

SIEMENS

SIMATIC

S7-1200 Easy Book

Gerätehandbuch

Vorwort

Einführung in die leistungsstarke und flexible S7-1200

1

STEP 7 Basic vereinfacht Ihre Arbeit

2

Erste Schritte

3

SPS-Grundlagen leicht gemacht

4

Programmiergrundlagen leicht gemacht

5

Einfache Kommunikation zwischen Geräten

6

Einfaches Arbeiten mit den integrierten Impulsgeneratoren

7

Einfaches Arbeiten mit den Online-Tools

8

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 Basic 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 Basic 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 92) und bis zu 6 schnelle Zähler (Seite 88)
- In die CPU-Module integrierte E/A bieten 6 bis 14 Eingänge sowie 4 bis 10 Ausgänge



Bereitstellung zusätzlicher E/A (Seite 9) durch Signalmodule für Gleichstrom-, Relais- oder analoge E/A sowie durch an der CPU-Vorderseite einzurastende innovative Signalboards.

Die SIMATIC HMI Basic Panels (Seite 10) 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*. Auf der folgenden Website können Sie außerdem Informationen zu einzelnen Produkten finden oder sich an Mitarbeiter des technischen Supports wenden:

<http://www.siemens.com/automation/support-request>

Beachten Sie für weitere Informationen zu UL- und FM-Zulassung, CE-Kennzeichnung, C-Tick-Zulassung und anderen Normen die technischen Daten (Seite 107).

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	7
1.1	Erweiterung der CPU-Funktionen	9
1.2	HMI Basic Panels.....	10
1.3	Einbaumaße und notwendiger Freiraum	11
2	STEP 7 Basic vereinfacht Ihre Arbeit	13
2.1	Zugriff auf die Hilfe in jeder Situation	14
2.1.1	Thema aus der Online-Hilfe drucken	15
2.2	Bedienerfreundliche Werkzeuge.....	16
2.2.1	Einfaches Einfügen von Anweisungen in Ihr Anwenderprogramm.....	16
2.2.2	Schneller Zugriff auf viel verwendete Operationen über die Funktionsleiste	16
2.2.3	Einfaches Drag & Drop zwischen Editoren.....	17
2.2.4	Einfaches Ändern des CPU-Betriebszustands	17
2.2.5	Einfaches virtuelles "Abziehen" von Baugruppen ohne Verlust der Konfiguration.....	18
2.2.6	Einfaches Ändern des Erscheinungsbildes und der Konfiguration von STEP 7 Basic.....	19
3	Erste Schritte	21
4	SPS-Grundlagen leicht gemacht	33
4.1	Bei jedem Zyklus ausgeführte Arbeitsschritte.....	33
4.2	Betriebszustände der CPU	35
4.3	Speicherbereiche, Adressierung und Datentypen	36
4.4	Ausführung des Anwenderprogramms	40
4.5	Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach.....	43
5	Programmiergrundlagen leicht gemacht	45
5.1	Einfache Erstellung der Gerätekonfiguration	45
5.1.1	Konfigurieren des CPU- und Modulbetriebs	50
5.1.2	Konfigurieren der IP-Adresse der CPU.....	52
5.2	Einfaches Entwerfen Ihres Anwenderprogramms	53
5.2.1	OBs für die Organisation Ihres Anwenderprogramms.....	56
5.2.2	FBs und FCs vereinfachen die Programmierung der modularen Aufgaben.....	57
5.2.3	Datenbausteine sorgen für die unkomplizierte Speicherung von Programmdateien	59
5.3	Problemloser Einsatz der leistungsstarken Programmiersprachen.....	60
5.3.1	Bereitstellung der erwartbaren grundlegenden Operationen.....	61
5.4	Weitere Funktionen für einfacheres Programmieren	71
5.4.1	Systemmerker und Taktmerker bieten Standardfunktionen	71
5.4.2	Beobachtungstabellen vereinfachen die Überwachung des Anwenderprogramms	73
5.4.3	Projektbibliotheken und globale Bibliotheken für einfachen Zugriff.....	74
5.4.4	Querverweis zum Anzeigen der Verwendung	75
5.4.5	Aufrufstruktur zur Prüfung der Aufrufhierarchie	76

6	Einfache Kommunikation zwischen Geräten.....	79
6.1	PROFINET-Operationen (T-Bausteine).....	80
6.2	Kommunikationsprotokolle PtP, USS und Modbus.....	81
6.2.1	PtP-Operationen	82
6.2.2	Bibliothek der USS-Operationen.....	83
6.2.3	Bibliothek der Modbus-Operationen	85
7	Einfaches Arbeiten mit den integrierten Impulsgeneratoren.....	87
7.1	Schnelle Zähler	88
7.2	Impulsdauermodulation (PWM)	92
8	Einfaches Arbeiten mit den Online-Tools.....	97
8.1	Online-Verbindung mit einer CPU herstellen	97
8.2	Laden einer IP-Adresse in eine Online-CPU	98
8.3	Interaktion mit der Online-CPU	99
8.4	Laden von der Online-CPU	100
8.5	Vergleichen von Offline- und Online-CPU's.....	102
8.6	Diagnoseereignisse anzeigen	103
8.7	Beobachtungstabelle zur Überwachung der CPU verwenden.....	103
8.8	Variablen in der CPU forcen	105
A	Technische Daten.....	107
A.1	Allgemeine technische Daten.....	107
A.2	CPU-Module.....	112
A.3	Signalboards	117
A.4	Digitale Signalmodule	119
A.5	Analoge Signalmodule	124
A.6	Kommunikationsmodule.....	126
	Index.....	129

Einführung in die leistungsstarke und flexible

S7-1200

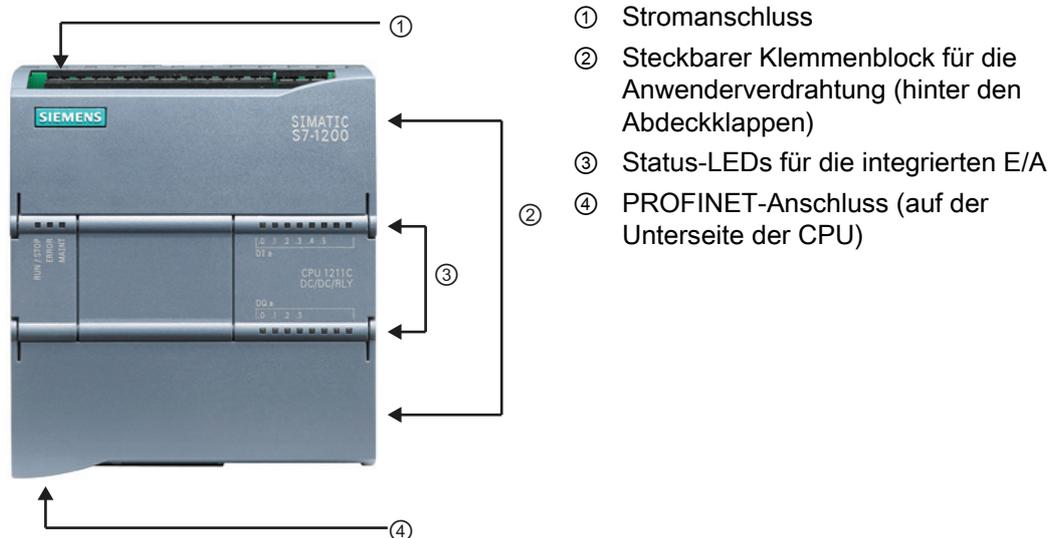
1

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.

Für die Kommunikation mit einem Programmiergerät verfügt die CPU über einen integrierten PROFINET-Port. Im PROFINET-Netzwerk kann die CPU mit HMI-Bediengeräten oder anderen CPUs kommunizieren.

Als Sicherheitseinrichtung für Ihre Anwendung ist jede S7-1200 CPU mit einem Passwortschutz ausgestattet, mit dem der Zugriff auf die CPU-Funktionen nach Bedarf eingerichtet werden kann.



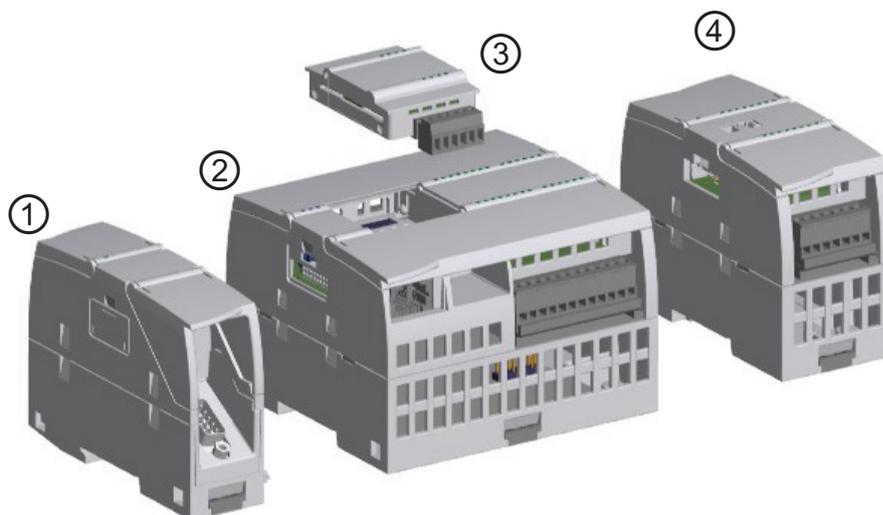
Merkmal	CPU 1211C	CPU 1212C	CPU 1214C
Abmessungen (mm)	90 x 100 x 75	90 x 100 x 75	110 x 100 x 75
Anwenderspeicher	<ul style="list-style-type: none"> • Arbeitsspeicher • Ladespeicher • Remanenter Speicher 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB
Integrierte E/A	<ul style="list-style-type: none"> • Digital • Analog 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB 	<ul style="list-style-type: none"> • 50 KB • 2 MB • 2 KB
Größe des Prozessabbilds	<ul style="list-style-type: none"> • Eingänge • Ausgänge 	<ul style="list-style-type: none"> • 6 Eingänge 4 Ausgänge • 2 Eingänge 	<ul style="list-style-type: none"> • 8 Eingänge 6 Ausgänge • 2 Eingänge
Merker (M)	<ul style="list-style-type: none"> • 14 Eingänge 10 Ausgänge • 2 Eingänge 	<ul style="list-style-type: none"> • 1024 Byte • 1024 Byte 	<ul style="list-style-type: none"> • 1024 Byte • 1024 Byte
Zusätzliche Signalmodule	4096 Byte	4096 Byte	8192 Byte
Signalboard	Keine	2	8
Kommunikationsmodule	1	1	1
Schnelle Zähler	3	3	3
• Einphasenzähler	<ul style="list-style-type: none"> • 3 bei 100 kHz 	<ul style="list-style-type: none"> • 3 bei 100 kHz • 1 bei 30 kHz 	<ul style="list-style-type: none"> • 3 bei 100 kHz • 3 bei 30 kHz
• A/B-Zähler	<ul style="list-style-type: none"> • 3 bei 80 kHz 	<ul style="list-style-type: none"> • 3 bei 80 kHz • 1 bei 20 kHz 	<ul style="list-style-type: none"> • 3 bei 80 kHz • 3 bei 20 kHz
Impulsausgänge ¹	2	2	2
Memory Card (optional)	Ja	Ja	Ja
Pufferung Echtzeituhr	Typ. 10 Tage/min. 6 Tage bei 40 Grad C		
Ausführungszeit arithm. Operationen	18 µs/Operation		
Ausführungszeit Boolesche Operationen	0,1 µs/Operation		

¹ Impulsausgänge werden nur von CPUs mit Gleichstromausgängen (Nicht-Relais-Ausgängen) unterstützt.

Die verschiedenen CPU-Ausführungen bieten eine Vielfalt an Leistungsmerkmalen und Funktionen, damit Sie effektive Lösungen für verschiedenste Anwendungen anlegen können. Ausführliche Informationen zu bestimmten CPUs finden Sie in den technischen Daten (Seite 112).

1.1 Erweiterung der CPU-Funktionen

Die Produktfamilie S7-1200 bietet eine Vielzahl von Signalmodulen und Signalboards zur Erweiterung der CPU. Sie können auch zusätzliche Kommunikationsmodule installieren, um andere Kommunikationsprotokolle zu unterstützen. Ausführliche Informationen zu bestimmten Modulen finden Sie in den technischen Daten (Seite 107).



① Kommunikationsmodul (CM)

② CPU

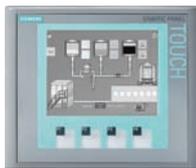
③ Signalboard (SB)

④ Signalmodul (SM)

Modul		Nur Eingang	Nur Ausgang	Ein-/Ausgang kombiniert
Signalmodul (SM)	Digital	8 x DC-Eingänge	8 x DC-Ausgänge / 8 x Relaisausgänge	8 x DC-Eingänge/8 x DC-Ausgänge 8 x DC-Eingänge/8 x Relaisausgänge
		16 x DC-Eingänge	16 x DC-Ausgänge / 16 x Relaisausgänge	16 x DC-Eingänge/16 x DC-Ausgänge 16 x DC-Eingänge/16 x Relaisausgänge
	Analog	4 x Analogeingänge 8 x Analogeingänge	2 x Analogausgänge 4 x Analogausgänge	4 x Analogeingänge/2 x Analogausgänge
Signalboard (SB)	Digital	-	-	2 x DC-Eingänge / 2 x DC-Ausgänge
	Analog	-	1 x Analogausgang	-
Kommunikationsmodul (CM)				
<ul style="list-style-type: none"> • RS485 • RS232 				

1.2 HMI Basic Panels

Visualisierung gehört heute bei den meisten Maschinen zum Standard-Repertoire. 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.



KTP 400 Basic PN

- Mono (STN, Graustufen)
- 4"-Touchscreen mit 4 taktilen Tasten
- Hochformat oder Querformat
- Größe: 3.8"
- Auflösung: 320 x 240
- 128 Variablen
- 50 Prozessbilder
- 200 Meldungen
- 25 Kurven
- 32 KB-Rezepturspeicher
- 5 Rezepturen, 20 Datensätze, 20 Einträge



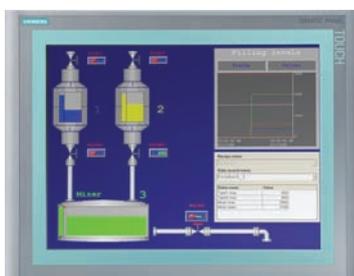
KTP 600 Basic PN

- Farbe (TFT, 256 Farben) oder Mono (STN, Graustufen)
- 6"-Touchscreen mit 6 taktilen Tasten
- Hochformat oder Querformat
- Größe: 5.7"
- Auflösung: 320 x 240
- 128 Variablen
- 50 Prozessbilder
- 200 Meldungen
- 25 Kurven
- 32 KB-Rezepturspeicher
- 5 Rezepturen, 20 Datensätze, 20 Einträge



KTP1000 Basic PN

- Farbe (TFT, 256 Farben)
- 10"-Touchscreen mit 8 taktilen Tasten
- Größe: 10.4"
- Auflösung: 640 x 480
- 256 Variablen
- 50 Prozessbilder
- 200 Meldungen
- 25 Kurven
- 32 KB-Rezepturspeicher
- 5 Rezepturen, 20 Datensätze, 20 Einträge



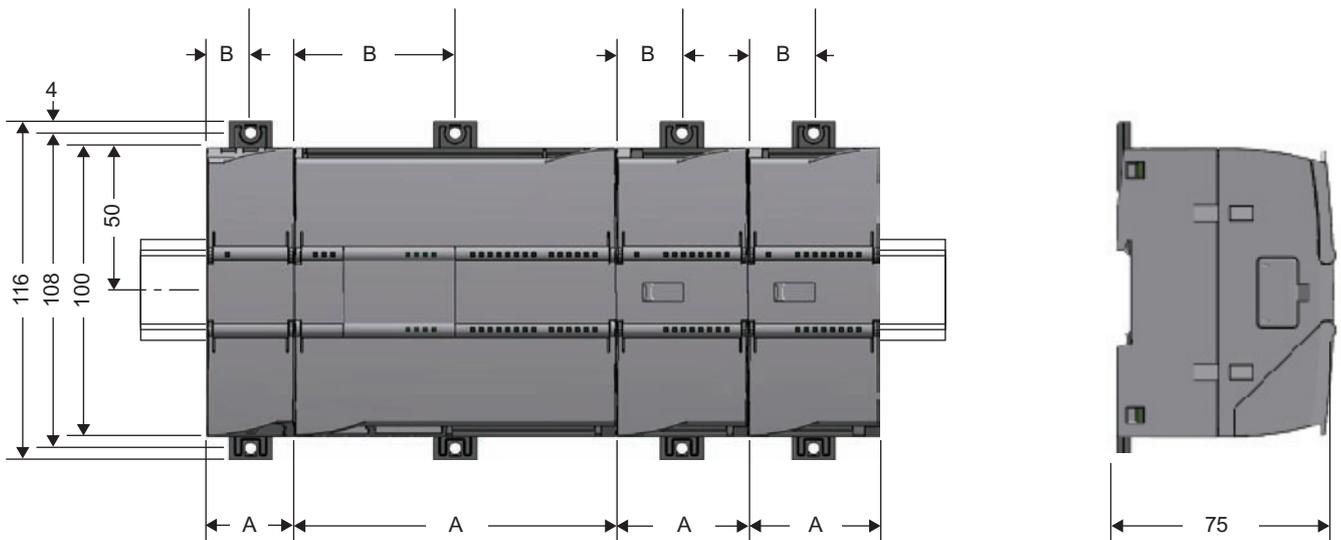
TP1500 Basic PN

- Farbe (TFT, 256 Farben)
- 15"-Touchscreen
- Größe: 15.1"
- Auflösung: 1024 x 768
- 256 Variablen
- 50 Prozessbilder
- 200 Meldungen
- 25 Kurven
- 32 KB-Rezepturspeicher (integrierter Flash)
- 5 Rezepturen, 20 Datensätze, 20 Einträge

1.3 Einbaumaße und notwendiger Freiraum

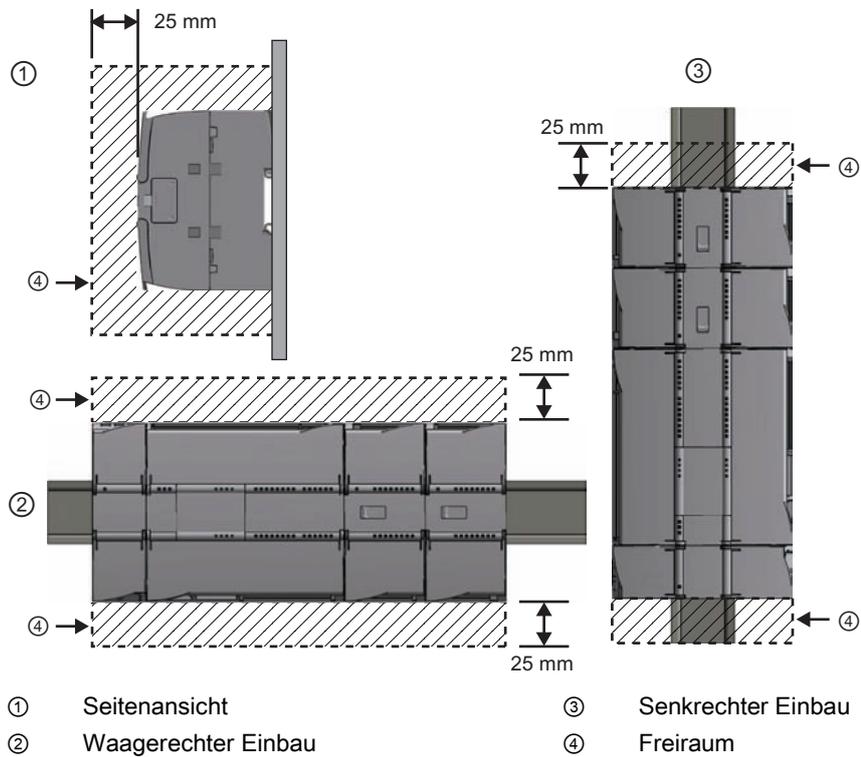
Die Steuerung S7-1200 wurde so ausgelegt, dass sie einfach einzubauen ist. Beim Schalttafeleinbau ebenso wie bei der Montage auf einer Standard-Hutschiene gestattet die kompakte Baugröße eine optimale Platzausnutzung.

