ¹	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ²	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich 0 °C bis 55 °C
	Cu 100						

¹ Die RTD-Werte unterhalb des Minimums für den unteren Bereich werden als -32768 ausgegeben.

² Die RTD-Werte oberhalb des Minimums für den oberen Bereich werden als -32768 ausgegeben.

Tabelle A- 52 Beständigkeit gegen:

Bereich	Minimum unterer Bereich	Unterer Grenzwert Nennbereich	Oberer Grenzwert Nennbereich	Maximum oberer Bereich ¹	Genauigkeit Normalbereich bei 25 °C	Genauigkeit Normalbereich 0 °C bis 55 °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 RTD-Werte oberhalb des Minimums für den oberen Bereich werden als -32768 ausgegeben.

A.7.8 Filterauswahl und Aktualisierungszeiten beim RTD

Tabelle A- 53 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.

A.8 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/support-request>).

A.8.1 PROFIBUS-Master/Slave

Hinweis

Die S7-1200 PROFIBUS CMs und der GPRS CP haben keine Zulassung für Anwendungen für das Seewesen

Die folgenden Module haben keine Zulassung für das Seewesen:

- CM 1242-5 PROFIBUS-Slavemodul
 - CM 1243-5 PROFIBUS-Mastermodul
 - CP 1242-7 GPRS-Modul
-

Hinweis

Um diese Module nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

A.8.1.1 CM 1242-5 PROFIBUS-Slave

Tabelle A- 54 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 %

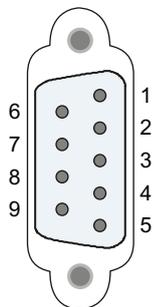
Technische Daten

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	
• Breite	• 30 mm
• Höhe	• 100 mm
• Tiefe	• 75 mm
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- 55 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.8.1.2 CM 1243-5 PROFIBUS-Master

Tabelle A- 56 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	

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 / extern

- Min. Leitungsquerschnitt • min.: 0,14 mm² (AWG 25)
- Max. Leitungsquerschnitt • max.: 1,5 mm² (AWG 15)
- Anzugsmoment der Schraubklemmen • 0,45 Nm (4 lb.in.)

Maße und Gewichte

- Breite • 30 mm
- Höhe • 100 mm
- Tiefe • 75 mm

Gewicht

- Nettogewicht • 134 g
- Gewicht inklusive Verpackung • 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

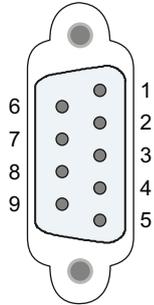


Tabelle A- 57 Kontaktbelegung der Sub-D-Buchse

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

ACHTUNG

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.8.2

GPRS CP

Hinweis

Die S7-1200 PROFIBUS CMs und der GPRS CP haben keine Zulassung für Anwendungen für das Seewesen

Die folgenden Module haben keine Zulassung für das Seewesen:

- CM 1242-5 PROFIBUS-Slavemodul
- CM 1243-5 PROFIBUS-Mastermodul
- CP 1242-7 GPRS-Modul

Hinweis

Um diese Module nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

A.8.2.1 Technische Daten des CP 1242-7

Tabelle A- 58 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 ±2dBm • GSM 900, Class 4: +33 dBm ±2dBm • GSM 1800, Class 1: +30 dBm ±2dBm • GSM 1900, Class 1: +30 dBm ±2dBm
GPRS	Multislot-Klasse 10 Endgeräteklasse B Kodierungsschema 1...4 (GMSK)
SMS	Betriebsmodus abgehend: MO Dienst: Punkt zu Punkt
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
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

Technische Daten

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.) |

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 • Transport-/Lagertemperatur • Relative Feuchte
	<ul style="list-style-type: none"> • -40 °C bis +70 °C • -40 °C bis +70 °C • 100 %
Außenmaterial	Hart-PVC, UV-beständig
Konstruktiver Aufbau	Antenne mit 5 m fest verbundenem Kabel und SMA-Stecker
Maße (D x H) in mm	25 x 193

ANT794-4MR

Gewicht

- | | |
|-----------------------|---------|
| • 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.8.3 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

A.8.4 RS485-, RS232- und RS422-Kommunikation

A.8.4.1 Technische Daten des CM 1241 RS485

Hinweis

Um dieses CB nutzen zu können, benötigen Sie eine CPU mit Firmware ab V2.0.

Tabelle A- 59 Allgemeine technische Daten

Technische Daten	CB 1241 RS485
Bestellnummer	6ES7 241-1CH30-1XB0
Abmessungen	38 x 62 x 21
Gewicht	40 Gramm

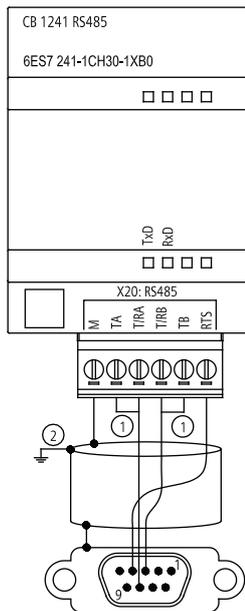
Tabelle A- 60 Sender und Empfänger

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- 61 Spannungsversorgung

Technische Daten	CB 1241 RS485
Verlustleistung	1,5 W
Max. Stromaufnahme (SM-Bus)	50 mA
Max. Stromaufnahme (24 V DC)	80 mA

CB 1241 RS485 (6ES7 241-1CH30-1XB0)



- ① "TA" und "TB" wie gezeigt anschließen, um das Netzwerk abzuschließen. (Nur die Endgeräte im RS485-Netz abschließen.)
- ② Verwenden Sie geschirmte, verdrehte Leiterpaare und schließen Sie den Kabelschirm an Erde an.

Sie schließen nur die zwei Enden des RS485-Netzes ab. Die Geräte zwischen den beiden Endgeräten werden nicht abgeschlossen. Siehe Abschnitt "Abschließen eines RS485-Busanschlussteckers".

A.8.4.2 CM 1241 RS485 und RS232

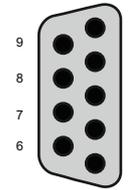
Tabelle A- 62 Allgemeine technische Daten

Technische Daten	CM 1241 RS485	CM 1241 RS232
Abmessungen (B x H x T)	30 x 100 x 75 (mm)	30 x 100 x 75 mm
Gewicht	150 Gramm	150 Gramm
Verlustleistung	1,1 W	1,1 W
aus +5 V DC	220 mA	220 mA