Die CPUs, SMs und CMs sind für die Montage auf einer Standard-Hutschiene und für den Schalttafeleinbau geeignet. 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.



S71200 Geräte		Breite A	Breite B
CPU	CPU 1211C und CPU 1212C	90 mm	45 mm
	CPU 1214C	110 mm	55 mm
Signalmodul (SM)	8 und 16 E/A, DC und Relais (8E, 16E, 8A, 16A, 8E/8A) Analog (4AE, 8AE, 4AE/4AA, 2AA, 4AA)	45 mm	22,5 mm
	16E/16A Relais (16E/16A)	70 mm	35 mm
	Kommunikationsmodul (CM)	CM 1241 RS232 und CM 1241 RS485	30 mm

1.3 Einbaumaße und notwendiger Freiraum



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*.

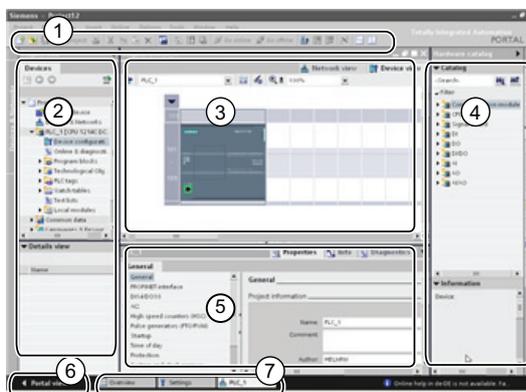
STEP 7 Basic vereinfacht Ihre Arbeit

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



Die Portalansicht bietet eine funktionale Ansicht der Projektaufgaben und gliedert die verfügbaren Funktionen nach zu erledigenden Aufgaben. Sie können auf einfache Weise festlegen, wie Sie vorgehen möchten und welche Aufgaben Sie auswählen.

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



Die Projektansicht ermöglicht den Zugriff auf alle Komponenten in einem Projekt.

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

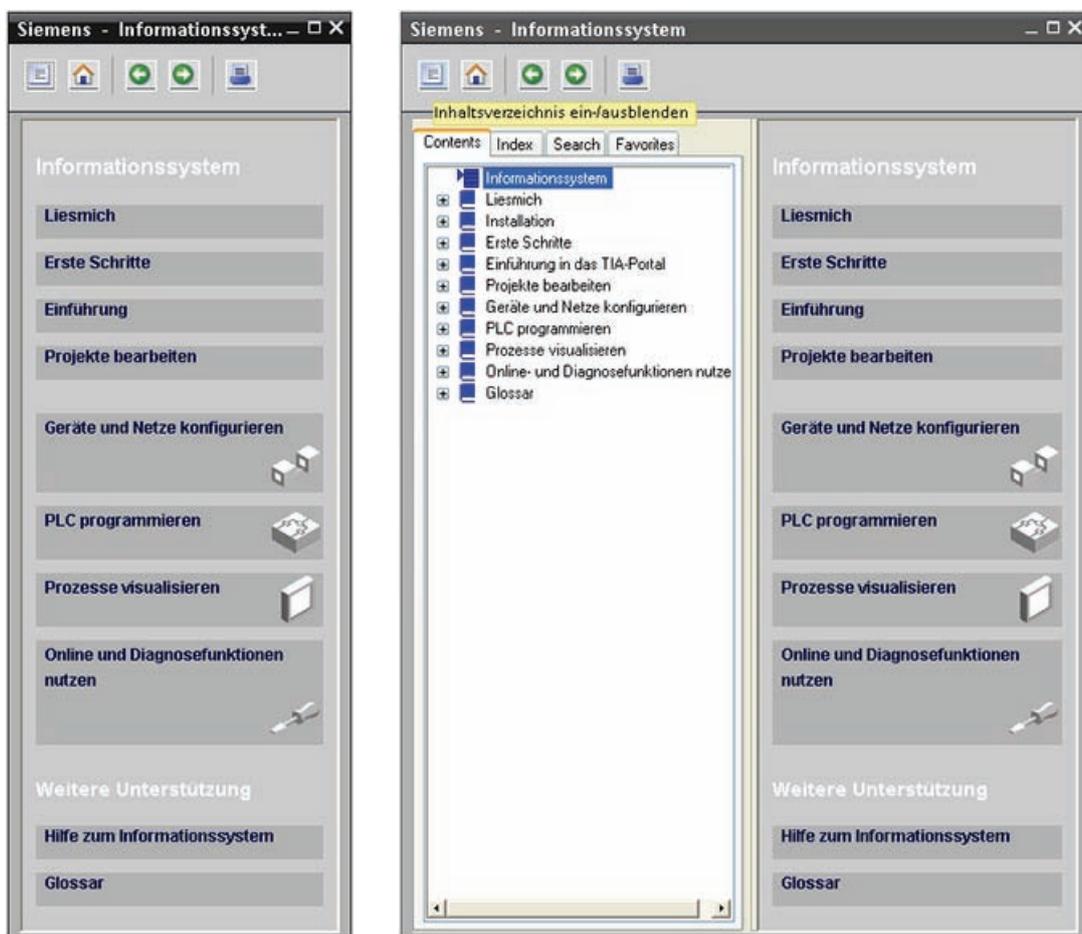
Da sich alle Komponenten an einer Stelle befinden, haben Sie schnellen Zugriff auf jeden Bereich Ihres Projekts. Beispielsweise zeigt das Inspektorfenster die Eigenschaften und weitere Informationen für das Objekt an, das im Arbeitsbereich ausgewählt wurde. Für die verschiedenen von Ihnen gewählten Objekte 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 Zugriff auf die Hilfe in jeder Situation

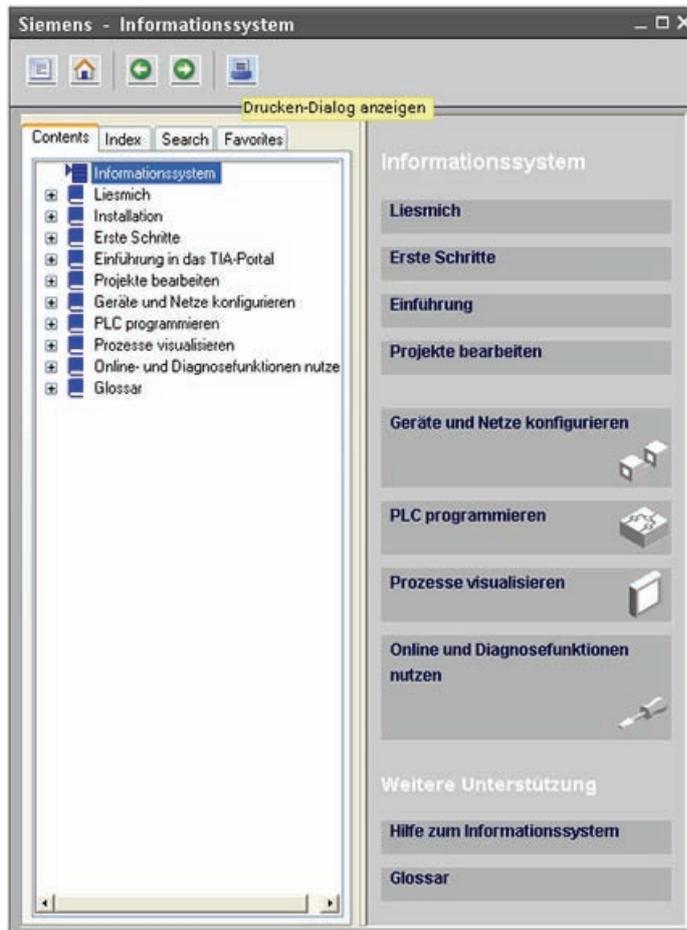
Damit Sie sich zusätzliche Informationen verschaffen oder schnelle und wirkungsvolle Hilfe bei Problemen finden können, bietet Ihnen das TIA-Portal dort intelligente Unterstützung, wo Sie sie benötigen: Beispielsweise sind einige Tooltips (zum Beispiel für die Operationen) "kaskadierend" und bieten so zusätzliche Informationen. Ein schwarzes Dreieck neben dem Tooltip weist darauf hin, dass weitere Informationen verfügbar sind.

STEP 7 Basic bietet Ihnen ein umfangreiches Online-Informations- und Hilfesystem, in dem alle von Ihnen installierten SIMATIC TIA-Produkte beschrieben sind. Das Informationssystem wird in einem Fenster geöffnet, das die Arbeitsbereiche nicht verdeckt. Um das Inhaltsverzeichnis anzuzeigen und das Hilfenfenster abzukoppeln, klicken Sie auf die Schaltfläche "Inhalt anzeigen/ausblenden" im Informationssystem. Sie können dann die Größe des Hilfenfensters ändern.



Wenn STEP 7 Basic maximiert ist, können Sie das Hilfenfenster über die Schaltfläche "Inhalt anzeigen/ausblenden" nicht abkoppeln. Um das Hilfenfenster abzukoppeln, klicken Sie in STEP 7 Basic auf die Schaltfläche zum Wiederherstellen. Sie können dann das Hilfenfenster verschieben und seine Größe ändern.

2.1.1 Thema aus der Online-Hilfe drucken



Um Themen aus dem Informationssystem zu drucken, klicken Sie im Hilfenfenster auf die Schaltfläche "Drucken".



Im Dialog "Drucken" können Sie die zu druckenden Themen auswählen. Achten Sie darauf, dass das Teilfenster ein Thema anzeigt. Dann können Sie jedes beliebige andere Thema zum Drucken auswählen. Klicken Sie auf die Schaltfläche "Drucken", um die ausgewählten Themen an Ihren Drucker zu senden.

2.2 Bedienerfreundliche Werkzeuge

2.2.1 Einfaches Einfügen von Anweisungen in Ihr Anwenderprogramm

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

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



2.2.2 Schneller Zugriff auf viel verwendete Operationen über die Funktionsleiste



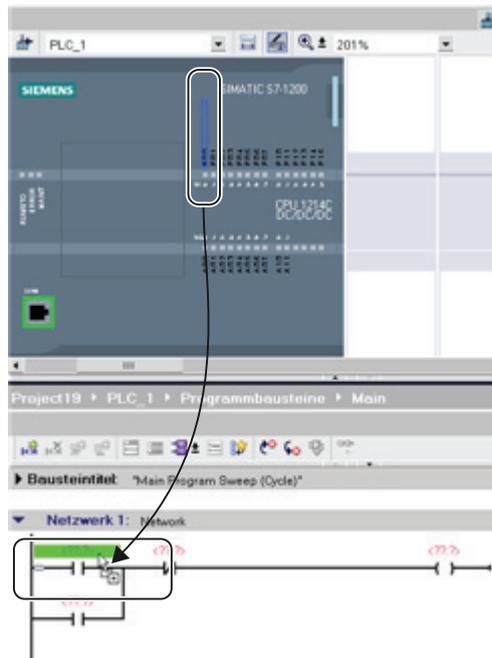
STEP 7 Basic enthält eine Funktionsleiste "Favoriten" für den schnellen Zugriff auf Operationen, die Sie häufig verwenden.

Um eine Operation in Ihr Netzwerk einzufügen, genügt ein Klick auf das entsprechende Symbol.

Die Funktionsleiste "Favoriten" lässt sich problemlos anpassen und durch neue Operationen erweitern. Fügen Sie die gewünschte Operation einfach mit "Drag & Drop" den "Favoriten" hinzu. Für den Zugriff auf die Operation genügt jetzt ein Klick!



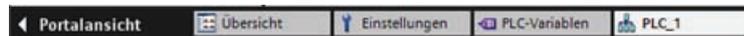
2.2.3 Einfaches Drag & Drop zwischen Editoren



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



2.2.4 Einfaches Ändern des CPU-Betriebszustands

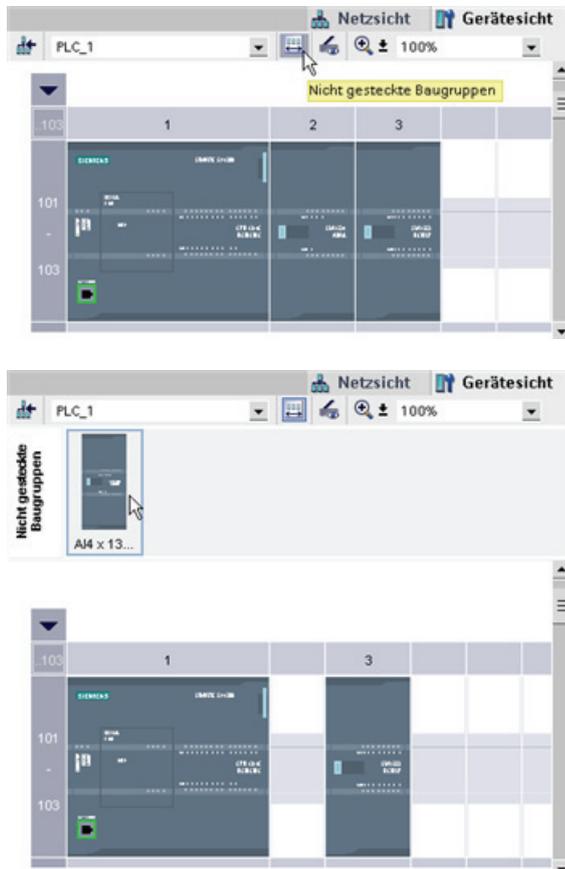
Die CPU verfügt nicht über einen physischen Schalter zum Ändern des Betriebszustands (STOP oder RUN). Beim Konfigurieren der CPU in der Gerätekonfiguration legen Sie das Anlaufverhalten der CPU über ihre Eigenschaften fest (Seite 50). Im Portal "Online & Diagnose" steht Ihnen ein Bedienpanel zur Verfügung, 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.)

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

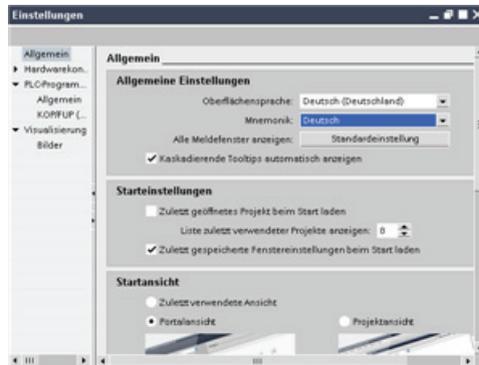


STEP 7 Basic bietet Ihnen eine virtuelle Ablage für "nicht gesteckte" Baugruppen. Sie können eine Baugruppe vom Baugruppenträger "abziehen" und dabei die Konfiguration der Baugruppe speichern. Diese abgezogenen Baugruppen werden mit dem Projekt zusammen gespeichert; dadurch können Sie sie später wieder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

Der Austausch einer Baugruppe hat keine Auswirkungen auf die PLC-Variablen, sofern sich die Basisadressierung der Baugruppe nicht ändert. So können Sie beispielsweise ein digitales SM-Modul mit 8 Eingängen entweder durch ein digitales 8x8-Kombinationsmodul oder ein digitales SM-Modul mit 16 Eingängen ersetzen.

2.2.6 Einfaches Ändern des Erscheinungsbildes und der Konfiguration von STEP 7 Basic

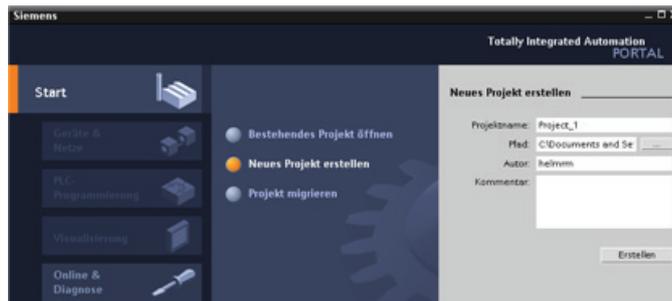


Sie haben zahlreiche Einstellmöglichkeiten; diese betreffen z. B. das Aussehen der Bedienoberfläche, die Sprache oder das Verzeichnis zum Speichern Ihrer Arbeitsergebnisse.

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

Erste Schritte

Das Arbeiten mit STEP 7 Basic ist einfach! Auf den nächsten Seiten erfahren Sie, wie schnell es geht, ein Projekt anzulegen.



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".

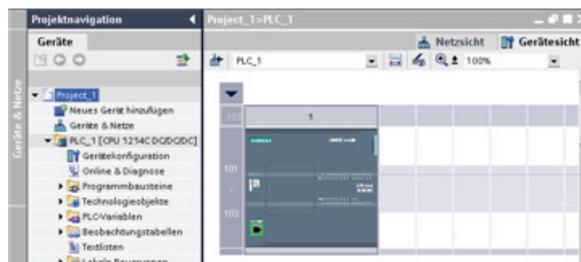
Klicken Sie auf "Neues Gerät hinzufügen".



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.

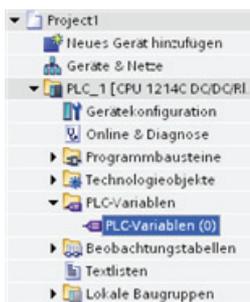


In der Gerätesicht wird die hinzugefügte CPU angezeigt.

Variablen für die E/A der CPU anlegen

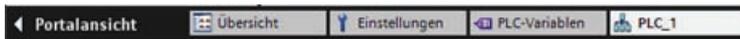
Hinweis

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

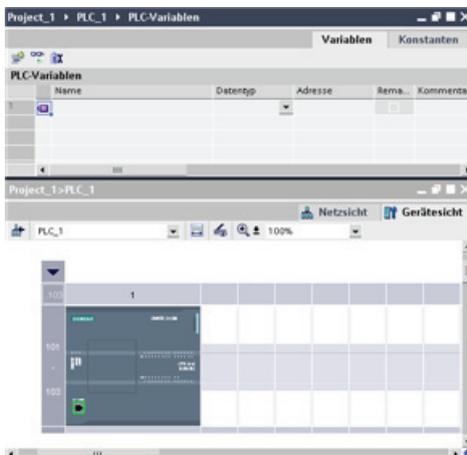


Bei geöffnetem Geräteeditor können Sie eine Variablen-tabelle öffnen.

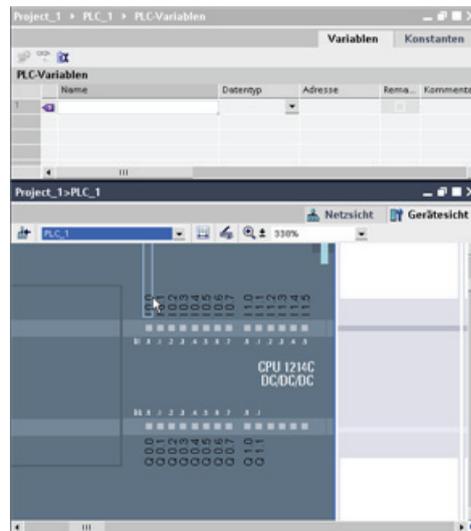
Die geöffneten Editoren werden in der Editorleiste angezeigt.



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

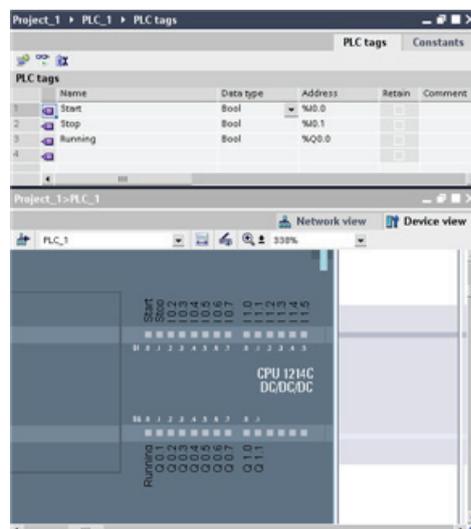


STEP 7 Basic zeigt sowohl die Variablen-tabelle als auch den Geräteeditor an.



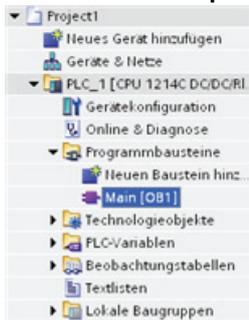
Zeigen Sie die Gerätekonfiguration mit einem Zoomfaktor über 200 % an, sodass die E/A lesbar und auswählbar sind.

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

Einfaches Netzwerk im Anwenderprogramm anlegen



Ihr Programmcode besteht aus Anweisungen, die von der Steuerung 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.

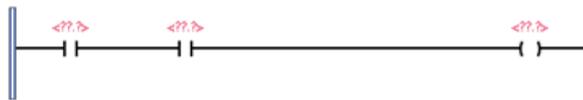
Um den Programmiereditor zu öffnen, gehen Sie folgendermaßen vor:

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:

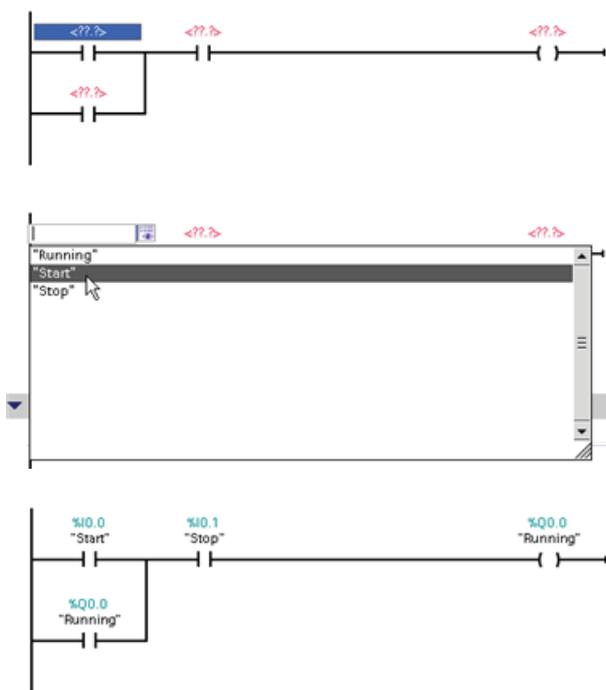
1. Um eine Verzweigung zur Schiene des Netzwerks hinzuzufügen, klicken Sie auf das Symbol für "Verzweigung öffnen".
2. Fügen Sie einen weiteren Schließer in die geöffnete Verzweigung ein
3. Ziehen Sie den Pfeil mit zwei Spitzen zu einem Verbindungspunkt (dem grünen Quadrat auf dem Strompfad) zwischen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

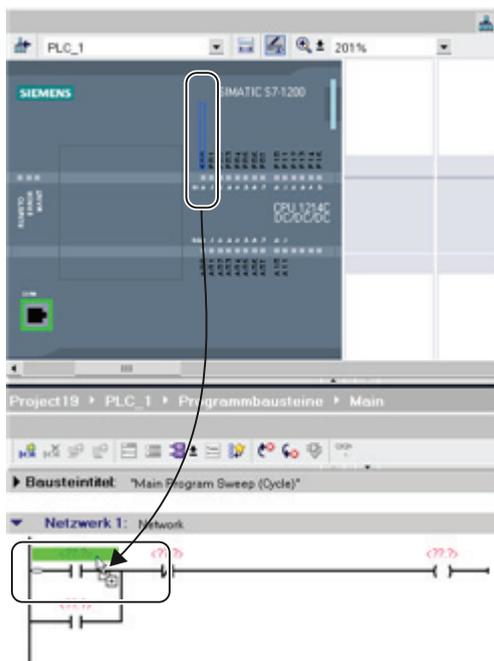
Sie haben ein Netzwerk mit KOP-Operationen angelegt. Sie können jetzt den Variablennamen die entsprechenden Anweisungen zuordnen.

Adressieren Sie die Operationen 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 17).

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

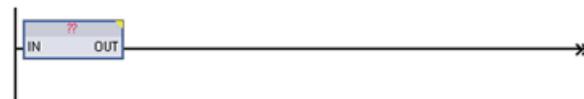
So können Sie die E/A der CPU aus der "Gerätekonfiguration" in die KOP-Operation im Programmiereditor ziehen und damit nicht nur die Adresse für die Anweisung, sondern gleichzeitig einen Eintrag in der PLC-Variablen-tabelle anlegen.

Dem zweiten Netzwerk eine arithmetische Operation hinzufügen

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



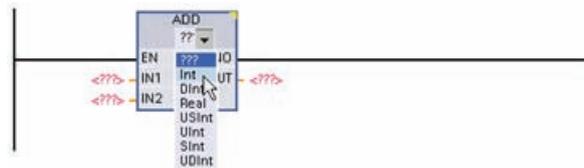
Klicken Sie in der Funktionsleiste "Favoriten" auf die allgemeine "Box"-Operation.



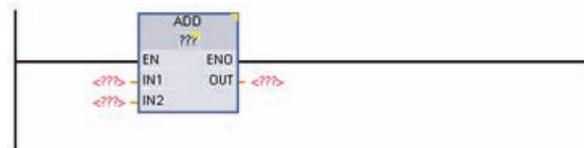
Unter der allgemeinen "Box"-Operation sind eine Reihe von Operationen verfügbar. Für dieses Beispiel erstellen Sie eine Operation ADD:



1. Öffnen Sie die Klappliste mit Operationen, indem Sie auf die gelbe Ecke der "Box"-Operation klicken.
2. Blättern Sie in der Liste nach unten und wählen Sie die Operation ADD aus.
3. Klicken Sie auf die gelbe Ecke neben dem "?", um den Datentyp für Eingaben und Ausgaben zu wählen.



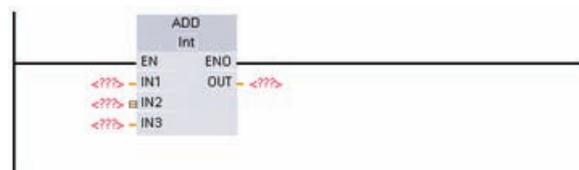
Jetzt können Sie die Variablen (oder Speicheradressen) für die Werte eingeben, die mit der Operation ADD verwendet werden sollen.



Sie können außerdem für bestimmte Operationen zusätzliche Eingänge angeben:

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

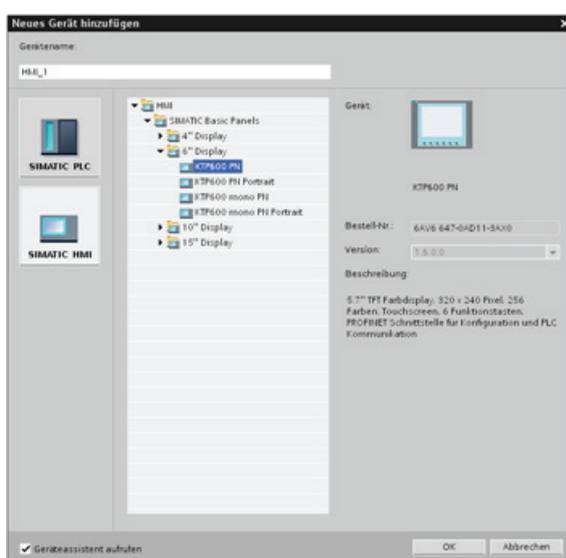
Die Operation ADD verwendet nun drei Eingänge.



Hinzufügen eines HMI-Geräts zum Projekt



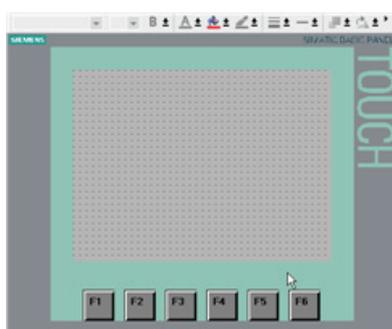
Das Hinzufügen eines HMI-Geräts zum Projekt ist einfach!



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.

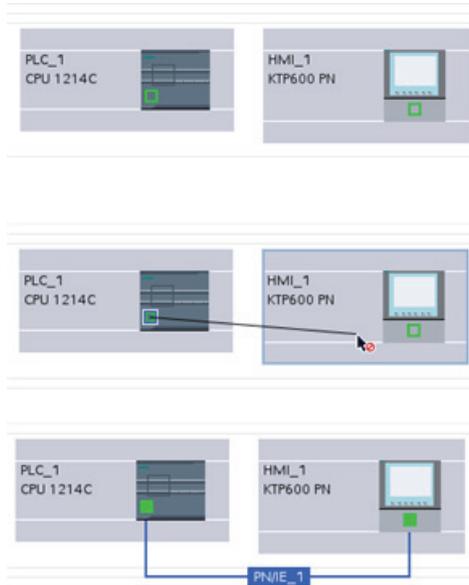


Das HMI-Gerät wird zum Projekt hinzugefügt.

STEP 7 Basic 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 Basic ein einfaches HMI-Standardbild an.

Netzwerkverbindung zwischen CPU und HMI-Gerät herstellen



Das Erstellen einer Netzwerkverbindung ist einfach!

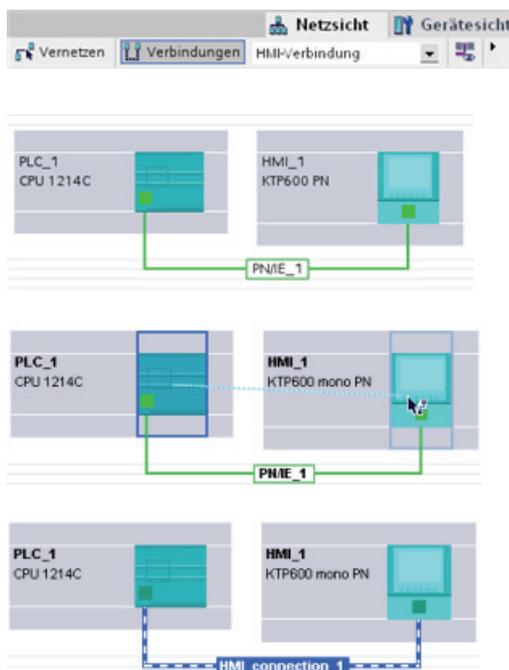
Navigieren Sie zu "Geräte & Netze" und wählen Sie die Netzsicht, um CPU und HMI-Gerät anzuzeigen.

Um ein PROFINET-Netzwerk zu erstellen, ziehen Sie eine Linie von dem grünen Quadrat (Ethernet-Port) auf dem Gerät zu dem grünen Quadrat des anderen Geräts.

Für die beiden Geräte wird eine Netzwerkverbindung hergestellt.

Erstellen einer HMI-Verbindung zur gemeinsamen Nutzung der Variablen

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 "HMI-Verbindung".

Hierdurch erscheinen die beiden Geräte in blauer Anzeigefarbe.

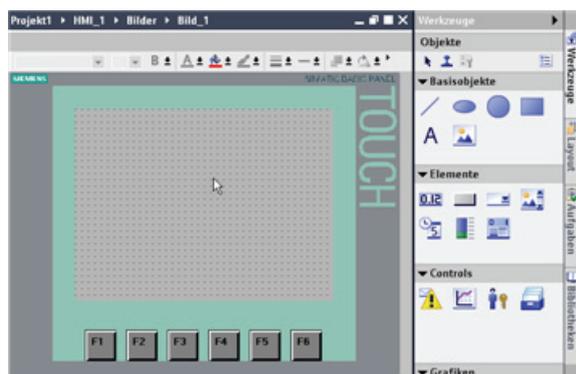
Wählen Sie die CPU aus und ziehen Sie die Linie zum HMI-Gerät.

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.

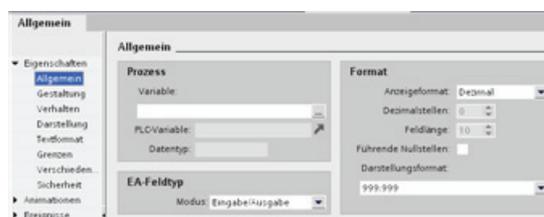
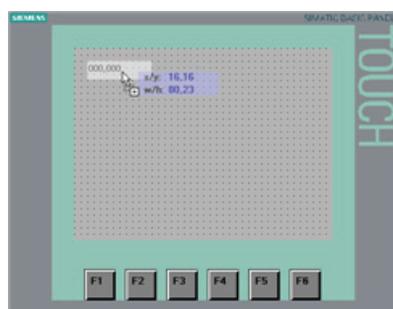
HMI-Bild anlegen



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

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

Um ein Element einzufügen, brauchen Sie das Element nur auf das Bild zu ziehen und dort abzulegen.

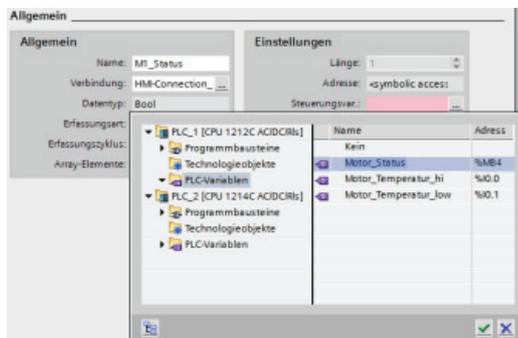


Das Konfigurieren von Aussehen und Verhalten des Elements erfolgt über die Elementeigenschaften im Inspektorfenster.

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

PLC-Variable für ein 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 Schaltfläche neben dem Feld "Verbindungen" 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

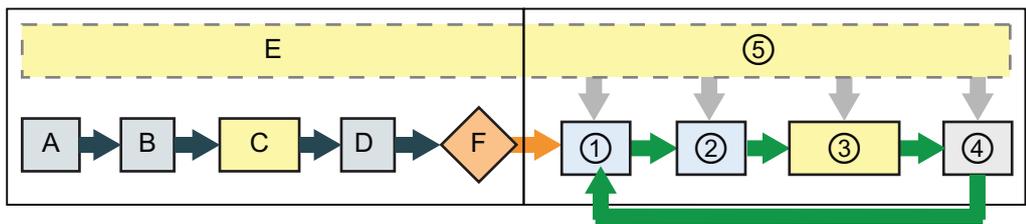
Jeder Zyklus umfasst das Schreiben der Ausgänge, das Lesen der Eingänge, das Bearbeiten der Anweisungen des Anwenderprogramms und die Durchführung der Systemwartung oder Hintergrundverarbeitung. Dieser Zyklus wird als Abtastzyklus oder Abtastung bezeichnet. Unter Standardbedingungen werden alle digitalen 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 Anwenderoperationen konsistent bleiben.

Die CPU führt die Logik der Anwenderoperationen 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.



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 Interruptereignisse 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 Interrupts und Kommunikation werden in allen Teilen des Zyklus bearbeitet

Dieser Vorgang sorgt während der gesamten Ausführung der Anwenderoperationen in dem jeweiligen Zyklus für eine konsistente Logik und verhindert ein Pendeln der physischen Ausgänge mit mehrmaligen Zustandswechseln im Prozessabbild der Ausgänge.

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. Interruptereignisse werden im Betriebszustand STARTUP nicht bearbeitet.
- Im Betriebszustand RUN wird der Zyklus wiederholt ausgeführt. Interruptereignisse können auftreten und an beliebigen Punkten innerhalb der Programmzyklusphase abgearbeitet werden.

Hinweis

Im Betriebszustand RUN der CPU kann kein Projekt geladen werden. Das Laden Ihres Projekts ist **nur** im Betriebszustand STOP der CPU möglich.

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.

Sie können das Anlaufverhalten der CPU und die Anlaufart mit Hilfe der Programmiersoftware festlegen. 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. 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). Zu diesem Zweck können Sie das CPU-Bedienpanel in den "Online-Tools" von STEP 7 Basic verwenden.

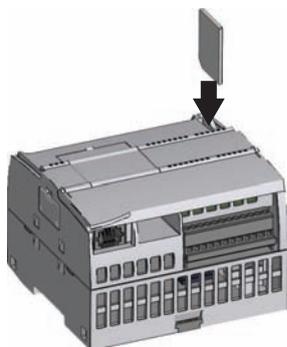
Sie können ferner eine STP-Operation in Ihr Programm einfügen, um die CPU in den Betriebszustand STOP zu versetzen. Auf diese Weise können Sie die Bearbeitung Ihres Programms abhängig von der Programmlogik unterbrechen.

4.3 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. Die Memory Card unterstützt einen größeren Speicherbereich als den in der CPU integrierten.
- Der Arbeitsspeicher ist ein flüchtiger Speicher für einige Elemente des Anwenderprojekts während der Bearbeitung des Anwenderprogramms. Die CPU kopiert einige Elemente des Projekts 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. Bei einem Spannungsausfall verfügt die CPU über eine ausreichende Haltezeit für die Datenhaltung einer begrenzten Anzahl von Speicheradressen. Bei Spannungswiederkehr werden diese remanenten Werte dann 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.



Die CPU unterstützt nur eine vorformatierte SIMATIC Memory Card.

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

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:

- Mit der Übertragungskarte kopieren Sie Ihr Projekt ohne STEP 7 Basic in mehrere CPUs. Die Übertragungskarte kopiert ein gespeichertes Projekt von der Memory Card in den Speicher der CPU. Sie müssen die Übertragungskarte nach dem Kopieren des Programms in die CPU ziehen.
- Die Programmkarte nimmt die Stelle des CPU-Speichers ein. Alle CPU-Funktionen werden von der Programmkarte gesteuert. Wenn Sie die Programmkarte stecken, wird der interne Ladespeicher der CPU gelöscht (auch das Anwenderprogramm und ggf. geforcte E/A). Die CPU führt das Anwenderprogramm dann von der Programmkarte aus.

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

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.