Tabelle A- 63 Sender und Empfänger

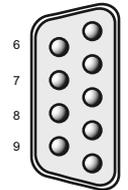
Technische Daten		Beschreibung
Typ		<ul style="list-style-type: none"> • RS485 (2-Leiter-Halbduplex) • RS232 (Vollduplex)
Sender (RS485)	Gleichtaktspannungsbereich	-7 V bis +12 V, 1 s, 3 V fortlaufender Effektivwert
	Differentialausgangsspannung Sender	min. 2 V bei $R_L = 100 \Omega$ min. 1,5 V bei $R_L = 54 \Omega$
	Abschluss und Bias	10 k Ω zu +5 V an B, PROFIBUS Pin 3 10 k Ω zu GND an A, PROFIBUS Pin 8
Sender (RS232)	Ausgangsspannung Sender	min. +/- 5 V bei $R_L = 3 \text{ k}\Omega$
	Ausgangsspannung Sender	max. +/- 15 V DC
Empfänger	Eingangsimpedanz Empfänger	<ul style="list-style-type: none"> • RS485: min. 5,4 kΩ einschließlich Abschluss • RS232: min. 3 kΩ
	Ansprechgrenze/Sensibilität Empfänger	<ul style="list-style-type: none"> • RS485: min. +/- 0,2 V, 60 mV typ. Hysterese • RS232: min. 0,8 V Low, max. 2,4 V High, typ. Hysterese 0,5 V
	Empfänger-Eingangsspannung (nur RS232)	max. +/- 30 V DC
Elektrische Trennung	Signal zu Masse	500 V AC für 1 Minute
	Signal zu CPU-Logik	
Kabellänge, geschirmt (max.)		<ul style="list-style-type: none"> • RS485: 1000 m • RS232: 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
Flussskontrolle (RS232)		Nicht unterstützt
Flussskontrolle (RS485)		Hardware, Software
Wartezeit		0 bis 65535 ms

Tabelle A- 64 CM 1241 RS485 Anschluss und Verdrahtung

Pol	Beschreibung	Steckverbinde r (Buchse)	Pol	Beschreibung
1 GND	Logik- oder Kommunikationsmasse		6 PWR	+5 V mit 100 Ohm Reihenwiderstand: Ausgang
2	Nicht angeschlossen		7	Nicht angeschlossen
3 TxD+	Signal B (RxD/TxD+): Eingang/Ausgang		8 TXD-	Signal A (RxD/TxD-): Eingang/Ausgang
4 RTS	Sendeanforderung (TTL-Pegel): Ausgang		9	Nicht angeschlossen
5 GND	Logik- oder Kommunikationsmasse		SHELL	Erdungsanschluss

¹ RTS ist ein Signal auf TTL-Ebene und kann dazu genutzt werden, ein weiteres Halbduplex-Gerät basierend auf diesem Signal zu steuern. Es wird beim Senden aktiviert und ist ansonsten deaktiviert. Im Gegensatz zum CM 1241 RS232, kann dieses Signal beim CM 1241 RS485 nicht vom Anwender gesteuert werden. Sie können es weder manuell setzen noch seine Verlängerung bewirken.

Tabelle A- 65 CM 1241 RS232 Anschluss und Verdrahtung

Pol	Beschreibung	Steckverbinder (Stecker)	Pol	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.8.4.3 Technische Daten des CM 1241 RS422/485

Technische Daten des CM 1241 RS422/485

Tabelle A- 66 Allgemeine technische Daten

Technische Daten	CM 1241 RS422/485
Bestellnummer	6ES7 241-1CH31-0XB0
Abmessungen	30 x 100 x 75 mm
Gewicht	155 Gramm

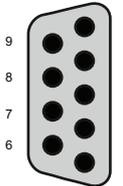
Tabelle A- 67 Sender und Empfänger

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 jedem Fall ist der externe Abschluss erforderlich (<i>siehe Kapitel 12, Kommunikationsprotokolle, Seite xxx</i>)
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- 68 Stromversorgung

Technische Daten	CM 1241 RS422/485
Verlustleistung	1,2 W
aus +5 V DC	240 mA

Tabelle A- 69 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 (RxD/TxD+): Eingang/Ausgang		8 TXD-	Signal A (RxD/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 Zugehörige Produkte

A.9.1 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

Weitere Informationen zu diesem Produkt und die Produktdokumentation finden Sie auf der Kundensupport-Website (<http://www.siemens.com/automation/support-request>).

A.9.2 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 Kundensupport-Website (<http://www.siemens.com/automation/support-request>).

Index

"

"Box"-Anweisung
Erste Schritte,

A

Ad-hoc-Modus
ISO on TCP, 136
TCP, 136

Adressierung
Boolesche Werte oder Bitwerte, 67
Datenbaustein, 66
Einzelne Eingänge (E) oder Ausgänge (A), 67
Globaler Speicher, 66
Prozessabbild, 66
Speicherbereiche, 66
Temporärer Speicher, 66

Aktive/passive Kommunikation
Parameter, 142
Partner konfigurieren, 140, 157
Verbindungs-IDs, 137

Alarmer
Alarmlatenz, 56
Organisationsbaustein (OB), 93
Übersicht, 55

Allgemeine technische Daten, 249

Analoge E/A
Darstellung Ausgang (Spannung), 278
Darstellung Ausgang (Strom), 278
Darstellung Eingang (Spannung), 275
Schrittantwortzeiten der Eingänge, 276
Umwandlung in physikalische Einheiten, 45

Analogwerte skalieren, 45

Ändern
Zustand im Programmiereditor, 232

Anfangswerte
Startwerte eines DBs erfassen und
zurücksetzen, 237

Anlauf-OB, 55

Anlaufparameter, 80

Anlegen einer HMI-Verbindung, 47

Anschlüsse
Anzahl der Verbindungen (PROFINET), 134
Ethernet-Protokolle, 156

HMI-Verbindung, 47
Kommunikationsarten, 133
Konfiguration, 142
Netzwerkverbindung, 47
Partner, 140
S7-Verbindung, 156
Typen, Verbindungen mit mehreren
Teilnehmern, 156
Verbindungs-IDs, 137

Anweisung Bereich kopieren (MOVE_BLK), 104
Anweisung Bereich ununterbrechbar kopieren
(UMOVE_BLK), 104