Datentyp	Größe (Bit)	Bereich	Beispiel für konstanten Eintrag
Bool	1	0 bis 1	TRUE, FALSE, 0, 1
Byte	8	16#00 bis 16#FF	16#12, 16#AB
Word	16	16#0000 bis 16#FFFF	16#ABCD, 16#0001
DWord	32	16#00000000 bis 16#FFFFFFFF	16#02468ACE
Char	8	16#00 bis 16#FF	'A', 't', '@'
Sint	8	-128 bis 127	123, -123
Int	16	-32.768 bis 32.767	123, -123
Dint	32	-2.147.483.648 bis 2.147.483.647	123, -123
USInt	8	0 bis 255	123
UInt	16	0 bis 65.535	123
UDInt	32	0 bis 4.294.967.295	123
Real	32	+/-1,18 x 10 ⁻³⁸ bis +/-3,40 x 10 ³⁸	123,456, -3,4, -1,2E+12, 3,4E-3
LReal	64	+/-2.23 x 10 ⁻³⁰⁸ bis +/-1.79 x 10 ³⁰⁸	12345.123456789 -1.2E+40
Time	32	T#-24d_20h_31m_23s_648ms to T#24d_20h_31m_23s_647ms Gespeichert als: -2,147,483,648 ms to +2,147,483,647 ms	T#5m_30s 5#-2d T#1d_2h_15m_30x_45ms
String	Variable	0 bis 254 Zeichen in Bytegröße	'ABC'
DTL ¹	12 Byte	Minimum: DTL#1970-01-01-00:00:00.0 Maximum: DTL#2554-12-31-23:59:59.999 999 999	DTL#2008-12-16-20:30:20.250

¹ Der Datentyp DTL ist eine Struktur aus 12 Bytes, in der Informationen zum Datum und zur Uhrzeit in einer vordefinierten Struktur gespeichert werden. Sie können den Datentyp DTL entweder im temporären Speicher des Bausteins oder in einem DB definieren.

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

Format	Größe (Bit)	Bereich	Beispiele
BCD16	16	-999 bis 999	123, -123
BCD32	32	-9999999 bis 9999999	1234567, -1234567

Speicherbereiche und Adressierung

STEP 7 Basic 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 Operationsparameter 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.

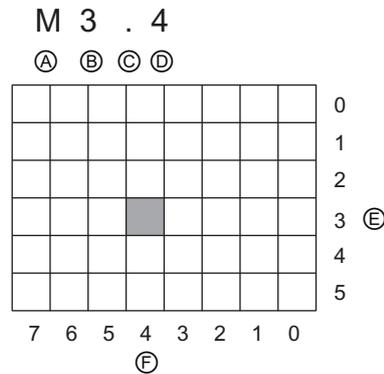
Durch Verweise auf Speicherbereiche für Eingänge (E) oder Ausgänge (A), z. B. E0.3 oder A1.7, erfolgt ein Zugriff auf das Prozessbild. Für den direkten Zugriff auf den physischen Eingang oder Ausgang fügen Sie den Verweis ":P" (z. B. E0.3:P, A1.7:P oder "Stop:P") als Anhang hinzu.

Durch Forcen wird lediglich ein Wert in einen Eingang (E) oder Ausgang (A) 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 105).

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

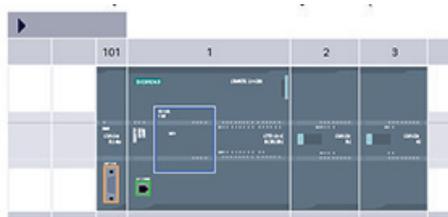
¹ Für den direkten Zugriff auf die physischen Eingänge und Ausgänge (oder um diese zu forcen) hängen Sie den Code ":P" an die Adresse oder Variable an (z. B. E0.3:P, A1.7:P oder "Stop:P").

Jeder Speicherplatz hat eine eindeutige Adresse. Anhand dieser Adresse kann Ihr Anwenderprogramm auf die Informationen an diesem Speicherplatz zugreifen. Das Bild zeigt ein Beispiel für den Zugriff auf ein Bit (Adressierung im Byte.Bit-Format). In diesem 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.



- 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

Adressen für die E/A konfigurieren



Baugruppe	Steckp.	E-Adresse	A-Adresse	Typ	Bestel
	103				
	102				
RS485_1	101			CM 1241 (RS485)	6ES7
PLC_1	1			CPU 1214C DDC/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 (f)	
HSC_2	1.17			Schneller Zähler (f)	
HSC_3	1.18			Schneller Zähler (f)	
HSC_4	1.19			Schneller Zähler (f)	
HSC_5	1.20			Schneller Zähler (f)	
HSC_6	1.21			Schneller Zähler (f)	
Pulse_1	1.32			Impulsgenerator (f)	
Pulse_2	1.33			Impulsgenerator (f)	
PROFINET...	X1			PROFINET-Schnitts.	
DIB x DC24V..	2	8		SM 1221 DIB x DC	6ES7

Wenn Sie eine CPU und E/A-Module in Ihren Konfigurationsbildschirm einfügen, werden E- und A-Adressen automatisch zugewiesen.

Sie können die voreingestellte Adressierung ändern, indem Sie im Konfigurationsbildschirm das Adressfeld auswählen und neue Zahlen eingeben. Digitale Eingänge und Ausgänge werden in vollständigen Bytes (je 8 Bit) zugewiesen, unabhängig davon, ob das Modul alle Ein- bzw. Ausgänge nutzt oder nicht. Analoge Eingänge und Ausgänge werden in Gruppen zu je 2 Ein- bzw. Ausgängen zugewiesen (4 Byte). In diesem Beispiel können Sie die Adresse von DE16 in 2..3 (statt 8..9) ändern. Das Werkzeug unterstützt Sie, indem Adressbereiche, die die falsche Größe haben oder mit anderen Adressen in Konflikt stehen, geändert werden.

Das Bild zeigt ein Beispiel für eine CPU 1214C mit zwei SMs.

4.4 Ausführung des Anwenderprogramms

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

- Ein Organisationsbaustein (OB) ist ein Codebaustein, der normalerweise den Hauptteil der Programmlogik enthält. Der OB reagiert auf ein bestimmtes Ereignis in der CPU und kann die Ausführung des Anwenderprogramms unterbrechen. Der Standardbaustein für die zyklische Ausführung des Anwenderprogramms (OB 1) stellt die Grundstruktur für Ihr Anwenderprogramm bereit und ist der einzige Codebaustein, der für das Anwenderprogramm erforderlich ist. Die anderen OBs führen bestimmte Funktionen aus, z. B. beim Anlaufen, für die Interrupt- und Fehlerbearbeitung oder für die Ausführung eines bestimmten Programmcodes in bestimmten Abständen.
- Ein Funktionsbaustein (FB) ist ein Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Der aufrufende Baustein übergibt dem FB die Parameter und definiert einen bestimmten Datenbaustein (DB), der die Daten für diesen Aufruf oder diese Instanz des FBs speichert. Wird der Instanz-DB geändert, so kann ein allgemeiner FB den Betrieb einer Gerätegruppe steuern. Beispielsweise kann ein FB mehrere Pumpen oder Ventile mit unterschiedlichen Instanz-DBs steuern, die die speziellen Betriebsparameter jeder Pumpe bzw. jedes Ventils enthalten. Der Instanz-DB speichert die Werte des FB zwischen verschiedenen oder aufeinander folgenden Aufrufen des FB, z. B. um asynchrone Kommunikation zu unterstützen.
- Eine Funktion (FC) ist eine Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Einer FC ist kein Instanz-DB zugeordnet. Der aufrufende Baustein übergibt der FC die Parameter. Die Ausgangswerte der FC müssen in eine Speicheradresse oder in einen globalen DB geschrieben werden, wenn andere Komponenten des Anwenderprogramms diese Werte benötigen.

Die Größe des Anwenderprogramms, der Daten und der Konfiguration ist durch den verfügbaren Ladespeicher und den Arbeitsspeicher in der CPU begrenzt. Die Anzahl unterstützter Bausteine ist nicht begrenzt, einziger Grenzwert ist der freie Speicherplatz.

OBs zur Ereignisbearbeitung verwenden

Die Bearbeitung des CPU-Zyklus wird durch Ereignisse gesteuert. Das Standardereignis ist ein Programmzyklusereignis, das die Ausführung des Programmzyklus-OBs startet. (Die Verwendung eines Programmzyklus-OBs in Ihrem Programm ist nicht obligatorisch. Wenn Sie jedoch nicht über einen Programmzyklus-OB verfügen, finden keine normalen E/A-Aktualisierungen statt. In diesem Fall müssen Sie zum Lesen der bzw. Schreiben in die E/A das Prozessbild verwenden.) Andere Ereignisse können bei Bedarf aktiviert werden. Einige Ereignisse, z. B. zyklische Ereignisse, werden während der Konfiguration aktiviert. Andere Ereignisse, z. B. Zeitverzögerungereignisse, werden zur Laufzeit aktiviert. Ein aktiviertes Ereignis ist mit einem zugehörigen OB verknüpft. (Die Programmzyklus- und Anlaufereignisse können jeweils mit mehreren OBs verknüpft werden.) Tritt ein Ereignis ein, so wird die Ausführung des entsprechenden Unterprogramms für die Ereignisbearbeitung, d. h. des zugewiesenen OBs und anderer, aus dem OB aufgerufener Funktionen, gestartet. In welcher Reihenfolge diese Unterprogramme bearbeitet werden, kann durch das Einrichten von Prioritäten, Prioritätsgruppen und Warteschlangen festgelegt werden.

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. Jedem Ereignis ist eine Priorität zugeordnet, und die Ereignisprioritäten sind in Prioritätsgruppen eingeteilt. Dies zeigt die folgende Tabelle. 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. Nachdem die Ausführung eines OBs gestartet ist, kann seine Bearbeitung nicht durch ein weiteres Ereignis aus der gleichen oder einer niedrigeren Prioritätsgruppe unterbrochen werden. Solche Ereignisse werden zur späteren Bearbeitung in eine Warteschlange gestellt, sodass die CPU die Ausführung des aktuellen OBs abschließen kann.

Ein OB innerhalb einer Prioritätsgruppe kann keinen OB aus derselben Prioritätsgruppe unterbrechen. Ein Ereignis aus Prioritätsgruppe 2 jedoch unterbricht die Ausführung eines OBs aus Prioritätsgruppe 1, und ein Ereignis aus Prioritätsgruppe 3 unterbricht die Ausführung von OBs aus beiden niedrigeren Prioritätsgruppen, 1 und 2.

Ereignis (OB)	Anzahl	OB-Nummer	Warteschlangenlänge	Prioritätsgruppe	Prioritätsklasse
Programmzyklus	1 Programmzyklusereignis Mehrere OBs zulässig	1 (Standard) 200 oder höher	1	1	1
Anlauf	1 Anlaufereignis ^{1, 2} Mehrere OBs zulässig	100 (Standard) 200 oder höher	1		1
Verzögerung	Bis zu 4 Verzögerungsereignisse ³ 1 OB pro Ereignis	200 oder größer	8	2	3
Zyklisch	Bis zu 4 Verzögerungsereignisse ³ 1 OB pro Ereignis	200 oder größer	8		4
Flanken	16 steigende Flanken 16 fallende Flanken 1 OB pro Ereignis	200 oder größer	32		5
HSC	6 Ereignisse AW = VW 6 Richtungswechsel 6 externe Reset-Ereignisse 1 OB pro Ereignis	200 oder größer	16		6
Diagnosefehler	1 Ereignis (nur OB 82)	nur 82	8	3	9
Zeitfehler	1 Zeitfehler-Ereignis 1 MaxZykluszeit-Ereignis (Nur OB 80) 1 2xMaxZyklus	nur 80	8		26
					27

¹ Sonderfall für das Anlaufereignis: 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 Sonderfall für das Anlaufereignis: Das Anlaufereignis kann nur durch einen Diagnosefehler (in Verbindung mit OB 82) unterbrochen werden. Alle anderen Ereignisse werden für die spätere Bearbeitung nach dem Anlaufereignis in die Warteschlange gestellt.

³ Die CPU stellt insgesamt 4 Zeitergebnisse 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.

Ein OB aus einer höheren Prioritätsgruppe unterbricht die Ausführung eines OBs aus einer niedrigeren Prioritätsgruppe. Beispielsweise unterbricht ein OB aus Prioritätsgruppe 2 (z. B. ein Weckalarm-OB) einen Programmzyklus-OB (Prioritätsgruppe 1), und ein OB 80 (Prioritätsgruppe 3) unterbricht alle OBs aus Prioritätsgruppe 1 oder 2. Die OBs aus derselben Prioritätsgruppe unterbrechen einander nicht. Alle Ereignisse während der Bearbeitung eines OBs werden von der CPU gespeichert. Nach abgeschlossener Bearbeitung eines OBs führt die CPU die OBs in der Warteschlange aus, wobei sie sich an der relativen Prioritätsklasse innerhalb der Prioritätsgruppe orientiert und Ereignisse mit höherer Prioritätsklasse zuerst bearbeitet. Innerhalb einer Prioritätsgruppe führt jedoch die CPU einen OB immer vollständig aus, bevor sie die Ausführung des nächsten OB in dieser Prioritätsgruppe startet. Nach Bearbeitung aller Ereignisse in der unterbrechenden Prioritätsgruppe kehrt die CPU zum unterbrochenen OB der niedrigeren Prioritätsgruppe zurück und setzt dessen Ausführung an der Stelle fort, an der sie unterbrochen wurde.

Wenn die CPU einen Fehler in Prioritätsgruppe 3 (z. B. ein Zeitfehler-Ereignis) feststellt, unterbricht der Zeitfehler-OB die Verarbeitung sowohl der Prioritätsgruppe 1 (mit z. B. einem Programmzyklus-OB) als auch der Prioritätsgruppe 2 (mit z. B. einem Zyklus-OB). Die CPU führt dann den Zeitfehler-OB aus und kehrt anschließend zur Ausführung des zuvor unterbrochenen OBs zurück, entweder in Prioritätsgruppe 2 (falls dort unterbrochen) oder in Prioritätsgruppe 1.

4.5 Die Einrichtung des Zugriffsschutzes für CPU oder Codebausteine ist einfach

Die CPU bietet 3 Sicherheitsstufen, um den Zugriff auf bestimmte 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.

Zum Einrichten des Passworts gehen Sie folgendermaßen vor:

1. Wählen Sie in der "Gerätekonfiguration" die CPU aus.
2. Wählen Sie im Inspektorfenster das Register "Eigenschaften" aus.
3. Wählen Sie die Eigenschaft "Schutz", um die Schutzstufe auszuwählen und ein Passwort einzugeben.

Die Groß- und Kleinschreibung des Passwortes ist zu beachten.

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.

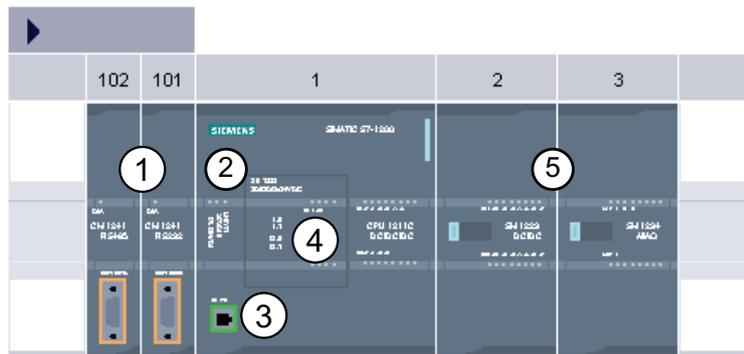
Die Kommunikation zwischen CPUs (über die Kommunikationsfunktionen in den Codebausteinen) wird durch die Schutzstufe der CPU nicht eingeschränkt. Auch die HMI-Funktionalität bleibt uneingeschränkt erhalten.

Schutzstufe	Zugangsbeschränkungen
Kein Schutz	Ungehinderter Zugang ohne Passwortschutz.
Schreibschutz	Nur Lesezugriff auf die CPU, HMI-Zugang und 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.

Programmiergrundlagen leicht gemacht

5.1 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 Projektnamen auf "Neues Gerät hinzufügen".



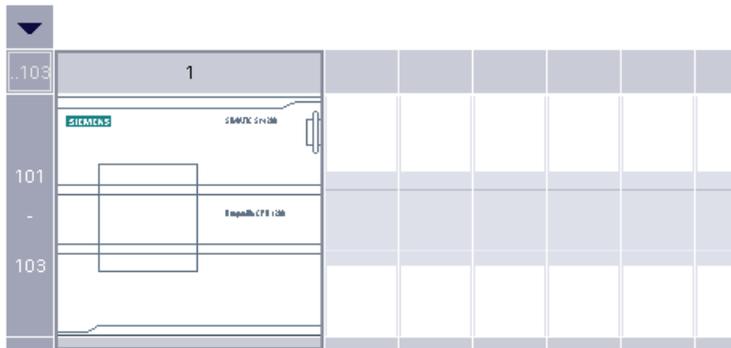
Das Laden einer vorhandenen Hardwarekonfiguration ist einfach



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 Basic legt dann automatisch eine nicht spezifizierte CPU an.)

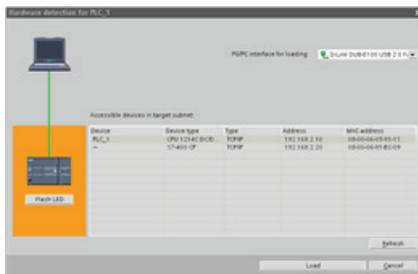
Wählen Sie im Programmiereditor im Menü "Online" den Befehl "Hardwareerkennung".

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.

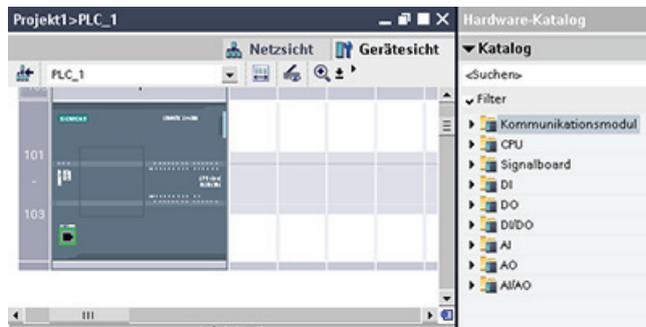
Nachdem Sie im Online-Dialog die CPU ausgewählt haben, lädt STEP 7 Basic die Hardwarekonfiguration einschließlich möglicher Module (SM, SB oder CM) aus der CPU. Sie können dann die Parameter für die CPU und die Module (Seite 50) konfigurieren.



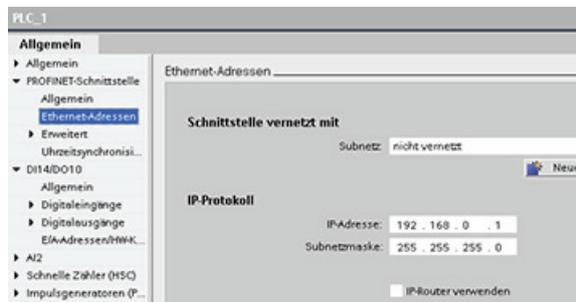
CPU zur Konfiguration hinzufügen



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 50).

Hinweis

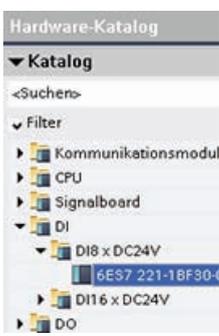
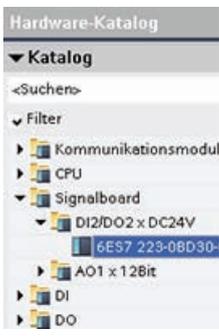
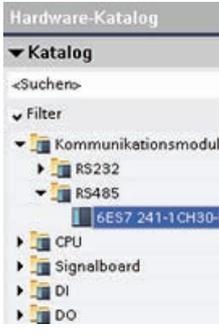
Die CPU hat keine vorkonfigurierte IP-Adresse. Sie müssen der CPU daher bei der Gerätekonfiguration manuell eine IP-Adresse zuweisen. Ist Ihre CPU an einen Router im Netzwerk angeschlossen, so muss auch die IP-Adresse des Routers eingegeben werden.