Anweisung CTRL_PWM, 115

Anweisung TRCV_C, 134

Anweisung TSEND_C, 134

Anweisung Verschieben, 104

Anweisungen

Analogwerte skalieren, 45

Beobachten, 231, 232

Bereich kopieren (MOVE_BLK), 104

Bereich ununterbrechbar kopieren

(UMOVE_BLK), 104

Bitverknüpfung, 102

CALCULATE, 44, 107

CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl
erzeugen), 106

CONV (Umwandeln), 105

CTRL_PWM, 115

DeviceStates, 123

Dezentrale Peripherie PROFIBUS, 146

Drag & Drop, 28

Drag & Drop zwischen Editoren, 32

Einfügen, 28

Ergänzen von Eingängen oder Ausgängen in KOP-
und FUP-Anweisungen, 29

Erste Schritte, 42, 43

Erweiterbare Anweisungen, 30

Favoriten, 29

FLOOR, 106

Forcefunktion, 234

Forcen, 234

GET, 156

GET_DIAG, 123

Hinzufügen eines Parameters, 43

HSC (schneller Zähler), 124, 125

LED-Zustand, 122

MC_ChangeDynamic, 225

MC_CommandTable, 222

MC_Halt, 211
 MC_Home, 207
 MC_MoveAbsolute, 213
 MC_MoveJog, 220
 MC_MoveRelative, 215
 MC_MoveVelocity, 217
 MC_Power, 203
 MC_Reset, 206
 ModuleStates, 123
 NORM_X (Normalisieren), 106
 PID_Compact, 175
 PUT, 156
 ROUND, 106
 SCALE_X (Skalieren), 106
 Spalten und Überschriften, 32
 TRCV_C, 134
 TRUNC (Ganzzahl erzeugen), 106
 TSEND_C, 134
 Vergleichen, 104
 Verschieben, 104
 Versionen von Anweisungen, 32
 WWW, 196
 Zähler, 113
 Zustand, 231, 232
 Anwenderprogramm
 Bausteine aus einer Online-CPU kopieren, 238
 Drag & Drop zwischen Editoren, 32
 Einfügen von Anweisungen, 28
 Ergänzen von Eingängen oder Ausgängen in KOP-
 und FUP-Anweisungen, 29
 Erweiterbare Anweisungen, 30
 Favoriten, 29
 Mit CPU oder Memory Card verknüpfen, 88
 Passwortschutz, 87
 Anzeige erster Zyklus, 83
 Arbeitsspeicher, 14, 62
 Arithmetik, 44, 107
 ASi
 ASi-Adresse,
 AS-i-Slave hinzufügen,
 CM 1243-2 AS-i-Mastermodul hinzufügen,
 Eigenschaften der ASi-Adresse,
 ASi-Adresse,
 Konfigurieren,
 AS-i-Master-Modul, 153
 Merkmale des CM 1243-2 AS-i-Moduls, 153
 ATEX-Zulassung, 250
 Auf Werkseinstellungen zurücksetzen, 242
 Aufrufstruktur, 121
 Aus CPU laden
 Erkennen, 243
 Ausgangsparameter, 94

B

Baustein
 Arten, 53
 Aufrufen eines anderen Codebausteins, 97
 Erste Schritte, 96
 Konsistenzprüfung, 121
 Bausteinaufruf
 Grundwissen, 53
 Bausteine
 Alarmer, 15, 56
 Anlauf-OBs, 56
 Anzahl der Codebausteine, 15
 Anzahl der OBs, 15, 56
 Bausteine aus einer Online-CPU kopieren, 238
 Datenbausteine (DBs), 53
 Ereignisse, 56
 Funktionen (FCs), 53
 Funktionsbausteine (FBs), 53
 Größe des Anwenderprogramms, 15
 Organisationsbausteine (OBs), 15, 53, 55, 56
 Passwortschutz, 87
 Schachtelungstiefe, 15
 Überwachung, 15
 Zähler (Anzahl und Speicherbedarf), 15
 Zeiten (Anzahl und Speicherbedarf), 15
 Bausteine aus einer Online-CPU kopieren, 238
 Bearbeitung von Alarmereignissen
 Organisationsbaustein (OB), 93
 Bedienoberfläche
 Portalansicht, 27
 Projektansicht, 27
 Bedienpanel, 20, 30, 53, 230
 Bemessungsspannungen, 253
 Benutzerdefinierte Webseiten, 189, 193
 Aktivieren mit WWW-Anweisung, 196
 Aktualisieren, 194
 Anlegen mit HTML-Editor, 193
 Einschränkungen Ladespeicher, 194
 Konfigurieren, 195
 Programmbausteine generieren, 195
 Programmieren in STEP 7, 196
 Benutzerdefinierte Webseiten aktualisieren, 194
 Benutzerdefinierte Webseiten anlegen, 193
 Beobachten
 Beobachtungstabelle, 231
 Forcefunktion, 234
 Forcetable, 234
 KOP-Zustand, 231, 232
 Startwerte eines DBs zurücksetzen, 237
 Werte eines DBs erfassen, 237
 Beobachtungstabelle
 Beobachten, 231

- Forcen, 119
- Beobachtungstabellen, 232
- Betriebszustand, 30, 53, 230
- Betriebszustand RUN, 54
 - Bedienpanel, 30, 53, 230
 - Forcefunktion, 234
 - Schaltflächen in der Funktionsleiste, 30
- Betriebszustand STOP
 - Bedienpanel, 30, 230
 - Forcefunktion, 234
 - Schaltflächen in der Funktionsleiste, 30
- Bewegungssteuerung
 - Achse konfigurieren, 200
 - MC_ChangeDynamic, 225
 - MC_CommandTable, 222
 - MC_Halt, 211
 - MC_Home, 207
 - MC_MoveAbsolute, 213
 - MC_MoveJog, 220
 - MC_MoveRelative, 215
 - MC_MoveVelocity, 217
 - MC_Power, 203
 - MC_Reset, 206
 - Referenzpunktfahrt der Achse, 209
 - Übersicht, 197
- Bibliothek für das USS-Protokoll, 167
- Bitverknüpfung, 102
- Boolesche Werte oder Bitwerte, 67
- Busstecker, 19

- C**
- CALCULATE, 44, 107
 - Analogwerte skalieren, 45
- CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 106
- CE-Zulassung, 249
- CM 1241
 - Technische Daten RS422/RS485, 298
- CM 1241 RS232, Technische Daten, 298
- CM 1241 RS485, Technische Daten, 298
- CM 1243-2 AS-i
 - Merkmale des AS-i-Moduls, 153
- Codebaustein
 - Alarmer, 15
 - Anzahl der Codebausteine, 15
 - Anzahl der OBs, 15
 - Baustein aufrufen, 97
 - Beobachten, 15
 - DB (Datenbaustein), 96
 - FB (Funktionsbaustein), 94
 - FC (Funktion), 94
 - Größe des Anwenderprogramms, 15
 - Knowhow-Schutz, 87
 - Kopierschutz, 88
 - Mit CPU oder Memory Card verknüpfen, 88
 - Organisationsbausteine (OBs), 15
 - Schachtelungstiefe, 15
 - Zähler (Anzahl und Speicherbedarf), 15
 - Zeiten (Anzahl und Speicherbedarf), 15
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33
- Codebausteine, 91
- Codebausteine vergleichen, 239
- CONV (Umwandeln), 105
- Cookie-Einschränkungen, Standard-Webseiten, 192
- CPU
 - Anlaufparameter, 80
 - Anlaufverarbeitung, 80
 - Anzahl der Kommunikationsverbindungen, 134
 - ASi,
 - ASi-Adresse,
 - AS-i-Port, 154
 - Auf Werkseinstellungen zurücksetzen, 242
 - Baustein aufrufen, 97
 - Bausteine aus einer Online-CPU kopieren, 238
 - Bausteine vergleichen und synchronisieren, 239
 - Bedienpanel, 30, 53, 230
 - Beobachten, 231
 - Beobachtungstabellen, 232
 - Betriebszustände, 52
 - Darstellung Analogeingang (Spannung), 275
 - Diagnosepuffer, 240
 - Erste Schritte, 37
 - Ethernet-Port, 85
 - Forcen, 234
 - Gerätekonfiguration, 75
 - HMI-Geräte, 20
 - HSC-Konfiguration, 129
 - IP-Adresse, 84
 - Knowhow-Schutz, 87
 - Kommunikation mit HMI konfigurieren, 131
 - Kommunikationsarten, 133
 - Kommunikationsboard (CB), 19
 - Module hinzufügen, 78
 - Netzwerkverbindung, 132
 - Neues Gerät hinzufügen, 77
 - Nicht spezifizierte CPU, 76, 243
 - Online, 231, 241
 - Online gehen, 229
 - Organisationsbaustein (OB), 93
 - Parameter konfigurieren, 80, 84
 - Passwortschutz, 86
 - PROFIBUS, 151

PROFIBUS-Adresse, 151
 PROFIBUS-Port, 151
 PROFINET, 85
 Programmausführung, 51
 Schaltflächen RUN/STOP, 30
 Schrittantwortzeiten der analogen Eingänge, 276
 Schutzstufen, 86
 Signalboard (SB), 19
 Startwerte eines DBs zurücksetzen, 237
 Technische Daten der CPU 1211C, 255
 Technische Daten der CPU 1212C, 255
 Technische Daten der CPU 1214C, 255
 Thermischer Bereich, 21
 Übersicht, 13
 Vergleichstabelle, 14
 Werte eines DBs erfassen, 237
 Zugriffsschutz, 86
 Zustand eines Codebausteins erfassen, 33
 Zustand eines Codebausteins wiederherstellen, 33
 CPU-Eigenschaften, benutzerdefinierte
 Webseiten, 195
 C-Tick-Zulassung, 251
 cULus-Zulassung, 250

D

Datenbaustein
 Globaler Datenbaustein, 66, 96
 Instanz-Datenbaustein, 66
 Startwerte zurücksetzen, 237
 Werte erfassen, 237
 Datenbaustein (DB), 96
 Datenhantierungsbausteine (DHBs), 96
 Datenprotokoll
 Überblick über Datenprotokolle, 116
 Datentyp DTL, 64
 Datentyp DTL (Date and Time Long), 64
 Datentypen, 64
 DTL, 64
 DB (Datenbaustein), 96
 Startwerte zurücksetzen, 237
 Werte erfassen, 237
 DB-Fragmente (benutzerdefinierte Webseiten)
 Generieren, 195
 DBs für benutzerdefinierte Webseiten erstellen, 195
 DBs für benutzerdefinierte Webseiten generieren, 195
 DeviceStates, 123
 Diagnose
 Anweisung LED, 122
 DeviceStates, 123
 GET_DIAG, 123
 ModuleStates, 123

Statusanzeige, 83
 Diagnosefehler-OB, 56
 Diagnosepuffer, 240
 Digitales Signalmodul (SM)
 SM 1221, 262
 SM 1222, 263
 SM 1223, 264, 265
 Technische Daten der Eingänge und
 Ausgänge, 266
 Dokumentation, 4
 Drag & Drop zwischen Editoren, 32

E

E/A
 Adressierung, 68
 Darstellung Analogausgang (Spannung), 278
 Darstellung Analogausgang (Strom), 278
 Darstellung Analogeingang (Spannung), 275
 Forcefunktion, 234
 Forcen, 234
 Schrittantwortzeiten der analogen Eingänge, 276
 Zustand in KOP beobachten, 232
 E/A-Module
 Beobachtungstabellen, 232
 Editoren teilen
 Erste Schritte, 38, 42
 Ein- und Ausgänge
 Beobachten, 231
 Einbau
 Montageabmessungen, 21
 Signalmodul (SM), 19
 Thermischer Bereich, 21
 Einfügen eines Geräts
 Nicht spezifizierte CPU, 76, 243
 Einfügen von Anweisungen
 Drag & Drop, 28
 Drag & Drop zwischen Editoren, 32
 Favoriten, 29
 Einschränkungen
 Benutzerdefinierte Webseiten, 194
 Webserver, 190
 Einstellungen, 31
 Einstellungen von STEP 7 ändern, 31
 Elektromagnetische Verträglichkeit (EMV), 252
 Entwerfen einer Automatisierungslösung mit einem
 PLC-Gerät, 53, 91
 Ereignisausführung, 56
 Ereignisse, 240
 Organisationsbaustein (OB), 93
 Ergänzen von Eingängen oder Ausgängen in KOP-
 und FUP-Anweisungen, 29

Erkennen, 243
 Erkennung zum Laden einer Online-CPU, 76
 Ersetzen von Modulen, 36
 Erste Schritte
 "Box"-Anweisung,
 Adressierung, 42
 Anweisungen, 42
 Arithmetische Anweisung, 43
 Codebaustein, 96
 CPU, 37
 Editoren teilen, 38, 42
 HMI, 46, 48
 HMI-Verbindung, 47
 Kontakte, 40
 KOP-Programm, 40, 43
 Netzwerk, 40
 Netzwerkverbindung, 47
 neue SPS, 37
 PLC-Variablen, 38, 42
 Programmbaustein, 96
 Projekt, 37
 Variablen, 38, 42
 Erweiterbare Anweisungen, 30
 Erweiterung der Leistungsmerkmale der S7-1200, 16
 Ethernet
 Ad-hoc-Modus, 136
 Anzahl der Kommunikationsverbindungen, 134
 GET, 156
 IP-Adresse, 84
 Kommunikationsarten, 133
 Netzwerkverbindung, 132
 PUT, 156
 Übersicht, 135
 Verbindungs-IDs, 137
 Ethernet-Anweisungen
 TRCV_C, 134
 TSEND_C, 134
 Ethernet-Kommunikation, 131
 Ethernet-Protokolle, 135
 Verbindungen mit mehreren Teilnehmern, 156

F

FAQs, 4
 Favoriten, Funktionsleiste, 29
 FB (Funktionsbaustein), 94
 FC (Funktion), 94
 Fehler
 Diagnosefehler, 61
 Zeitfehler, 59
 FLOOR, 106
 FM-Zertifizierung, 250

Forcen, 234
 Eingänge der Peripherie, 234
 Eingänge und Ausgänge, 234
 Speicherbereich E, 234
 Zyklus, 234
 Forcetable
 Eingänge der Peripherie ansprechen, 234
 Forcefunktion, 234
 Forcen, 234
 Frequenz, Taktbits, 83
 Funktion (FC), 94
 Knowhow-Schutz, 87
 Zustand eines Codebausteins erfassen, 33
 Zustand eines Codebausteins wiederherstellen, 33
 Funktionsbaustein (FB)
 Anfangswert, 94
 Ausgangsparameter, 94
 Instanz-Datenbaustein, 94
 Knowhow-Schutz, 87
 Zustand eines Codebausteins erfassen, 33
 Zustand eines Codebausteins wiederherstellen, 33
 FUP (Funktionsplan), 98

G

Gerätekonfiguration, 75
 ASi,
 AS-i-Port, 154
 Erkennen, 76, 243
 Ethernet-Port, 85
 Konfigurieren der CPU, 80, 84
 Konfigurieren der Module, 80, 84
 Module hinzufügen, 78
 Netzwerkverbindung, 132
 Neues Gerät hinzufügen, 77
 Nicht gesteckte Module, 36
 PROFIBUS, 151
 PROFIBUS-Port, 151
 PROFINET, 85
 GET, 156
 Verbindung konfigurieren, 141
 GET_DIAG, 123
 Globale Bibliothek
 USS, 167
 Globaler Datenbaustein, 66, 96
 Globaler Speicher, 66
 Grundlegende Panels (HMI), 20

H

Handbücher, 4

Hardwarekonfiguration, 75

- ASi,
- AS-i-Port, 154
- Erkennen, 76, 243
- Ethernet-Port, 85
- Konfigurieren der CPU, 80, 84
- Konfigurieren der Module, 80, 84
- Module hinzufügen, 78
- Netzwerkverbindung, 132
- Neues Gerät hinzufügen, 77
- PROFIBUS, 151
- PROFIBUS-Port, 151
- PROFINET, 85