Gerät zur Konfiguration hinzufügen

Im Hardwarekatalog können Sie Module zur CPU hinzufügen. Es gibt drei Arten von Modulen:

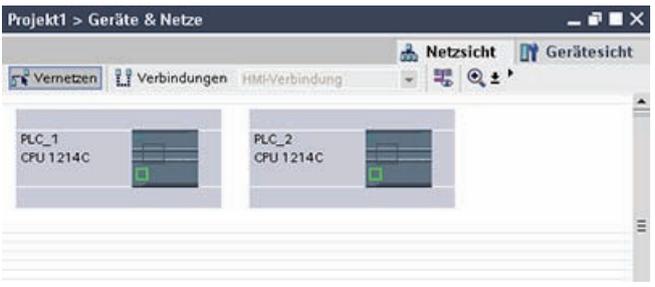
- Signalboards (SB) bieten eine begrenzte Zahl von zusätzlichen Ein-/Ausgängen für die CPU. Das SB wird auf der Vorderseite der CPU gesteckt.
- Signalmodule (SM) für zusätzliche digitale oder analoge Ein- und Ausgänge. Diese Module werden an der rechten Seite der CPU angeschlossen.
- Kommunikationsmodule (CM) stellen eine zusätzliche Kommunikationsschnittstelle (RS232 oder RS485) für die CPU bereit. 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.

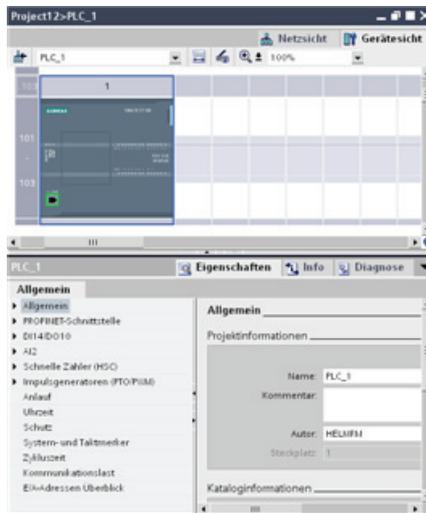
Modul	Modul auswählen	Modul einsetzen	Ergebnis
SM			
SB			
CM			

Netzwerkverbindung konfigurieren

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.

Netzsicht der Gerätekonfiguration	Beschreibung
	<p>Wählen Sie die "Netzsicht" 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>

5.1.1 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 Aus-zu-Ein-Wechsel
- 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

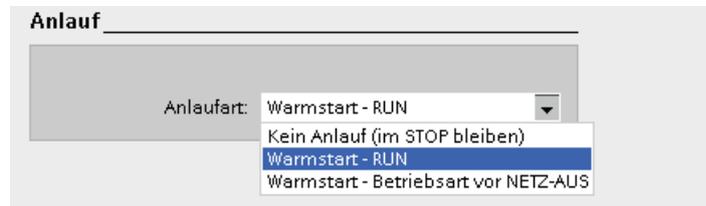
Konfigurieren des CPU-Betriebs für den Wechsel von STOP nach RUN

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 Operationen im Anlauf-OB den Wert Null und nicht den aktuellen Wert des physischen Eingangs.) Um den aktuellen Zustand eines physischen Eingangs im Betriebszustand STARTUP zu lesen, müssen Sie den Eingang direkt lesen. 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.

- Interrupts 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: Wechsel entweder in den Betriebszustand STOP oder RUN oder in den letzten 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 jedem Laden immer einen Kaltstart 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 Kaltstart durch. Neben dem Löschen der Eingänge, Initialisieren der Ausgänge und Löschen des nicht remanenten Speichers werden beim Kaltstart auch die remanenten Speicherbereiche gelöscht.

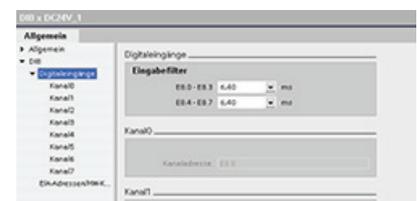
Nach dem Kaltstart, der auf einen Ladevorgang folgt, wird bei jedem nachfolgenden Wechsel von STOP in RUN ein Warmstart durchgeführt (dabei wird der remanente Speicher nicht gelöscht).

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".

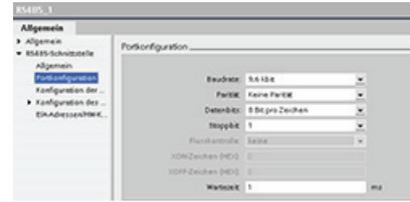
Signalmodul (SM) und Signalboard (SB)

- Digitale E/A: Konfigurieren der einzelnen Eingänge, z.B. für die Flankenerkennung und den Impulsabgriff (Eingang bleibt nach einem Impuls eingeschaltet) 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
- Diagnoseadressen: Konfigurieren der Startadresse für die Ein- und Ausgänge des Moduls



Kommunikationsmodul (CM)

- 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.

5.1.2 Konfigurieren der IP-Adresse der CPU

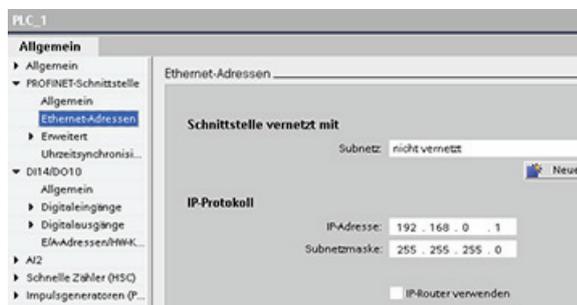
Da die CPU keine vorkonfigurierte IP-Adresse besitzt, müssen Sie manuell eine IP-Adresse zuweisen. Sie konfigurieren die IP-Adresse und die übrigen Parameter für die PROFINET-Schnittstelle bei der Konfiguration der Eigenschaften für die CPU.

- 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 Ihres Computers mit der IP-Adresse des Programmiergeräts kompatibel ist.

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

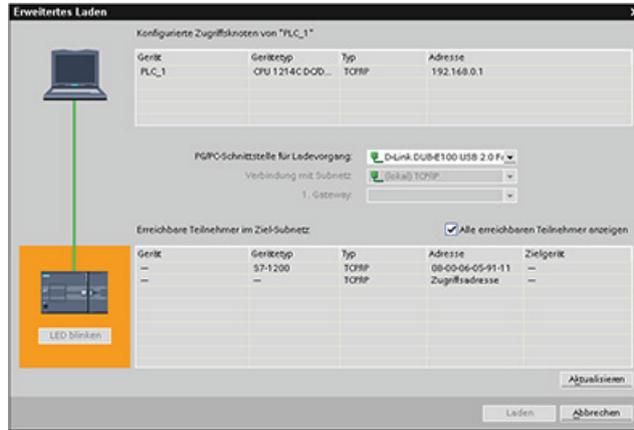
1. Klicken Sie in der Projektnavigation mit der rechten Maustaste auf "Online-Zugänge", um das Kontextmenü aufzurufen.
2. Wählen Sie "Eigenschaften".



Im Dialog werden die Einstellungen für das Programmiergerät angezeigt.

Die IP-Adresse 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.

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-Adresse für die CPU und ggf. den Router werden beim Laden des Projekts eingerichtet.

5.2 Einfaches Entwerfen Ihres Anwenderprogramms

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

Ein OB ist ein Codebaustein, mit dem Sie das Anwenderprogramm für Ihre Anwendung strukturieren bzw. organisieren. Für viele Anwendungen gibt es einen periodisch durchlaufenden OB, wie z. B. den Programmzyklus-OB "OB 1", der die Programmlogik enthält. Neben den Programmzyklus-OBs stellt die CPU weitere OBs für spezifische Funktionen bereit, z. B. für Aufgaben beim Anlauf, die Behandlung von Interrupts und Fehlern oder die Ausführung eines bestimmten Programmcodes in bestimmten Zeitabständen. Jeder OB reagiert auf ein bestimmtes Ereignis in der CPU und kann, orientiert an vordefinierten Prioritätsgruppen und -klassen, die Ausführung des Anwenderprogramms unterbrechen.

Ein Funktionsbaustein (FB) ist ein Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Der aufrufende Baustein übergibt dem FB die Parameter und definiert einen bestimmten Datenbaustein (DB), der die Daten für diesen Aufruf oder diese Instanz des FBs speichert. Wird der Instanz-DB geändert, so kann ein allgemeiner FB den Betrieb einer Gerätegruppe steuern. Beispielsweise kann ein FB mehrere Pumpen oder Ventile mit unterschiedlichen Instanz-DBs, die die speziellen Betriebsparameter jeder Pumpe bzw. jedes Ventils enthalten, steuern. Der Instanz-DB speichert die Werte des FBs zwischen verschiedenen oder aufeinander folgenden Aufrufen des FBs, z. B. um die asynchrone Kommunikation zu unterstützen.

Eine Funktion (FC) ist ein Unterprogramm, das ausgeführt wird, wenn es aus einem anderen Codebaustein (OB, FB oder FC) heraus aufgerufen wird. Einer FC ist kein Instanz-DB zugeordnet. Der aufrufende Baustein übergibt der FC die Parameter. Die Ausgangswerte der FC müssen in eine Speicheradresse oder in einen globalen DB geschrieben werden.

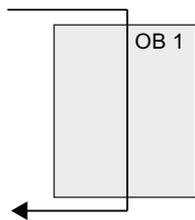
Art der Struktur für das Anwenderprogramm wählen

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

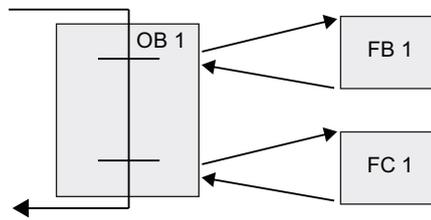
Ein lineares Programm führt alle Anweisungen für Ihre Automatisierungsaufgaben nacheinander aus. Typischerweise werden bei einem linearen Programm alle Programmweisungen in einem einzigen Programmzyklus-OB (OB 1) abgelegt, anhand dessen das Programm zyklisch abgearbeitet wird.

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:

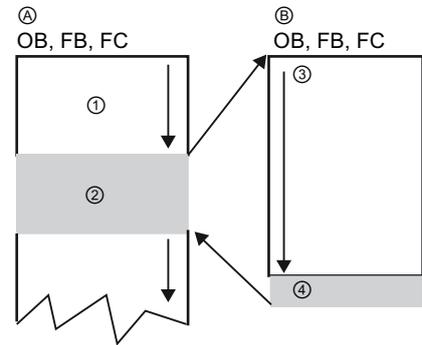


Modulare Struktur:



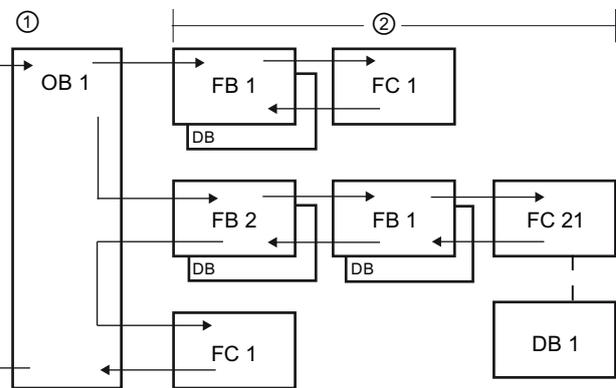
Modulare Codebausteine erstellen Sie durch den Entwurf von FBs und FCs für die Ausführung allgemeiner Aufgaben. Anschließend strukturieren Sie Ihr Programm, indem andere Codebausteine wiederum die wiederverwendbaren Module aufrufen. Der aufrufende Baustein gibt gerätespezifische Parameter an den aufgerufenen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

- A Aufrufender Baustein
- B Aufgerufener (oder unterbrechender) Baustein
- ① Programmausführung
- ② Anweisung oder Ereignis, das 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.

- ① Zyklusbeginn
 - ② Schachtelungstiefe
- In diesem Beispiel umfasst die Schachtelungstiefe 4 Ebenen: den Programmzyklus-OB plus 3 Ebenen mit Aufrufen von Codebausteinen.



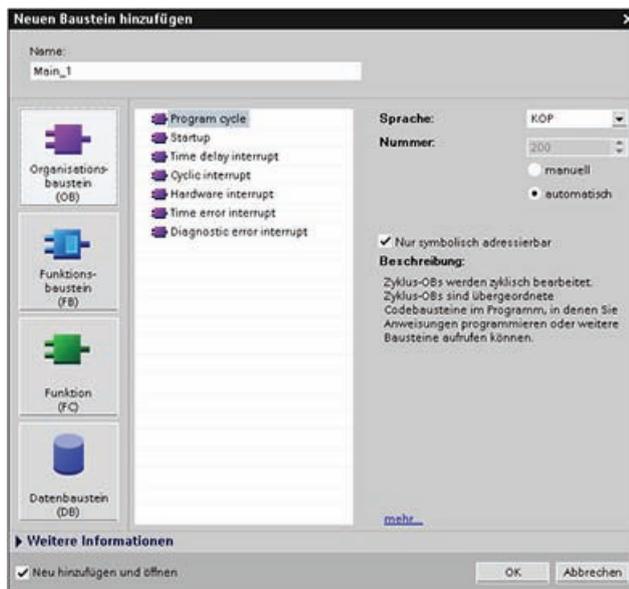
Durch das Anlegen allgemeiner Codebausteine, die im Anwenderprogramm mehrfach genutzt werden können, lassen sich Entwurf und Implementierung des Anwenderprogramms vereinfachen.

- 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.

5.2.1 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 den Programmzyklus-OB 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 Interrupt- und Fehlerbearbeitung oder für die Ausführung eines bestimmten Programmcodes in bestimmten Abständen.

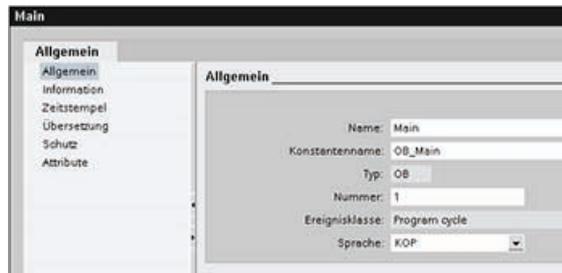
Einen neuen OB für Ihr Anwenderprogramm legen Sie im Dialog "Neuen Baustein hinzufügen" an.

Die CPU ermittelt die Reihenfolge für die Bearbeitung von Interruptereignissen anhand der den einzelnen OBs zugeordneten Prioritäten (Seite 40).

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 Hauptprogrammzyklus-OB (standardmäßig: OB 1). Beispiel: Nach dem Ende des ersten Programmzyklus-OB (OB 1) führt die CPU den zweiten Programmzyklus-OB aus (z. B. OB 200).

Konfigurieren der Funktionsweise eines OBs



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

5.2.2 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 Operation 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.

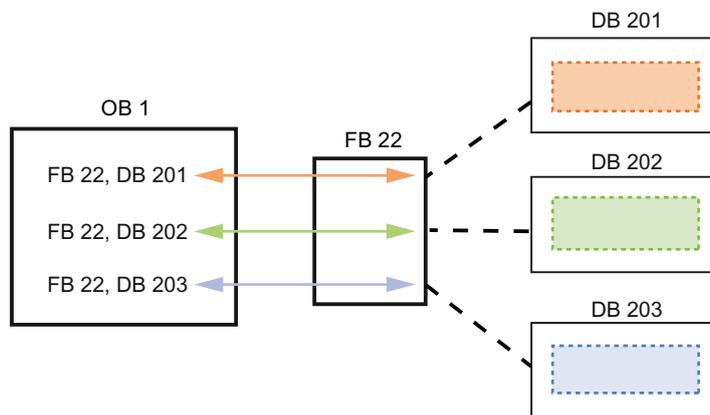
Ein Funktionsbaustein (FB) ist eine Art Unterprogramm mit Speicher. Ein Funktionsbaustein (FB) ist ein Codebaustein, dessen Aufrufe mit Bausteinparametern programmiert werden können. Ein FB speichert die Eingangs- (IN), Ausgangs- (OUT) und Durchgangparameter (IN_OUT) in einem variablen Speicherbereich, 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. Der Instanz-DB speichert diese Werte nach dem Beenden des FBs.

Sie können den Parametern in der FB-Schnittstelle Anfangswerte 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.



5.2.3 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. Ein DB kann so konfiguriert werden, dass er nur gelesen werden kann.

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.

Neuen Codebaustein anlegen



1. Öffnen Sie den Ordner "Programmbausteine".
2. Doppelklicken Sie auf "Neuen Baustein hinzufügen".
3. Wählen Sie im Dialog "Neuen Baustein hinzufügen" die Schaltfläche "Funktion (FC)".
4. Wählen Sie die Programmiersprache für die FC 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.



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 Aufrufoperation zu erstellen.

Hinweis

Mit dem Anwenderprogramm können Sie keinen OB aufrufen, da OBs ereignisgesteuert (Seite 40) sind. Die Ausführung eines OBs wird von der CPU in Reaktion auf den Empfang eines Ereignisses gestartet.

5.3 Problemloser Einsatz der leistungsstarken Programmiersprachen

Als Programmiersprache können Sie entweder den Kontaktplan (KOP) oder den Funktionsplan (FUP) wählen.

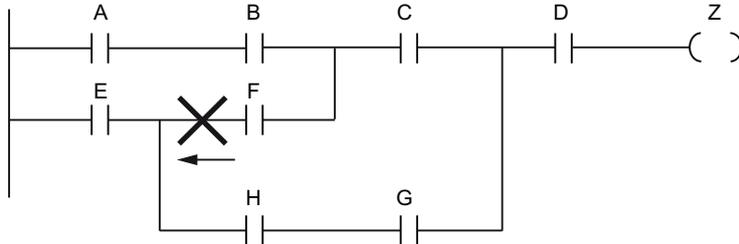
KOP ist eine grafische Programmiersprache. Die Darstellung beruht auf Schaltplänen. Um Verknüpfungen für komplexe Operationen anzulegen, können Sie Verzweigungen für parallele Kreise einfügen. Parallele Verzweigungen sind nach unten geöffnet oder direkt mit der Stromschiene verbunden. Sie beenden die Verzweigungen nach unten. KOP bietet außerdem Box-Operationen für eine Vielzahl von Funktionen wie arithmetische Operationen, Zeiten, Zähler und Übertragen.



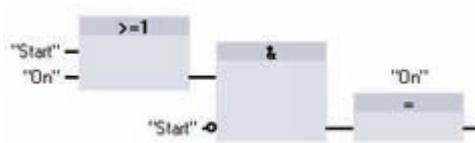
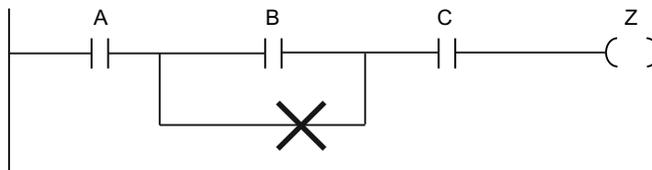
Die Elemente eines Schaltplans, wie Öffnerkontakte, Schließerkontakte und Spulen, werden zu Netzwerken verknüpft.