Hinzufügen eines Geräts

- Nicht spezifizierte CPU, 243

HMI

- Bild, 48
- Erste Schritte, 46, 48
- HMI-Verbindung, 47
- Netzwerkverbindung, 47
- PROFINET-Kommunikation konfigurieren, 131

HMI-Geräte

- Netzwerkverbindung, 132
- Übersicht, 20

HMI-Verbindung, 47

Hotline, 5

HSC (schneller Zähler)

- Funktionsweise, 124, 125
- Konfiguration, 129

HTML-Seiten, benutzerdefiniert, 193

- Aktualisieren, 194
- Entwickeln, 193
- Seitenanordnung, 195

I

Impulsfolge (PTO), 115

Informationsquellen, 4

Instanz-Datenbaustein, 66

IP-Adresse, 84, 85

- CPU online konfigurieren, 241

IP-Adresse des Routers, 85

IP-Router, 84

ISO on TCP

- Ad-hoc-Modus, 136

ISO-on-TCP

- Parameter, 142
- Verbindungs-IDs, 137
- Verbindungskonfiguration, 140

ISO-on-TCP-Protokoll, 135

J

JavaScript-Einschränkungen, Standard-Webseiten, 191

K

Knowhow-Schutz

- Passwortschutz, 87

Kommunikation

- aktiv/passiv, 157
- Aktiv/passiv, 140, 142
- Anzahl der Verbindungen (PROFINET), 134
- ASi-Adresse,
- IP-Adresse, 84
- Konfiguration, 140, 142, 157
- Modbus, 165
- Netzwerk, 131
- Netzwerkverbindung, 132
- Parameter, 142
- PROFIBUS-Adresse, 151
- PROFINET und PROFIBUS, 133
- PtP, 165
- TCON_Param, 142
- USS, 165
- Verbindungs-IDs, 137

Kommunikationsboard (CB)

- CB 1241 RS485, 296
- LED-Anzeigen, 166
- Module hinzufügen, 78
- RS485, 165
- Übersicht, 19
- Vergleichstabelle, 16

Kommunikationsmodul

- CM 1241 RS422/RS485, Technische Daten, 298
- Gerätekonfiguration, 75
- Neues Gerät hinzufügen, 77

Kommunikationsmodul (CM)

- CM 1243-2 AS-i-Mastermodul hinzufügen, 153
- CM 1243-5 (DP-Master) hinzufügen, 150
- LED-Anzeigen, 166
- Module hinzufügen, 78
- RS232 und RS485, 165
- Technische Daten, 298
- Übersicht, 19
- Vergleichstabelle, 16

Kommunikationsmodul (CM), USS-Bibliothek, 167

Kommunikationsprozessor (CP)

- Module hinzufügen, 78
- Übersicht, 19
- Vergleichstabelle, 16

Kommunikationsschnittstellen

- CB 1241 RS485, 296
- Module hinzufügen, 78
- RS232 und RS485, 165
- Vergleichstabelle der Module, 16
- Konfiguration
 - Anlaufparameter, 80
 - ASi,
 - ASi-Adresse,
 - AS-i-Port, 154
 - Benutzerdefinierte Webseiten, 195
 - Erkennen, 76, 243
 - HSC (schneller Zähler), 129
 - Industrial Ethernet-Port, 85
 - IP-Adresse, 84
 - Module hinzufügen, 78
 - Netzwerkverbindung, 132
 - PROFIBUS, 151
 - PROFIBUS-Adresse, 151
 - PROFIBUS-Port, 151
 - PROFINET, 85
- Konfiguration Sendeparameter, 140, 157
- Konfigurieren von Parametern
 - Module, 80
- Konsistenzprüfung, 121
- Kontaktdaten, 5
- Kontakte
 - Erste Schritte, 40
- KOP (Kontaktplan)
 - Beobachten, 231, 232
 - Programmiereditor, 232
 - Übersicht, 97
 - Zustand, 231, 232, 234
- Kopierschutz
 - Mit CPU oder Memory Card verknüpfen, 88
- Kunden-Support, 5

L

- Laden aus CPU
 - Anwenderprogramm, 238
 - Bausteine aus einer Online-CPU kopieren, 238
- Ladespeicher, 14, 62
- Ladespeicher, benutzerdefinierte Webseiten, 194
- Latenz, 56
- Lebensdauer eines Relais, 255
- LED (LED-Status lesen), 122
- LED-Anzeigen
 - Anweisung LED, 122
 - Kommunikationsschnittstelle, 166
- Lineare Programmierung, 91

M

- MAC-Adresse, 84
- MC_ChangeDynamic, 225
- MC_CommandTable, 222
- MC_Halt, 211
- MC_Home, 207
- MC_MoveAbsolute, 213
- MC_MoveJog, 220
- MC_MoveRelative, 215
- MC_MoveVelocity, 217
- MC_Power, 203
- MC_Reset, 206
- Memory Card
 - Ladespeicher, 62
- Mit CPU oder Memory Card verknüpfen, 88
- Modbus, 165
- MODBUS
 - Versionen, 32
- Module
 - CB 1241 RS485, 296
 - Kommunikationsboard (CB), 19
 - Kommunikationsmodul (CM), 19
 - Kommunikationsprozessor (CP), 19
 - Parameter konfigurieren, 80, 84
 - SB 1231 AI 1x12 Bit, 271
 - SB 1232 AO 1x12 Bit, 271
 - Signalboard (SB), 19
 - Signalmodul (SM), 19
 - SM 1221, 262
 - SM 1222, 263
 - SM 1223, 264, 265
 - SM 1231 AI 8 x 13 Bit, 272
 - SM 1232 AO 2 x 14 Bit, 272
 - SM 1232 AO 4 x 14 Bit, 272
 - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 272
 - Thermischer Bereich, 21
 - Vergleichstabelle, 16
- ModuleStates, 123
- Montage
 - Abmessungen, 21
 - Thermischer Bereich, 21
- MRES
 - Bedienpanel, 30, 53, 230
- My Documentation Manager, 4

N

- Netzwerk
 - Erste Schritte, 40, 43
 - Netzwerkverbindung, 47
- Netzwerkcommunication, 131

- Netzwerkverbindung
 - Konfiguration, 132
- Netzwerkverbindung erstellen, 132
- Netzwerkverbindung erstellen, 132
- Neues Gerät hinzufügen
 - Nicht spezifizierte CPU, 76
 - Vorhandene Hardware erkennen, 76
- Neues Projekt
 - Erste Schritte, 37
 - Hinzufügen eines HMI-Geräts, 46
 - HMI-Bild, 48
 - HMI-Verbindung, 47
 - Netzwerkverbindung, 47
- Nicht gesteckte Module, 36
- Nicht spezifizierte CPU, 76, 243
- NORM_X (Normalisieren), 106

O

- Online
 - Bedienpanel, 30, 53, 230
 - Beobachten, 231
 - Beobachtungstabelle, 231, 232
 - Erkennen, 243
 - Forcefunktion, 234
 - Forcen, 234
 - IP-Adresse, 241
 - Online gehen, 229
 - Schaltflächen RUN/STOP, 30
 - Startwerte eines DBs zurücksetzen, 237
 - Tageszeit, 241
 - Überwachung der Speicherauslastung, 230
 - Vergleichen und synchronisieren, 239
 - Werte eines DBs erfassen, 237
 - Zustand, 231, 232
 - Zykluszeitüberwachung, 230
- Online- und Diagnose-Tools
 - Laden im Betriebszustand RUN, 245
- Online- und Offline-CPU's vergleichen und synchronisieren, 239
- OPC, 160
- Organisationsbaustein
 - Anlegen, 94
 - Aufrufen, 55
 - Bearbeiten, 93
 - Funktion, 55
 - Funktionsweise konfigurieren, 94
 - Knowhow-Schutz, 87
 - mehrere Programmzyklus-OBs, 94
 - Prioritätsklassen, 55
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33

P

- Panels (HMI), 20
- Parameter konfigurieren
 - CPU, 80, 84
 - Ethernet-Port, 85
 - Module, 84
 - PROFINET, 85
- Parametrieren, 94
- Passive/aktive Kommunikation
 - Parameter, 142
 - Partner konfigurieren, 140, 157
 - Verbindungs-IDs, 137
- Passwortschutz
 - Codebaustein, 87
 - CPU, 86
 - Kopierschutz, 88
 - Mit CPU oder Memory Card verknüpfen, 88
 - Zugriff auf die CPU, 86
- PID
 - Algorithmus PID_3Step, 172, 178
 - Algorithmus PID_Compact, 172, 175
 - PID_3STEP, 178
 - PID_Compact, 175
 - Übersicht, 171
- PLC
 - Anweisungen, 42
 - Baustein aufrufen, 97
 - Bausteine aus einer Online-CPU kopieren, 238
 - Bausteine verwenden, 53, 91
 - Beobachten, 231
 - Erste Schritte, 37
 - Forcefunktion, 234
 - Forcen, 234
 - HSC-Konfiguration, 129
 - Knowhow-Schutz, 87
 - Module hinzufügen, 78
 - Übersicht über die CPU, 13
 - Variablen, 38, 42
 - Vergleichen und synchronisieren, 239
- PLC-Variablen
 - Erste Schritte, 38, 42
- Podcasts, 4
- Portalansicht, 27
 - Ethernet-Port konfigurieren, 85
 - Konfigurieren der CPU, 80, 84
 - Konfigurieren der Module, 80, 84
 - Neues Gerät hinzufügen, 77
 - PROFINET, 85
- Portnummer, 135
- Priorität
 - Priorität bei der Verarbeitung, 56
 - Prioritätsklassen, 55

- Prioritätsklasse, 55
- PROFIBUS
 - Anweisungen für die dezentrale Peripherie, 146
 - CM 1242-5 (DP-Slave), 145
 - CM 1243-5 (DP-Master), 145
 - CM 1243-5 (DP-Master) hinzufügen, 150
 - DP-Slave hinzufügen, 150
 - Eigenschaften PROFIBUS-Adresse, 151
 - GET, 156
 - Master, 144
 - Netzwerkverbindung, 132
 - PROFIBUS-Adresse, 151
 - PUT, 156
 - S7-Verbindung, 156
 - Slave, 144
- PROFIBUS-Adresse, 151
 - Konfigurieren, 151
- PROFINET, 131
 - Ad-hoc-Modus, 136
 - Anzahl der Kommunikationsverbindungen, 134
 - GET, 156
 - IP-Adresse, 84
 - Kommunikationsarten, 133
 - Netzwerkverbindung, 132
 - PUT, 156
 - S7-Verbindung, 156
 - Testen eines Netzwerks, 86
 - Übersicht, 135
 - Verbindungs-IDs, 137
- PROFINET RT, 135
- PROFINET-Schnittstelle
 - Eigenschaften der Ethernet-Adresse, 85
- Programm
 - Arithmetische Anweisung, 43
 - Baustein aufrufen, 97
 - Bausteine aus einer Online-CPU kopieren, 238
 - Beispielnetzwerk, 40, 43
 - Erste Schritte, 40, 43
 - Mit CPU oder Memory Card verknüpfen, 88
 - Passwortschutz, 87
 - Prioritätsklassen, 55
 - Startwerte eines DBs zurücksetzen, 237
 - Werte eines DBs erfassen, 237
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33
- Programm beobachten, 119
- Programm testen, 119
- Programmausführung, 51, 53
- Programmbaustein
 - Erste Schritte, 37, 96
- Programmiereditor
 - Beobachten, 232
 - Startwerte eines DBs zurücksetzen, 237
 - Werte eines DBs erfassen, 237
 - Zustand, 232
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33
- Programmierung
 - Algorithmus PID_3Step, 172, 178
 - Algorithmus PID_Compact, 172, 175
 - Codebausteine vergleichen und synchronisieren, 239
 - Drag & Drop zwischen Editoren, 32
 - Einfügen von Anweisungen, 28
 - Ergänzen von Eingängen oder Ausgängen in KOP- und FUP-Anweisungen, 29
 - Erste Schritte, 42
 - Erweiterbare Anweisungen, 30
 - Favoriten, 29
 - FUP (Funktionsplan), 98
 - KOP (Kontaktplan), 97
 - Linear, 91
 - Mit CPU oder Memory Card verknüpfen, 88
 - Nicht gesteckte Module, 36
 - Nicht spezifizierte CPU, 76, 243
 - PID_3STEP, 178
 - PID_Compact, 175
 - Prioritätsklassen, 55
 - SCL (Structured Control Language), 99, 100
 - Strukturiert, 91
 - Übersicht über den PID-Regler, 171
- Programminformationen
 - In der Aufrufstruktur, 121
- Programmkarte, 62
- Programmstruktur, 91
- Projekt
 - Codebaustein schützen, 87
 - Erste Schritte, 37
 - Hinzufügen eines HMI-Geräts, 46
 - HMI-Bild, 48
 - HMI-Verbindung, 47
 - Mit CPU oder Memory Card verknüpfen, 88
 - Netzwerkverbindung, 47
 - Programm, 42
 - Variablen, 38, 42
 - Vergleichen und synchronisieren, 239
 - Zugang zur CPU einschränken, 86
 - Zugriffsschutz, 86
- Projektansicht, 27
 - Ethernet-Port konfigurieren, 85
 - Gerätekonfiguration, 75
 - Konfigurieren der CPU-Parameter, 80, 84
 - Konfigurieren der Module, 80, 84
 - Neues Gerät hinzufügen, 77

- PROFINET, 85
- Protokoll
 - ISO on TCP, 135
 - PROFINET RT, 135
 - TCP, 135
 - UDP, 135
- Prozessabbild
 - Beobachten, 231, 232
 - Forcefunktion, 234
 - Forcen, 234
 - Zustand, 231, 232, 234
- Prozessalarm-OB, 55
- PTO (Impulsfolge), 115
 - Kann nicht geforct werden, 235
- PtP-Kommunikation, 165
- Punkt-zu-Punkt-Kommunikation, 165
- PUT, 156
 - Verbindung konfigurieren, 141
- PWM
 - Anweisung CTRL_PWM, 115
- PWM (Impulsdauermodulation)
 - Kann nicht geforct werden, 235

- Q**
- Querverweise, 120
 - Einleitung, 120
 - Verwendung, 120

- R**
- Remanenter Speicher, 14, 62
- ROUND, 106
- RS232- und RS485-Kommunikationsmodule, 165
- RUN, Betriebszustand, 52
 - Programmausführung, 51