Beim Anlegen eines KOP-Netzwerks sind die folgenden Regeln zu beachten:

- Jedes KOP-Netzwerk muss mit einer Spule oder einer Box abgeschlossen werden. Schließen Sie ein Netzwerk nicht mit einer Vergleichsoperation oder einer Flankenerkennung (fallende oder steigende Flanke) ab.
- Sie können keine Verzweigung anlegen, die zu einem Signalfluss in die Gegenrichtung führen könnte.



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



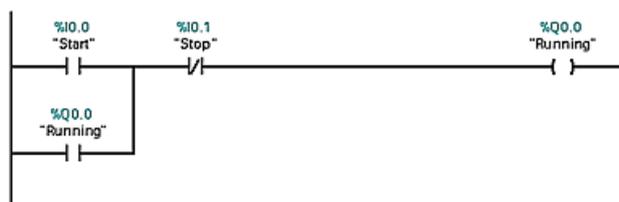
Ebenso wie KOP ist auch FUP eine grafische Programmiersprache. Die Darstellung der Verknüpfungslogik beruht auf den grafischen Symbolen, die in der booleschen Algebra üblich sind.

Arithmetische Funktionen und andere komplexe Funktionen können direkt in Verbindung mit den Logikboxen dargestellt werden. Um Verknüpfungen für komplexe Operationen anzulegen, fügen Sie parallele Verzweigungen zwischen den Boxen ein.

5.3.1 Bereitstellung der erwartbaren grundlegenden Operationen

Bitverknüpfungsoperationen

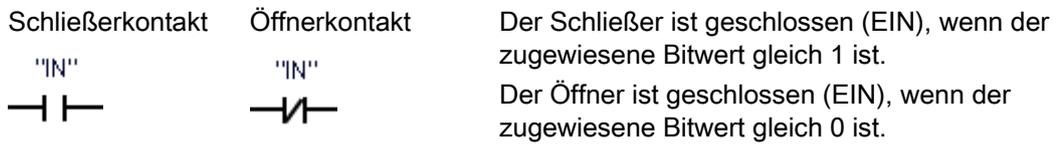
Die Grundlage von Bitverknüpfungsoperationen sind Kontakte und Spulen. Kontakte lesen den Status eines Bits aus, während Spulen den Status der Operation in ein Bit schreiben.



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.

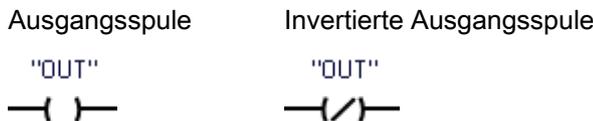
Wenn Sie eine Spule mit derselben Adresse in mehreren Programmsegmenten verwenden, so hängt vom Ergebnis der letzten Berechnung im Anwenderprogramm der Zustand des Werts für diese Adresse ab.



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 Operation 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 PLC0 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 Operationen einfügen.

Kontaktinverter
NOT (KOP)



UND-Box mit einem
invertierten Logikeingang
(FUP)



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

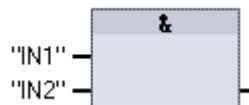


Der KOP-Kontakt NOT invertiert den logischen Zustand des Signalflusseingangs.

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

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 Logikinverterier für diesen Box-Anschluss zu erstellen.

UND-Box (FUP)



ODER-Box (FUP)



XOR-Box (FUP)



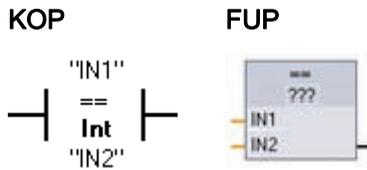
- Damit der Ausgang WAHR ist, müssen alle Eingänge einer UND-Box WAHR sein
- Damit der Ausgang WAHR ist, muss ein beliebiger Eingang einer ODER-Box WAHR sein
- Damit der Ausgang WAHR ist, muss eine ungerade Anzahl der Eingänge einer XOR-Box WAHR sein

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-Operation werden die Eingangszustände auf die binäre Box-Verknüpfung geschaltet und dann der Box-Ausgang, sofern zutreffend, auf Wahr gesetzt.

Vergleichsoperationen

Mit den Vergleichsoperationen können Sie zwei Werte des gleichen Datentyps vergleichen. Ist das Ergebnis des Vergleichs WAHR, dann ist der Kontakt aktiviert (KOP) oder der Box-Ausgang ist WAHR (FUP).

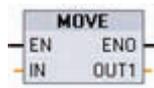


Nach dem Anklicken der Operation im Programmiereditor können Sie den Vergleichstyp in den Klapplisten auswählen.

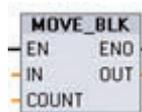
Beziehungstyp	Der Vergleich ist wahr, wenn:
==	IN1 gleich IN2 ist
<>	IN1 nicht gleich IN2 ist
>=	IN1 größer oder gleich IN2 ist
<=	IN1 kleiner oder gleich IN2 ist
>	IN1 größer als IN2 ist
<	IN1 kleiner als IN2 ist

Operationen Wert kopieren und Bereich kopieren

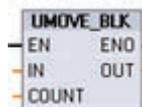
Mit den Operationen zum Verschieben kopieren Sie Datenelemente in eine neue Adresse im Speicher und wandeln die Daten von einem Datentyp in einen anderen um. Die Quelldaten werden dadurch nicht verändert.



MOVE kopiert ein unter einer bestimmten Adresse gespeichertes Datenelement in eine neue Adresse.



MOVE_BLK (Bereich kopieren) kopiert einen Bereich mit Datenelementen in eine neue Adresse.



UMOVE_BLK (Bereich ununterbrechbar kopieren) kopiert einen Bereich mit Datenelementen in eine neue Adresse

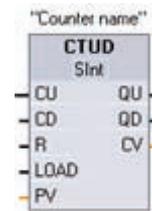
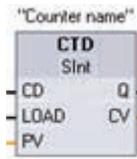
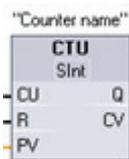
- Mit der Operation MOVE wird ein einzelnes Datenelement von der mit Parameter IN angegebenen Quelladresse in die mit Parameter OUT angegebene Zieladresse kopiert.
- Die Operationen MOVE_BLK und UMOVE_BLK verfügen zusätzlich über einen Parameter COUNT. Mit COUNT wird festgelegt, wie viele Datenelemente 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-Variablen-tabelle zugewiesen ist.

Zähler

Mit den Zähleroperationen können Sie programminterne Ereignisse und externe Prozessereignisse zählen. Jeder Zähler nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten des Zählers zu speichern. Sie weisen den Datenbaustein zu, wenn Sie die Zähloperation im Editor einfügen. Diese Operationen nutzen Softwarezähler, deren maximale Zählgeschwindigkeit durch die Ausführungsrate des OBs, in den sie eingefügt wurden, begrenzt ist.

CTU zählt vorwärts.

CTD zählt rückwärts. CTUD zählt vorwärts und rückwärts.



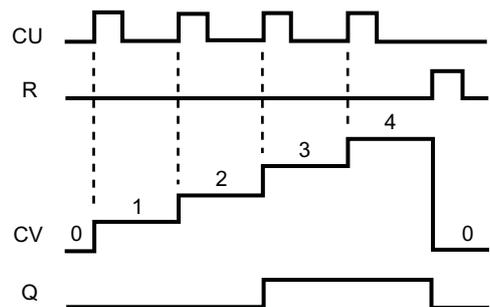
Wählen Sie den Datentyp für den Zählwert aus der Klappliste unterhalb des Zählernamens aus.

Die Anzahl der Zähler, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Zähler benötigen den folgenden Speicherplatz:

- Bei den Datentypen SInt oder USInt benötigt die Zähloperation 3 Byte.
- Bei den Datentypen Int oder UInt benötigt die Zähloperation 6 Byte.
- Bei den Datentypen DInt oder UDInt benötigt die Zähloperation 3 Byte.

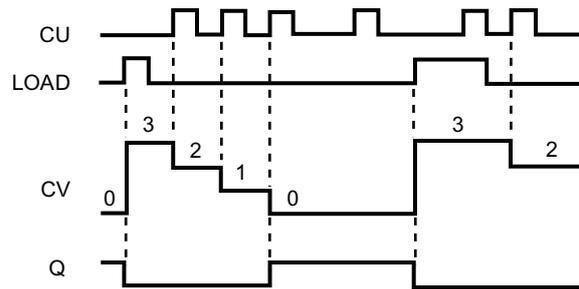
CTU zählt um 1 vorwärts, wenn der Wert des Parameters CU von 0 nach 1 wechselt. Die Abbildung zeigt ein CTU-Zeitdiagramm mit einem vorzeichenlosen ganzzahligen Zählwert (wobei PV = 3).

- 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 A = 1.
- Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt.



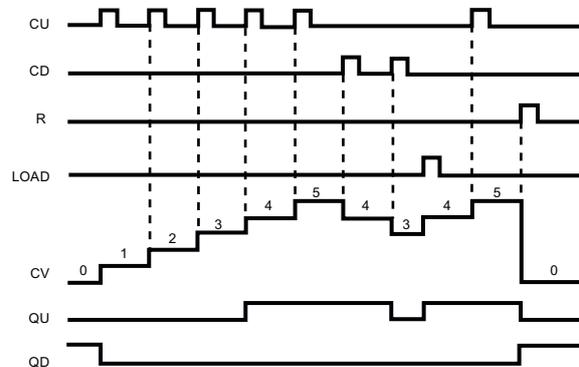
CTD zählt um 1 rückwärts, wenn der Wert des Parameters CD von 0 nach 1 wechselt. Die Abbildung zeigt ein CTD-Zeitdiagramm mit einem vorzeichenlosen ganzzahligen Zählwert und PV = 3.

- Ist der Wert des Parameters CV (aktueller Zählwert) kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang A = 1.
- Wechselt der Wert von Parameter LOAD von 0 nach 1, wird der Wert an Parameter PV (voreingestellter Wert) als neuer aktueller Zählwert CV in den Zähler geladen.



CTUD zählt um 1 vorwärts oder rückwärts, wenn der Eingang CU (Vorwärtszählen) oder CD (Rückwärtszählen) von 0 nach 1 wechselt. Die folgende Abbildung zeigt ein Zeitdiagramm für CTUD mit einem vorzeichenlosen ganzzahligen Zählwert und PV = 4.

- 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 AU = 1.
- Ist der Wert des Parameters CV kleiner oder gleich 0, so lautet der Parameter für den Zählerausgang AD = 1.
- Wechselt der Wert von Parameter LOAD von 0 nach 1, so wird der Wert an Parameter PV (voreingestellter Wert) als neuer aktueller Zählwert CV in den Zähler geladen. Wenn der Wert des Rücksetzparameters R von 0 nach 1 wechselt, wird der aktuelle Zählwert auf 0 zurückgesetzt.



Zeiten

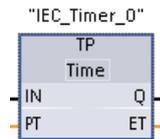
Mit den Zeitoperationen können Sie programmierte Zeitverzögerungen einrichten:

- TP: Der Impulszeitgeber erzeugt einen Impuls mit einer voreingestellten Dauer.
- TON: Ausgang Q der Einschaltverzögerung wird nach einer voreingestellten Zeit auf EIN gesetzt.
- TOF: Ausgang Q der Ausschaltverzögerung wird nach einer voreingestellten Zeit auf AUS gesetzt.
- TONR: Der Ausgang der Operation Zeit akkumulieren wird nach einer voreingestellten Zeit auf EIN gesetzt. Die abgelaufene Zeit wird über mehrere Zeitintervalle kumuliert, bis Eingang R zum Zurücksetzen der abgelaufenen Zeit angestoßen wird.
- RT: Setzt eine Zeit zurück, indem die Zeitdaten im angegebenen Instanz-Datenbaustein der Zeit gelöscht werden.

TP, TON und TOF haben die gleichen Ein- und Ausgangsparameter.

Die Zeit TONR verfügt über den zusätzlichen Parameter R für den Rücksetzeingang.

Die Operation RT setzt die Zeitdaten der angegebenen Zeit zurück.

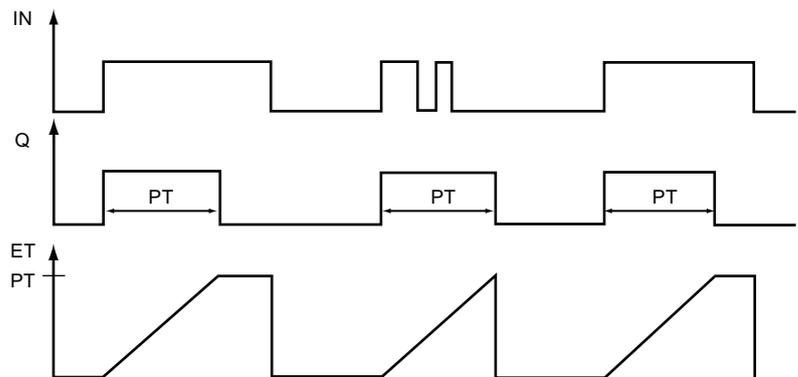


"Name der Zeit"
 ---[RT]---

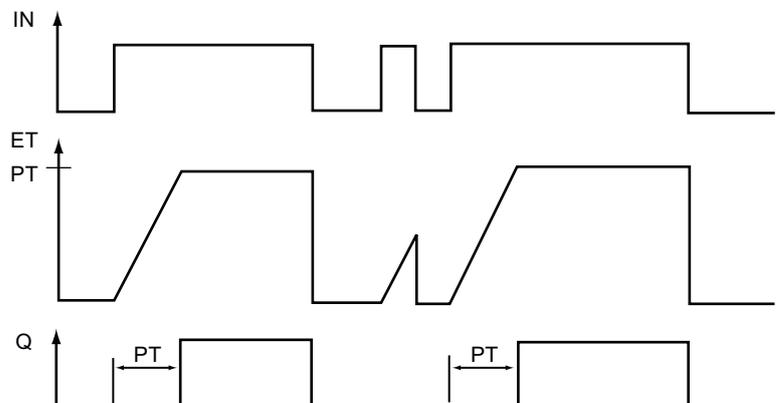
Die Anzahl der Zeiten, die Sie in Ihrem Anwenderprogramm verwenden können, ist lediglich durch den Speicherplatz in der CPU begrenzt. Jede Zeit nutzt die 16 Byte im Speicher:

Jede Zeit nutzt eine in einem Datenbaustein abgelegte Struktur, um die Daten der Zeit zu speichern. Sie weisen den Datenbaustein zu, wenn Sie die Zeitoperation im Editor einfügen. Wenn Sie Zeitoperationen in einem Funktionsbaustein platzieren, können Sie die Option "Multiinstanz-Datenbaustein" auswählen. Die Namen der Zeitstruktur können bei verschiedenen Datenstrukturen unterschiedlich sein, doch die Zeitdaten befinden sich in einem einzigen Datenbaustein, und es ist nicht für jede Zeit ein eigener Datenbaustein erforderlich. Dadurch verringert sich die Verarbeitungszeit und der für die Verwaltung der Zeiten erforderliche Datenspeicher. Zwischen den Datenstrukturen der Zeiten im gemeinsam genutzten Multiinstanz-Datenbaustein gibt es keine Wechselwirkungen.

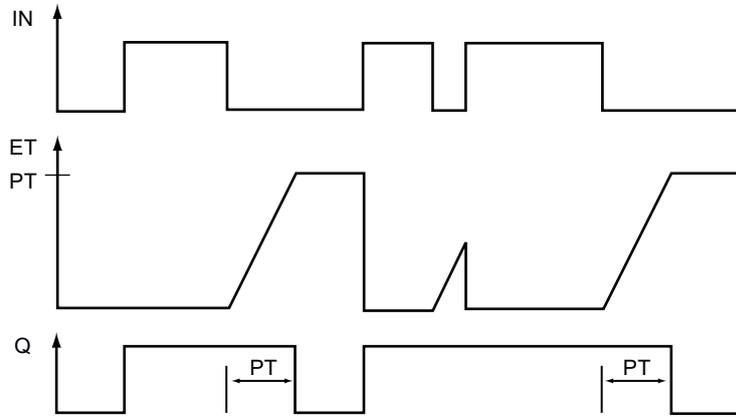
TP Zeit
 Zeitdiagramm
 Impuls



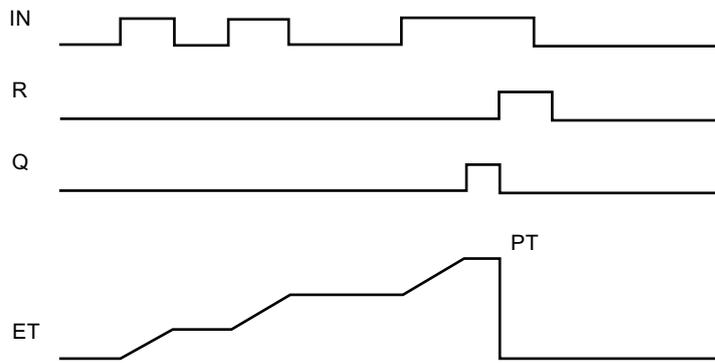
TON Zeit
 Zeitdiagramm
 Einschaltverzögerung



TOF Zeit
Zeitdiagramm
Ausschaltverzögerung



TONR Zeit
Zeitdiagramm Zeit
akkumulieren



S7-1200 bietet leistungsstarke Operationen

Neben den grundlegenden Operationen bietet die S7-1200 einen eindrucksvollen Befehlssatz, mit dem Sie einfache Lösungen für komplexe Steueranwendungen programmieren können. Die folgenden Operationen sind nur Beispiele für die Leistung, die in der S7-1200 steckt.

CTRL_PWM Operation: Die Operation CTRL_PWM für die Impulsdauermodulation (PWM) bietet Ihnen eine feste Zykluszeit mit variabler Einschaltdauer. Der PWM-Ausgang läuft nach dem Start kontinuierlich mit der angegebenen Frequenz (Zykluszeit). Die Impulsdauer wird nach Bedarf verändert.



Für weitere Informationen beachten Sie die Beschreibung der Impulsdauermodulation (Seite 92).

PID_Compact Operation: Der PID-Regler (Proportional/Integral/Differential) berechnet mittels PID-Algorithmus die Differenz zwischen einem zurückgemeldeten Wert und einem Sollwert und gibt das Ergebnis an Stellglieder wie z. B. Heizgeräte oder Frequenzumrichter aus, um den Sollwert aufrecht zu erhalten. Die Operation PID_Compact stellt einen PID-Regler mit selbsteinstellender Optimierung für den Automatik- und Handbetrieb zur Verfügung.



Führen Sie die Operation PID_Compact in einem konstanten Zeitintervall der Abtastzeit aus (nach Möglichkeit in einem Weckalarm-OB).

Die Operation PID_Compact misst das Zeitintervall zwischen zwei Aufrufen und wertet 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 im Baustein. Bei der Überwachung wird die aktuelle Messzeit zwischen zwei Aufrufen und der Mittelwert der für den Regler festgelegten Abtastzeit überwacht.

Modi	Beschreibung
Inaktiv	Nach dem ersten Laden des Anwenderprogramms bleibt der PID-Regler in der Betriebsart "Inaktiv". Führen Sie dann im Inbetriebnahmefenster eine "Selbsteinstellung während des ersten Starts" durch. Während des laufenden Betriebs wechselt der PID-Regler in die Betriebsart "Inaktiv", wenn ein Fehler auftritt oder wenn im Inbetriebnahmefenster auf das Symbol "Regler anhalten" geklickt wird.
Selbsteinstellung	Es wird in die Betriebsart "Selbsteinstellung während des ersten Starts" oder "Selbsteinstellung am Betriebspunkt" gewechselt, wenn die Funktion im Inbetriebnahmefenster aufgerufen wird.
Automatikbetrieb	Im Automatikbetrieb korrigiert die Operation PID_Compact den Regelkreis anhand festgelegter Parameter.
Handbetrieb	Im Handbetrieb des PID-Reglers kann die Stellvariable manuell verändert werden.