- S**
- S7-1200
 - Anlaufparameter, 80
 - ASi,
 - ASi-Adresse,
 - AS-i-Port, 154
 - Baustein aufrufen, 97
 - Bedienpanel, 30, 53, 230
 - Beobachten, 231
 - Codebausteine vergleichen, 239
 - Diagnosepuffer, 240
 - Ethernet-Port, 85
 - Forcefunktion, 234
 - Forcen, 234
 - Gerätekonfiguration, 75
 - HMI-Geräte, 20
 - HSC-Konfiguration, 129
 - IP-Adresse, 84
 - Knowhow-Schutz, 87
 - Kommunikationsboard (CB), 19
 - Kommunikationsmodul (CM), 19
 - Kommunikationsprozessor (CP), 19
 - Konfigurieren der CPU-Parameter, 80, 84
 - Konfigurieren der Module, 80, 84
 - Module, 16
 - Module hinzufügen, 78
 - Montageabmessungen, 21
 - Netzwerkverbindung, 132
 - Neues Gerät hinzufügen, 77
 - Organisationsbaustein (OB), 93
 - Passwortschutz, 86
 - PROFIBUS, 151
 - PROFIBUS-Adresse, 151
 - PROFIBUS-Port, 151
 - PROFINET, 85
 - Programmausführung, 51
 - Schaltflächen RUN/STOP, 30
 - Signalboard (SB), 19
 - Signalmodul (SM), 19
 - Startwerte eines DBs zurücksetzen, 237
 - Thermischer Bereich, 21
 - TS-Adapter, 16
 - Übersicht über die CPU, 13
 - Vergleichstabelle der CPU-Varianten, 14
 - Werte eines DBs erfassen, 237
 - Zugriffsschutz, 86
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33
- S7-Kommunikation
 - Verbindung konfigurieren, 141
- SCALE_X (Skalieren), 106
- Schaltflächen RUN/STOP, 30
- Schneller Zähler
 - Funktionsweise, 125
 - HSC, 124
 - Konfiguration, 129
- Schneller Zähler (HSC)
 - Kann nicht geforct werden, 235
- Schutz
 - CPU, 86
 - Knowhow-Schutz für einen Codebaustein, 87
- Schutzart, 253
- Schutzstufe
 - Codebaustein, 87
 - CPU, 86

- Mit CPU oder Memory Card verknüpfen, 88
- SCL (Structured Control Language)
 - Algorithmus PID_3Step, 172, 178
 - Algorithmus PID_Compact, 172, 175
 - CEIL (Aus Gleitpunktzahl nächsthöhere Ganzzahl erzeugen), 106
 - CONV (Umwandeln), 105
 - DeviceStates, 123
 - FLOOR, 106
 - Ganzzahl erzeugen, 106
 - GET_DIAG, 123
 - LED-Zustand, 122
 - MC_ChangeDynamic, 225
 - MC_CommandTable, 222
 - MC_Halt, 211
 - MC_Home, 207
 - MC_MoveAbsolute, 213
 - MC_MoveJog, 220
 - MC_MoveRelative, 215
 - MC_MoveVelocity, 217
 - MC_Power, 203
 - MC_Reset, 206
 - ModuleStates, 123
 - NORM_X (Normalisieren), 106
 - PID_3STEP, 178
 - PID_Compact, 175
 - Programmiereditor, 100
 - SCALE_X (Skalieren), 106
 - Übersicht, 99
 - Übersicht über den PID-Regler, 171
 - Var-Abschnitt, 100
 - Zahl runden, 106
- Serielle Kommunikation, 165
- Service und Support, 5
- Sicherheit
 - Kopierschutz, 88
 - Mit CPU oder Memory Card verknüpfen, 88
 - Zugriffsschutz, 86
- Signalboard (SB)
 - Darstellung Analogausgang (Spannung), 278
 - Darstellung Analogausgang (Strom), 278
 - Darstellung Eingang (Spannung), 275
 - Gerätekonfiguration, 75
 - Module hinzufügen, 78
 - SB 1231 AI 1x12 Bit, 271
 - SB 1232 AO 1x12 Bit, 271
 - Schrittantwortzeiten der analogen Eingänge, 276
 - Übersicht, 19
- Signalboard (SM)
 - Neues Gerät hinzufügen, 77
- Signalmodul (SM)
 - Darstellung Analogausgang (Spannung), 278
 - Darstellung Analogausgang (Strom), 278
 - Darstellung Analogeingang (Spannung), 275
 - Gerätekonfiguration, 75
 - Module hinzufügen, 78
 - Neues Gerät hinzufügen, 77
 - Schrittantwortzeiten der analogen Eingänge, 276
 - SM 1221, 262
 - SM 1222, 263
 - SM 1223, 264, 265
 - SM 1231 AI 4 x 13 Bit, 272
 - SM 1232 AO 2 x 14 Bit, 272
 - SM 1232 AO 4 x 14 Bit, 272
 - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 272
 - Übersicht, 19
- SM und SB
 - Vergleichstabelle, 16
- SMS, 159
- Spalten und Überschriften in Taskcards, 32
- Speicher
 - Adressen der Peripherieeingänge (Forcetabelle), 234
 - Arbeitsspeicher, 62
 - Ladespeicher, 62
 - Remanenter Speicher, 62
 - Systemspeicher, 82
 - Taktmerker, 82
 - Temporärer Speicher (L), 66
- Speicherbereich E
 - Adressen der Peripherieeingänge (Forcetabelle), 234
 - Beobachten, 231
 - Beobachtungstabelle, 231
 - Forcefunktion, 234
 - Forcen, 234
 - Forcetabelle, 234
 - HSC (schneller Zähler), 125
 - KOP beobachten, 232
- Speicherbereiche
 - Adressierung von Booleschen Werten oder Bitwerten, 67
 - Datenbaustein, 66
 - Direkter Zugriff, 66
 - Globaler Speicher, 66
 - Prozessabbild, 66
 - Temporärer Speicher, 66
- Standard-Webseiten, 189
 - Cookie-Einschränkungen, 192
 - JavaScript-Einschränkungen, 191
- STARTUP, Betriebszustand
 - Forcefunktion, 234
 - Programmausführung, 51
- Startwerte eines DBs zurücksetzen, 237

STEP 7

- Ändern von Einstellungen, 31
 - ASi,
 - AS-i-Port, 154
 - Bausteine aus einer Online-CPU kopieren, 238
 - Bedienpanel, 30, 53, 230
 - Beobachten, 231, 232
 - Diagnosepuffer, 240
 - Drag & Drop zwischen Editoren, 32
 - Einfügen von Anweisungen, 28
 - Ergänzen von Eingängen oder Ausgängen in einer KOP- oder FUP-Anweisung, 29
 - Erweiterbare Eingänge und Ausgänge, 30
 - Ethernet-Port, 85
 - Favoriten, 29
 - Forcefunktion, 234
 - Forcen, 234
 - Gerätekonfiguration, 75
 - HSC-Konfiguration, 129
 - Konfigurieren der CPU, 80, 84
 - Konfigurieren der Module, 80, 84
 - Module hinzufügen, 78
 - Netzwerkverbindung, 132
 - Neues Gerät hinzufügen, 77
 - Nicht gesteckte Module, 36
 - Passwortschutz, 87
 - Portalansicht, 27
 - Prioritätsklasse (OB), 55
 - PROFIBUS, 151
 - PROFIBUS-Port, 151
 - PROFINET, 85
 - Projektansicht, 27
 - Schaltflächen RUN/STOP, 30
 - Startwerte eines DBs zurücksetzen, 237
 - Vergleichen und synchronisieren, 239
 - Werte eines DBs erfassen, 237
 - Zustand eines Codebausteins erfassen, 33
 - Zustand eines Codebausteins wiederherstellen, 33
- STEP 7 Webseiten, 4
- STEP 7-Programmierung
- Benutzerdefinierte Webseiten, 196
- Steuer-DB für benutzerdefinierte Webseiten
- Parameter WWW-Anweisung, 196
- STOP, Betriebszustand, 52
- Bedienpanel, 53
- Strukturierte Programmierung, 91
- Subnetzmaske, 84
- Support, 5
- Systemspeicherbyte, 83

T

- Tageszeit
- CPU online konfigurieren, 241
- Takt
- Merkerbyte, 83
- Taskcards
- Spalten und Überschriften, 32
- TCON
- Konfiguration, 140
 - Verbindungs-IDs, 137
 - Verbindungsparameter, 142
- TCON_Param, 142
- TCP
- Ad-hoc-Modus, 136
 - Parameter, 142
 - Protokolle, 135
 - Verbindungs-IDs, 137
 - Verbindungskonfiguration, 140
- TCP/IP-Kommunikation, 131, 135
- Technische Daten, 249
- Allgemeine technische Daten, 249
 - ATEX-Zulassung, 250
 - Bemessungsspannungen, 253
 - CB 1241 RS485, 296
 - CE-Zulassung, 249
 - CPU 1211C, 255
 - CPU 1212C, 255
 - CPU 1214C, 255
 - C-Tick-Zulassung, 251
 - cULus-Zulassung, 250
 - Darstellung Analogausgang (Spannung), 278
 - Darstellung Analogausgang (Strom), 278
 - Darstellung Analogeingang (Spannung), 275
 - Digitale Eingänge und Ausgänge (SM), 266
 - Elektromagnetische Verträglichkeit (EMV), 252
 - FM-Zertifizierung, 250
 - Industrienumgebungen, 251
 - Kommunikationsmodul CM 1241 RS232, 298
 - Kommunikationsmodul CM 1241 RS485, 298
 - Lebensdauer eines Relais, 255
 - SB 1231 AI 1x12 Bit, 271
 - SB 1232 AO 1x12 Bit, 271
 - Schrittantwortzeiten der Eingänge, 276
 - Schutz, 253
 - SM 1221 Signalmodul, 262
 - SM 1222 Signalmodul, 263
 - SM 1223 Signalmodul, 264, 265
 - SM 1231 AI 4 x 13 Bit, 272
 - SM 1232 AO 2 x 14 Bit, 272
 - SM 1232 AO 4 x 14 Bit, 272
 - SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 272
 - Umgebungsbedingungen, 252

Zulassung für das Seewesen, 251
 Technische Daten des CM 1241 RS485, 296
 Technische Daten für analoge Signalmodule
 SM 1231 AI 8 x 13 Bit, 272
 SM 1232 AO 2 x 14 Bit, 272
 SM 1232 AO 4 x 14 Bit, 272
 SM 1234 AI 4 x 13 Bit / AO 2 x 14 Bit, 272
 Technische Daten für das analoge Signalboard (SB)
 SB 1231 AI 1x12 Bit, 271
 SB 1232 AO 1x12 Bit, 271
 Technischer Support, 5
 Technischer Support von Siemens, 5
 Technologische Objekte
 HSC (schneller Zähler), 125
 Telecontrol, 162
 TeleService über GPRS, 161, 162
 Temporärer Speicher (L), 66
 Testen
 Laden im Betriebszustand RUN, 245
 Thermischer Bereich, 21
 TIA-Portal
 Gerätekonfiguration, 75
 Konfigurieren der CPU, 80, 84
 Konfigurieren der Module, 84
 Neues Gerät hinzufügen, 77
 Portalansicht, 27
 PROFINET, 85
 Projektansicht, 27
 TRCV
 Ad-hoc-Modus, 136
 Verbindungs-IDs, 137
 TRCV_C
 Ad-hoc-Modus, 136
 Konfiguration, 140
 Verbindungs-IDs, 137
 Verbindungsparameter, 142
 TRUNC (Ganzzahl erzeugen), 106
 TS-Adapter, 16
 TSAP, 135
 TSAP (Transport Service Access Point), 142
 TSEND
 Verbindungs-IDs, 137
 TSEND_C
 Konfiguration, 140
 Verbindungs-IDs, 137
 Verbindungsparameter, 142
 TURCV
 Konfiguration, 140
 Verbindungsparameter, 142
 TUSEND
 Konfiguration, 140
 Parameter, 142

U

Überwachung
 Anweisung LED, 122
 Überwachung der Speicherauslastung, online, 230
 UDP
 Parameter, 142
 Verbindungskonfiguration, 140
 UDP-Protokoll, 135
 Umgebungsbedingungen
 Betriebsbedingungen, 252
 Industrienumgebungen, 251
 Transport und Lagerung, 252
 USS-Protokoll, 165

V

Variablen
 Beobachten, 231
 Erste Schritte, 38, 42
 Forcefunktion, 234
 Forcen, 234
 Zustand, 231
 Verbindungen
 Partner, 157
 Verbindungen mit mehreren Teilnehmern
 Ethernet-Protokolle, 156
 Verbindungsarten, 156
 Vergleichsanweisungen, 104
 Vergleichstabelle
 CPU-Varianten, 14
 HMI-Geräte, 20
 Module, 16
 Versionen von Anweisungen, 32
 Visualisierung
 HMI-Geräte, 20

W

Warteschlangen, 56
 Webseiten
 STEP 7, 4
 Webseiten, benutzerdefiniert, 193
 Webserver, 189
 Einschränkungen, 190
 Weckalarm-OB, 55
 Werte eines Online-DBs erfassen, 237
 WWW, 196

Z

Zähler

- Anzahl, 15
- Funktionsweise des HSC, 125
- Größe, 15
- HSC (schneller Zähler), 124
- HSC-Konfiguration, 129

Zähleranweisungen, 113

Zeiten

- Anzahl, 15
- Größe, 15

Zeitfehler-OB, 56

Zugriffsschutz

- CPU, 86

Zulassung für das Seewesen, 251

Zulassungen

- ATEX-Zulassung, 250
- CE-Zulassung, 249
- C-Tick-Zulassung, 251
- cULus-Zulassung, 250
- FM-Zertifizierung, 250
- Zulassung für das Seewesen, 251

Zurücksetzen auf Werkseinstellungen, 242

Zustand

- Anweisung LED, 122
- LED-Anzeigen (Kommunikationsschnittstelle), 166

Zustand eines Codebausteins erfassen, 33

Zustand eines Codebausteins wiederherstellen, 33

Zyklus

- Forcefunktion, 234
- Forcen, 234

Zykluszeitüberwachung, 230