Bewegungssteuerungsoperationen: Die Bewegungssteuerungsoperationen nutzen einen zugehörigen Technologie-Datenbaustein und die dafür zugewiesene PTO (Impulsfolge) der CPU, um die Bewegung einer Achse zu steuern. Informationen über die Funktionsweise der Bewegungssteuerungsoperationen finden Sie im Online-Informationssystem von STEP 7 Basic.



MC_Power aktiviert und deaktiviert eine Achse für die Bewegungssteuerung.



MC_Reset setzt alle Bewegungssteuerungsfehler zurück. Alle Bewegungssteuerungsfehler, die quittiert werden können, werden quittiert.



MC_Home stellt die Beziehung zwischen dem Achsensteuerungsprogramm und dem mechanischen Positionierungssystem der Achse her.



MC_Halt storniert alle Bewegungsvorgänge und bewirkt den Stopp der Achsenbewegung. Die Halteposition ist nicht definiert.



MC_MoveJog führt den Tippbetrieb zu Test- und Anlaufzwecken aus.



MC_MoveAbsolute startet die Bewegung zu einer absoluten Position. Der Auftrag ist beendet, wenn die Zielposition erreicht ist.



MC_MoveRelative startet eine Positionierbewegung relativ zur Startposition.



MC_MoveVelocity bewirkt, dass sich die Achse mit der angegebenen Geschwindigkeit bewegt.

5.4 Weitere Funktionen für einfacheres Programmieren

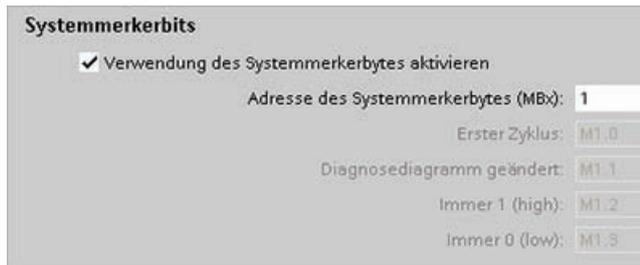
5.4.1 Systemmerker und Taktmerker bieten Standardfunktionen

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

- Sie können im Merkerbereich ein Byte als Systemmerker zuweisen. Das Byte des Systemmerkers bietet die folgenden vier Bits, die von Ihrem Anwenderprogramm referenziert werden können:
 - Das Bit "Immer aus" ist immer auf 0.
 - Das Bit "Immer ein" ist immer auf 1.
 - Das Bit "Diagnosediagramm geändert" wird einen Zyklus lang auf 1 gesetzt, nachdem die CPU ein Diagnoseereignis erfasst hat.
 Die CPU setzt das Bit "Diagnosediagramm geändert" erst im Anschluss an die erste Ausführung der Programmzyklus-OBs. Das Anwenderprogramm kann Änderungen des Diagnoseereignisses nicht erkennen, während die Anlauf-OBs ausgeführt oder die Programmzyklus-OBs das erste Mal ausgeführt werden.
 - Das Bit "Erster Zyklus" 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.)
- 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 Flankenoperationen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwenderprogramm verwenden.

Die CPU initialisiert diese Bytes beim Übergang vom Betriebszustand STOP in STARTUP, und die Bits der Taktmerker wechseln während der Betriebszustände STARTUP und RUN synchron zum CPU-Takt.

Weil Taktmerker und Systemmerker nicht reservierter Speicher im Bereich der Merker sind, können Operationen und Kommunikation in diese Adressen schreiben und Daten beschädigen. Durch Überschreiben der Systemmerker oder Taktmerker können die Daten in diesen Funktionen beschädigt werden, was zu Betriebsfehlern im Anwenderprogramm führen kann. Verwenden Sie bei der Konfiguration der Systemmerker und Taktmerker unbedingt Speicheradressen, auf die kein anderes Element des Anwenderprogramms zugreift.



Der Systemmerker schaltet unter folgenden Bedingungen ein Bit ein (Wert = 1):

- Erster Zyklus: Wird im ersten Zyklus nach dem Anlaufen eingeschaltet
- Diagnosediagramm geändert.
- Immer 1 (high): Immer eingeschaltet
- Immer 0 (low): Immer ausgeschaltet



Der Taktmerker schaltet die einzelnen Bits in festgelegten Abständen ein und aus.

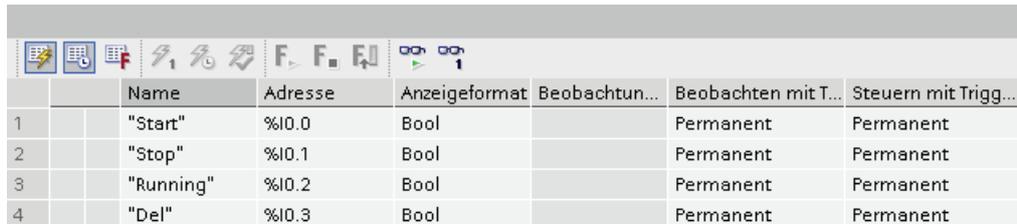
Die Taktmerker erzeugen jeweils einen Rechteckimpuls im entsprechenden Merkerspeicher M. Diese Bits können, vor allem in Verbindung mit Flankenoperationen, als Steuerbits für die zyklische Auslösung von Aktionen im Anwendercode verwendet werden.

Hinweis

Es kann sinnvoll sein, den Systemmerkerbits oder Taktmerkerbits den Namen einer PLC-Variablen zuzuweisen. Der Variablenname kann die Funktion des Bits beschreiben und es somit leichter identifizierbar machen, und er kann einfach von Ihnen in das Anwenderprogramm eingegeben werden.

5.4.2 Beobachtungstabellen vereinfachen die Überwachung des Anwenderprogramms

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.



	Name	Adresse	Anzeigeformat	Beobachtung...	Beobachten mit T...	Steuern mit Trigg...
1	"Start"	%I0.0	Bool		Permanent	Permanent
2	"Stop"	%I0.1	Bool		Permanent	Permanent
3	"Running"	%I0.2	Bool		Permanent	Permanent
4	"Del"	%I0.3	Bool		Permanent	Permanent

Mit einer Beobachtungstabelle können Sie die Ausführung des Anwenderprogramms durch die CPU überwachen und in die Ausführung eingreifen. Die Anzeige und Änderung von Werten ist nicht nur bei den Variablen der Codebausteine und Datenbausteine möglich, sondern auch bei den Speicherbereichen der CPU, darunter den Eingängen und Ausgängen (E und A), peripheren Eingängen, Merkern (M) und DBs (Seite 36). Mit einer Beobachtungstabelle können Sie die Peripherieausgänge (z. B. "Stop:P" oder "A3.4:P") einer CPU freigeben, die sich im Betriebszustand STOP befindet. Beispielsweise können Sie den Ausgängen bestimmte Werte zuweisen, während Sie die Verdrahtung der CPU testen.

Eine Beobachtungstabelle ermöglicht ferner das "Forcen", d. h. das Setzen einer Variablen auf einen bestimmten Wert (Seite 105). Geforcete Werte werden einmal pro Zyklus aufgeschaltet. Sie können während der Programmausführung geändert werden, doch für Ausgänge werden die geforceten Werte am Ende des Zyklus geschrieben. Um einen Eingang oder Ausgang (mit ":P") zu forcen, klicken Sie einfach auf eine der Schaltflächen "Forcen".

5.4.3 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.

Außerdem bietet STEP 7 Basic mehrere globale Bibliotheken zur Verwendung in beliebigen Projekten.

Hinweis

Durch Speichern des Projekts wird jedoch die globale Bibliothek nicht gespeichert oder aktualisiert. Um eine von Ihnen hinzugefügte oder geänderte globale Bibliothek zu speichern, verwenden Sie die Schaltfläche zum Speichern von Änderungen in der Bibliothek, die sich in der Funktionsleiste der globalen Bibliothek befindet.

5.4.4 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.

Spalte	Beschreibung
Objekt	Name des Objekts, das die angegebenen unterlagerten Objekte verwendet oder das von den unterlagerten Objekten verwendet wird
Anzahl	Anzahl Verwendungen
Adresse	Der Ort der Verwendung, z. B. ein Netzwerk
Eigenschaft	Besondere Eigenschaften der referenzierten Objekte, z. B. die Variablennamen in Multiinstanz-Deklarationen
als	Zeigt zusätzliche Informationen zum Objekt an, z. B. ob ein Instanz-DB als Vorlage oder als Multiinstanz verwendet wird
Zugriff	Art des Zugriffs, d. h. ob auf den Operanden Lesezugriff (R) und/oder Schreibzugriff (W) besteht
Adresse	Adresse des Operanden
Typ	Angabe, mit welchem Typ und welcher Sprache das Objekt angelegt wurde
Pfad	Pfad des Objekts in der Projektnavigation

5.4.5 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 Basic zeigt die erste Ebene der Aufrufstruktur hervorgehoben an und zeigt auch die Bausteine an, die durch keinen anderen Baustein im Programm aufgerufen werden. Die erste Ebene der Aufrufstruktur enthält die OBs sowie diejenigen FCs, FBs und DBs, die nicht durch einen OB aufgerufen werden. Von anderen Bausteinen aufgerufene Codebausteine erscheinen eingerückt unter dem aufrufenden Baustein. In der Aufrufstruktur werden nur die Bausteine angezeigt, die von einem Codebaustein aufgerufen werden.

Sie können selektiv nur jene Bausteine anzeigen, die innerhalb der Aufrufstruktur Konflikte verursachen. Folgende Bedingungen führen zu Konflikten:

- Bausteine, die Aufrufe mit älteren oder neueren Zeitstempeln im Code ausführen
- Bausteine, die einen Baustein mit geänderter Schnittstelle aufrufen
- Bausteine, die eine Variable mit geänderter Adresse und/oder geändertem Datentyp verwenden
- Bausteine, die weder direkt noch indirekt durch einen OB aufgerufen werden
- Bausteine, die einen nicht vorhandenen oder fehlenden Baustein aufrufen

Sie können mehrere Bausteinaufrufe und Datenbausteine zu einer Gruppe zusammenfassen. Über eine Klappliste können Sie die Verknüpfungen mit den verschiedenen Aufrufstellen anzeigen.

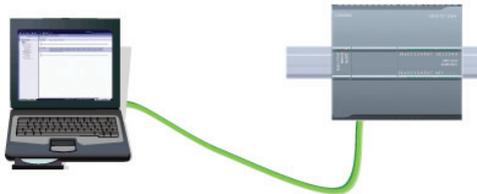
Sie können außerdem eine Konsistenzprüfung durchführen, um Zeitstempelkonflikte aufzuzeigen. Zeitstempelkonflikte können durch die Änderung des Zeitstempels eines Bausteins während oder nach der Programmgenerierung verursacht werden. Diese Konflikte führen zu Inkonsistenzen zwischen den aufrufenden und den aufgerufenen Bausteinen.

- Die meisten Zeitstempel- und Schnittstellenkonflikte lassen sich durch erneutes Übersetzen der Codebausteine beheben.
- Wenn durch Übersetzen die Inkonsistenzen nicht beseitigt werden, navigieren Sie mit dem Programmiereditor über die Verknüpfung in der Spalte "Details" zur Quelle des Problems. Hier können Sie die Inkonsistenzen manuell beseitigen.
- Sind Bausteine rot markiert, müssen sie erneut übersetzt werden.

Einfache Kommunikation zwischen Geräten

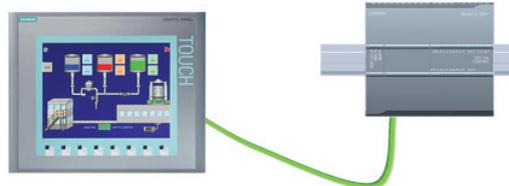
Der integrierte PROFINET-Port der CPU unterstützt die Kommunikation auf Basis von Ethernet- und TCP/IP-Kommunikationsstandards und ermöglicht die Kommunikation mit folgenden Geräten:

- Programmiergerät mit STEP 7 Basic
- HMI-Geräte
- Andere CPUs oder Siemens-fremde Geräte, die Anweisungen von Übertragungsbausteinen (T-Bausteinen) für standardmäßige TCP-Kommunikationsprotokolle nutzen



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 Basic ausgeführt werden.



Bei direkter Verbindung zwischen einem HMI-Bediengerät und einer CPU:

- Das Projekt muss sowohl die CPU als auch das HMI-Panel enthalten.



Bei direkter Verbindung zwischen zwei CPUs:

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

Bei einem Netzwerk mit mehr als zwei untereinander verbundenen Geräten:

- Das Projekt muss die Geräte (CPU und HMI) enthalten. Nehmen Sie den Router nicht in die Konfiguration auf.
- Sie müssen die Netzwerkverbindungen zwischen den Geräten konfigurieren.



Die CPU nutzt die Anwendungsprotokolle TCP (Transport Connection Protocol) und ISO Transport over TCP (RFC 1006). Beim Konfigurieren einer Verbindung mit einer CPU für ISO-over-TCP dürfen Sie in der TSAP-Erweiterung für die passiven Kommunikationsteilnehmer nur ASCII-Zeichen verwenden.

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.

Hinweis

Der PROFINET-Anschluss an der CPU enthält keinen Ethernet-Switch. Zum Anschließen Ihrer CPUs und HMI-Geräte können Sie einen auf dem Baugruppenträger montierten Ethernet-Switch[®] CSM1277 mit 4 Ports von Siemens verwenden.

6.1 PROFINET-Operationen (T-Bausteine)



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 von TCON, TDISCON und TSEND.

Verwenden Sie die Operationen eines T-Bausteins nur in einem Programmzyklus-OB (z. B. OB 1).

- Um eine Verbindung herzustellen, führen Sie TSEND_C mit CONT = 1 aus. Setzen Sie nach erfolgreicher Verbindungsherstellung den Parameter DONE für einen Zyklus.
- Um die Verbindung zu beenden, führen Sie TSEND_C mit CONT = 0 aus. Die Verbindung wird dann sofort abgebrochen. Dies betrifft auch den Empfänger. Die Verbindung wird beendet und es können Daten im Empfangspuffer verloren gehen.
- Um Daten über eine bestehende Verbindung zu senden, muss TSEND_C bei einer steigenden Flanke für REQ ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND_C den Parameter DONE für einen Zyklus auf 1.
- Um eine Verbindung herzustellen und Daten zu senden, muss TSEND_C mit CONT = 1 und REQ = 1 ausgeführt werden. Nach dem erfolgreichen Senden setzt TSEND_C den Parameter DONE für einen Zyklus auf 1.

Hinweis

Durch die asynchrone Bearbeitung von TSEND_C müssen die Daten im Absenderbereich konsistent bleiben, bis einer der Parameter DONE oder ERROR den Wert WAHR annimmt. Für TSEND_C bedeutet der Status WAHR des Parameters DONE, dass die Daten erfolgreich gesendet wurden. Er bedeutet nicht, dass die Partner-CPU den Empfangspuffer tatsächlich liest. Durch die asynchrone Bearbeitung von TRCV_C sind die Daten im Empfangsbereich nur konsistent, wenn Parameter DONE = 1 ist.



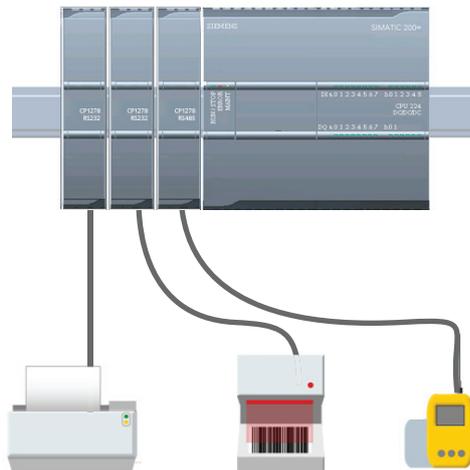
TRCV_C stellt eine TCP- oder ISO-on-TCP-Verbindung zu einer CPU her, empfängt Daten und kann die Verbindung auch wieder beenden. Nach dem Einrichten und Aufbauen der Verbindung wird diese automatisch von der CPU gehalten und überwacht. Die Operation TRCV_C verbindet die Funktionen von TCON, TDISCON und TRCV.

- Um eine Verbindung herzustellen, ist TRCV_C mit dem Parameter CONT = 1 auszuführen.
- Um Daten zu empfangen, führen Sie TRCV_C mit dem Parameter EN_R = 1 aus. Es werden kontinuierlich Daten empfangen, wenn die Parameter EN_R = 1 und CONT = 1 sind.
- Um die Verbindung zu beenden, führen Sie TRCV_C mit dem Parameter CONT = 0 aus. Die Verbindung wird sofort beendet, es können dabei Daten verloren gehen.

Hinweis

Die Verarbeitung der Operationen TSEND_C und TRCV_C kann eine unbestimmte Zeit dauern. Um sicherzustellen, dass diese Operationen in jedem Zyklus verarbeitet werden, müssen Sie sie immer aus dem Zyklus des Hauptprogramms aufrufen, z. B. aus einem Programmzyklus-OB oder aus einem Codebaustein, der aus dem Zyklus des Programms aufgerufen wird. Rufen Sie diese Operationen **nicht** aus einem Prozessalarm-OB, einem Verzögerungsalarm-OB, einem Weckalarm-OB, einem Fehleralarm-OB oder aus einem Anlauf-OB auf.

6.2 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 Gerätearten
- Senden und Empfangen von Daten an Geräte bzw. von Geräten wie z. B. GPS-Geräten, Kamera- oder Überwachungssystemen anderer Hersteller oder Funkmodems

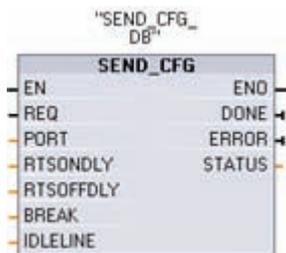
PtP ist eine Form serieller Kommunikation, die verschiedene Baudraten und Paritäten unterstützt. STEP 7 Basic bietet Bibliotheken mit Operationen, die Sie für die Programmierung Ihrer Anwendung nutzen können. Diese Bibliotheken bieten PtP-Kommunikationsfunktionen für das USS-Protokoll für Antriebe (nur RS485-Schnittstelle) sowie für das Modbus-RTU-Master- und Modbus-RTU-Slave-Protokoll.

6.2.1 PtP-Operationen



Mit Hilfe der Operationen 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 Operation konfiguriert die Bedingungen, die den Beginn und das Ende einer empfangenen Meldung kennzeichnen. Meldungen, die diese Bedingungen erfüllen, werden von der Operation RCV_PTP empfangen.



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

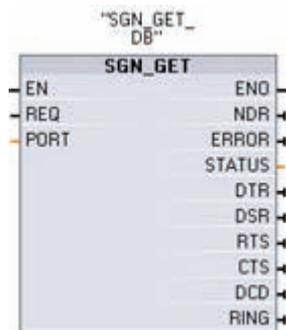


Die Operationen SEND_PTP, RCV_PTP und RCV_RST steuern die PtP-Kommunikation:

- SEND_PTP überträgt den angegebenen Puffer an das CM-Modul. 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-Modul empfangen wurden. Wenn eine Meldung verfügbar ist, wird sie vom CM zur CPU übertragen.
 - Die Operation RCV_RST setzt den Empfangspuffer zurück.
- Jedes CM-Modul kann bis zu 1 KB puffern. Dieser Puffer kann mehreren empfangenen Meldungen zugeordnet werden.



Die Operationen SGN_SET und SGN_GET sind nur für das RS232-CM-Modul gültig. Mit diesen Operationen lesen oder setzen Sie die RS232-Kommunikationssignale.



6.2.2 Bibliothek der USS-Operationen

Die USS-Bibliothek unterstützt das USS-Protokoll und stellt die Funktionen bereit, die speziell für die Kommunikation mit Antrieben über den RS485-Port eines CM-Moduls konzipiert sind. Mit der USS-Bibliothek können Sie den physischen Antrieb und die Parameter zum Lesen und Schreiben des Antriebs steuern. Jedes RS485-CM unterstützt bis zu 16 Antriebe.

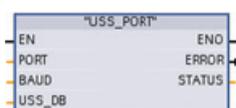
- Über die Operation USS_PORT wird die eigentliche Kommunikation zwischen der CPU und allen Antrieben, die an ein CM angeschlossen sind, abgewickelt. Fügen Sie in Ihrer Anwendung für jedes CM eine andere Operation USS_PORT ein. Stellen Sie sicher, dass das Anwenderprogramm die Operation USS_PORT schnell genug ausführt, um eine Kommunikationszeitüberschreitung durch den Antrieb zu verhindern. Sie können die Operation USS_PORT im Programmzyklus-OB oder in einem beliebigen Alarm-OB verwenden.
- Mit der Operation USS_DRV wird ein angegebener Antrieb im USS-Netzwerk angesteuert. Die Ein- und Ausgangsparameter der Operation USS_DRV entsprechen den Zuständen und Bedienfunktionen des Antriebs. Sind 16 Antriebe im Netzwerk vorhanden, so muss das Programm die Operation USS_DRV mindestens 16 Mal enthalten, d. h. einmal für jeden Antrieb. Stellen Sie sicher, dass die CPU die Operation USS_DRV mit der Geschwindigkeit ausführt, die für die Steuerung der Antriebsfunktionen erforderlich ist. Sie können die Operation USS_PORT nur im Programmzyklus-OB verwenden.
- Mit den Operationen 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 Operationen enthalten, wie benötigt werden. Es kann jedoch zu einem gegebenen Zeitpunkt immer nur eine Lese- oder Schreibanforderung für einen Antrieb aktiv sein. Die Operationen USS_RPM und USS_WPM können nur in einem Programmzyklus-OB verwendet werden.

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



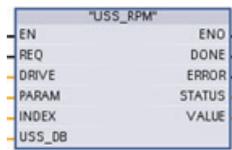
Die Operation USS_DRV tauscht Daten mit dem Antrieb aus, indem Anfragemeldungen erzeugt und die Antwortmeldungen des Antriebs ausgewertet werden. Alle mit einem USS-Netzwerk und CM verbundenen USS-Operationen müssen denselben Instanz-DB verwenden.

Verwenden Sie pro Antrieb jeweils eine separate Operation USS_DRV.

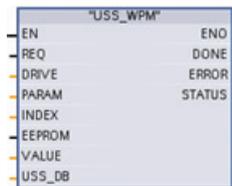


Die Operation USS_PORT bearbeitet die Kommunikation über das USS-Netzwerk. Normalerweise ist pro CM jeweils nur eine USS_PORT-Operation vorhanden, und die Anweisung USS_PORT verwaltet die Übertragung an oder von einem einzelnen Antrieb.

Führen Sie die Operation USS_PORT aus einem Verzögerungsalarm-OB aus, um Antriebs-Timeouts zu verhindern und die letzten USS-Datenaktualisierungen für Aufrufe von USS_DRV verfügbar zu haben.



Die Operation USS_RPM liest einen Parameter aus dem Antrieb. Führen Sie die Operation USS_RPM aus dem Programmzyklus-OB aus.



Die Operation USS_WPM ändert einen Parameter im Antrieb. Führen Sie die Operation USS_WPM aus dem Programmzyklus-OB aus.

Der Parameter "EEPROM" steuert das Schreiben der Daten in den EEPROM. Um die Betriebsdauer des EEPROM zu verlängern, empfiehlt es sich, die Anzahl der EEPROM-Schreiboperationen mithilfe des Parameters "EEPROM" zu minimieren.

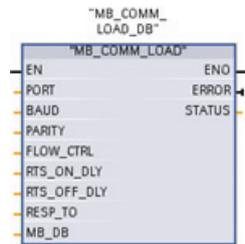
Zeit für die Kommunikation mit dem Antrieb berechnen

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. Standardmäßig führt die Bibliothek für das USS-Protokoll bei jeder Transaktion bis zu 2 Wiederholungen durch.

Baudrate	Berechnetes Mindestintervall für Aufruf von USS_PORT (ms)
1200	790
2400	405
4800	212.5
9600	116.3
19200	68.2
38400	44.1
57600	36.1
115200	28.1

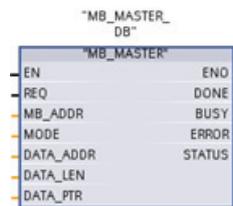
6.2.3 Bibliothek der Modbus-Operationen



Die Operation MB_COMM_LOAD konfiguriert einen Port am Kommunikationsmodul (CM) für die Kommunikation über das Modbus-RTU-Protokoll.

Hierfür können Sie entweder das CM RS232 oder das CM RS485 verwenden.

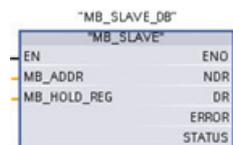
Das Anwenderprogramm muss MB_COMM_LOAD ausführen, um einen Port zu konfigurieren, sodass die Operationen MB_SLAVE oder MB_MASTER mit diesem Port kommunizieren können.



Bei Verwendung der Operation MB_MASTER kann Ihr Anwenderprogramm in der Kommunikation als Modbus-Master fungieren. Sie können auf Daten in einem oder mehreren Modbus-Slave-Geräten zugreifen.

Durch Einfügen der Operation MB_MASTER wird ein Instanz-DB angelegt. Den Namen dieses Datenbausteins geben Sie als Parameter MB_DB zur Operation MB_COMM_LOAD an.

Führen Sie alle Operationen MB_MASTER für einen bestimmten Port aus demselben OB (oder aus derselben OB-Prioritätsstufe) aus.



Bei Verwendung der Operation MB_SLAVE kann Ihr Anwenderprogramm in der Kommunikation als Modbus-Slave fungieren. Ein Modbus-RTU-Master kann eine Anforderung senden, und Ihr Programm antwortet über die Ausführung von MB_SLAVE.

Durch Einfügen der Operation MB_SLAVE wird ein Instanz-DB angelegt. Den Namen dieses Datenbausteins geben Sie als Parameter MB_DB zur Operation MB_COMM_LOAD an.

Führen Sie alle Operationen MB_SLAVE über einen Weckalarm-OB aus.

Die Modbus-Operationen nutzen keine Kommunikations-Interruptereignisse zum Steuern des Kommunikationsprozesses. Ihr Programm muss die Operationen MB_MASTER oder MB_SLAVE auf abgeschlossene Sende- und Empfangsvorgänge abfragen.

Wenn ein Port als Slave auf einen Modbus-Master reagieren soll, kann dieser Port nicht von MB_MASTER verwendet werden. Nur eine Instanz von MB_SLAVE kann für einen bestimmten Port verwendet werden. Entsprechend gilt: Wenn ein Port zum Initiieren von Modbus-Masteranforderungen verwendet werden soll, kann dieser Port nicht von MB_SLAVE verwendet werden. Eine oder mehrere Instanzen von MB_MASTER können mit diesem Port verwendet werden.

Wenn Ihr Programm einen Modbus-Slave betreibt, dann muss MB_SLAVE mit einer Häufigkeit, die eine zeitnahe Antwort auf eingehende Anforderungen eines Modbus-Masters ermöglicht, abfragen (periodische Ausführung).

Wenn Ihr Programm einen Modbus-Master betreibt und mit MB_MASTER eine Anforderung an einen Slave sendet, müssen Sie die Abfragen (die Ausführung von MB_MASTER) fortsetzen, bis die Antwort vom Slave eingegangen ist.

Einfaches Arbeiten mit den integrierten Impulsgeneratoren

7



Sie können die Ausgänge der CPU oder des Signalboards (SB) so konfigurieren, dass sie als Impulsgenerator oder Impulsfolge (PTO) fungieren. Diese Ausgänge werden von der Operation PWM (Impulsdauermodulation) und den grundlegenden Bewegungssteuerungsoperationen verwendet.

Weitere Informationen zu den grundlegenden Bewegungssteuerungsoperationen finden Sie in der Online-Hilfe von STEP 7 Basic.

Hinweis

Impulsfolgen können nicht von anderen Operationen im Anwenderprogramm verwendet werden.

Wenn Sie die Ausgänge der CPU oder des Signalboards als Impulsgeneratoren (für PWM oder grundlegende Bewegungssteuerungsoperationen) konfigurieren, werden die entsprechenden Adressen der Ausgänge (A0.0, A0.1, A4.0 und A4.1) 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 und 20 kHz für die digitalen Ausgänge des Signalboards.

Beachten Sie beim Konfigurieren der grundlegenden Bewegungssteuerungsoperationen, dass Sie von STEP 7 Basic **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.

7.1 Schnelle Zähler

Ein schneller Zähler (HSC) kann als Eingang für einen Winkelschrittgeber dienen. Der Winkelschrittgeber sorgt für eine bestimmte Anzahl von Zählwerten pro Umdrehung sowie für einen Rücksetzimpuls einmal pro Umdrehung. Der bzw. die Taktgeber und der Rücksetzimpuls des Winkelschrittgebers liefern die Eingänge für den schnellen Zähler.

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

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

Da die Interrupts in einer sehr viel geringeren Geschwindigkeit auftreten als der schnelle Zähler zählt, kann eine präzise Steuerung der schnellen Operationen mit relativ geringem Einfluss auf den Zyklus der CPU implementiert werden. Da Sie Interrupts bestimmten Interruptprogrammen zuordnen können, kann jede neue Voreinstellung in einem getrennten Interruptprogramm geladen werden und ermöglicht so eine einfache Steuerung des Zustands. (Sie können alternativ auch alle Interruptereignisse in einem einzigen Interruptprogramm bearbeiten.)

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 und A/B-Zähler. Beachten Sie, dass nicht jeder HSC alle Zählerarten unterstützt. Sie können jeden HSC-Typ mit oder ohne Rücksetzeingang verwenden. Wenn Sie den Rücksetzeingang aktivieren, setzt dieser den aktuellen Wert zurück. Der aktuelle Wert bleibt so lange zurückgesetzt, bis Sie den Rücksetzeingang deaktivieren.

Frequenzfunktion: Der HSC kann in bestimmten Betriebsarten so konfiguriert werden (Zählart), dass er statt der aktuellen Impulszahl eine Frequenz 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 Rücksetzen 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 Flankeninterrupts oder für HSC2 belegt werden.

Beschreibung		Voreinstellung Eingangsbelegung			Funktion	
HSC	HSC1	Integriert oder Signalboard oder Beobachtung PTO 0 ¹	E0.0 E4.0 PTO 0 Impuls	E0.1 E4.1 PTO 0 Richtung	E0.3 E4.3 -	
	HSC2	Integriert oder Signalboard oder Beobachtung PTO 1 ¹	E0.2 E4.2 PTO 1 Impuls	E0.3 E4.3 PTO 1 Richtung	E0.1 E4.1 -	
	HSC3 ²	Integriert	E0.4	E0.5	E0.7	
	HSC4 ³	Integriert	E0.6	E0.7	E0.5	
	HSC5 ⁴	Integriert oder Signalboard	E1.0 E4.0	E1.1 E4.1	E1.2 E4.3	
	HSC6 ⁴	Integriert oder Signalboard	E1.3 E4.2	E1.4 E4.3	E1.5 E4.1	
Betriebsart	Einphasenzähler mit interner Richtungssteuerung		Takt	-	-	Zählwert oder Frequenz
					Rücksetzen	Zählwert
	Einphasenzähler mit externer Richtungssteuerung		Takt	Richtung	-	Zählwert oder Frequenz
						Rücksetzen
	Zweiphasenzähler mit 2 Takteingängen		Takt vorwärts	Takt rückwärts	-	Zählwert oder Frequenz
						Rücksetzen
A/B-Zähler		Phase A	Phase B	-	Zählwert oder Frequenz	
					Phase Z	Zählwert
Beobachtung Impulsgenerator (PTO) ¹		Takt	Richtung	-	Zählwert	

¹ Für die PTO-Beobachtung werden immer Takt und Richtung verwendet. Ist der entsprechende PTO-Ausgang nur für Impulse konfiguriert, so sollte der Richtungsausgang grundsätzlich für Vorwärtzzählen eingerichtet sein.

² HSC3 mit einem Rücksetzeingang ist nicht möglich bei der CPU 1211C, die nur 6 integrierte Eingänge unterstützt.

³ HSC4 ist nicht möglich bei der CPU 1211C, die nur 6 integrierte Eingänge unterstützt.

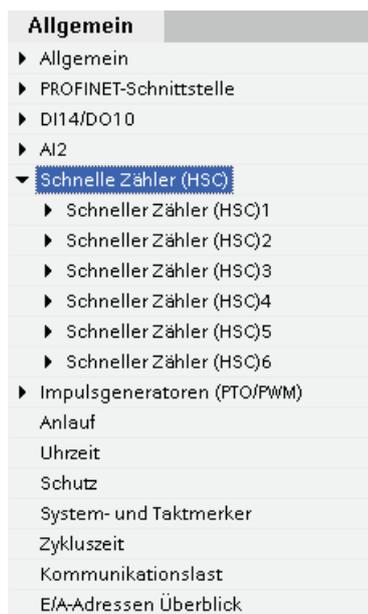
⁴ HSC5 und HSC6 werden von der CPU 1211C und der CPU 1212C nur unterstützt, wenn ein Signalboard eingebaut ist.

Zugreifen auf den aktuellen Wert des HSC

Die CPU speichert den aktuellen Wert jedes HSC in der Adresse eines Eingangs (E). Die folgende Tabelle zeigt die Standardadressen für den aktuellen Wert jedes HSC. Sie können die Eingangsadresse für den aktuellen Wert ändern, indem Sie die Eigenschaften der CPU (Seite 50) ändern.

Schneller Zähler	Datentyp	Standardadresse
HSC1	DInt	ID1000
HSC2	DInt	ID1004
HSC3	DInt	ID1008
HSC4	DInt	ID1012
HSC5	DInt	ID1016
HSC6	DInt	ID1020

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.

Sie richten die Parameter für die schnellen Zähler in den "Eigenschaften" der CPU ein, z. B. Zählerfunktion, Anfangswerte, Rücksetzoptionen und Interruptereignisse. Nach dem Konfigurieren des HSC können Sie mit der Operation CTRL_HSC in Ihrem Anwenderprogramm die Funktionsweise des HSC steuern.

Verwendung der Operation CTRL_HSC



Die Operation CTRL_HSC steuert die schnellen Zähler zum Zählen von Ereignissen, die schneller als der CPU-Zyklus auftreten.

Die einzelnen Operationen CTRL_HSC speichern die Daten in einem Instanz-DB. Dieser Instanz-DB wird durch das Einfügen der Operation CTRL_HSC in Ihr Anwenderprogramm angelegt.

Parameter	Datentyp	Beschreibung
HSC	HW_HSC	HSC-Kennung
DIR	BOOL	1 = Neue Richtung anfordern
CV	BOOL	1 = Anforderung zum Setzen eines neuen Zählwerts
RV	BOOL	1 = Anforderung zum Setzen eines neuen Referenzwerts
PERIOD	BOOL	1 = Anforderung zum Setzen eines neuen Zeitintervalls (nur bei Frequenzmessung)
NEW_DIR	INT	Neue Richtung: 1= vorwärts, -1= rückwärts
NEW_DIR	DINT	Neuer Zählwert
NEW_DIR	DINT	Neuer Referenzwert
NEW_DIR	INT	Neuer Zeitintervallwert in Sekunden: 0,01, 0,1 oder 1 (nur bei Frequenzmessung)
BUSY	BOOL	Funktion besetzt
STATUS	WORD	Ausführungsbedingung

Während die Zählfrequenz der Zähloperationen CTU, CTD und CTUD durch die Abtastrate der CPU begrenzt ist, arbeiten die schnellen Zähler asynchron zum CPU-Abtastzyklus und ermöglichen das Zählen von Ereignissen mit einer Zählfrequenz bis zu 100 kHz (HSC 1, 2 oder 3 und integrierter CPU-Zähleingang).

Sie müssen die schnellen Zähler in den Projekteinstellungen für die CPU-Gerätekonfiguration einrichten, bevor sie im Programm eingesetzt werden können. Bei der HSC-Gerätekonfiguration werden Zählermodus, E/A-Anschaltungen, Interruptzuweisung und Betrieb als schneller Zähler oder als Gerät für die Impulsfrequenzmessung eingestellt. Ein schneller Zähler kann mit oder ohne Programmsteuerung funktionieren.

Viele Parameter für die Konfiguration schneller Zähler werden nur in der Gerätekonfiguration des Projekts eingestellt. Einige schnelle Zähler werden in der Gerätekonfiguration des Projekts eingerichtet, können aber später innerhalb der Programmsteuerung geändert werden. Die Parameter der Operation CTRL_HSC bewirken die Programmsteuerung des Zählvorgangs:

- Setzen der Zählrichtung auf einen Wert NEW_DIR
- Setzen des aktuellen Zählwerts auf einen neuen Wert NEW_DIR
- Setzen des Referenzwerts auf einen neuen Wert NEW_DIR
- Setzen des Zeitintervallwerts (nur bei Frequenzmessung) auf einen neuen Wert NEW_PERIOD

Sind die folgenden Booleschen Merker auf 1 gesetzt, wenn die Operation 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 Operation CTRL_HSC verarbeitet.

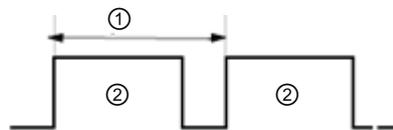
- DIR = 1 ist eine Anforderung zum Laden eines Werts NEW_DIR, 0 = keine Änderung
- CV = 1 ist eine Anforderung zum Laden eines Werts NEW_CV, 0 = keine Änderung
- RV = 1 ist eine Anforderung zum Laden eines Werts NEW_RV, 0 = keine Änderung
- PERIOD = 1 ist eine Anforderung zum Laden eines Werts NEW_PERIOD, 0 = keine Änderung

Die Operation CTRL_HSC wird üblicherweise in einem Prozessalarm-OB platziert, der ausgeführt wird, wenn das Prozessalarmereignis des Zählers ausgelöst wird. Wird zum Beispiel der Zählerinterrupt durch ein Ereignis CV=RV ausgelöst, so kann ein Prozessalarm-OB eine Operation CTRL_HSC ausführen, die den Referenzwert durch Laden eines Werts NEW_RV ändert.

Der aktuelle Zählwert ist in den Parametern für CTRL_HSC nicht vorgesehen. Die Adresse des Prozessabbilds, in der der aktuelle Zählwert gespeichert wird, wird bei der Hardwarekonfiguration dem schnellen Zähler zugewiesen. Sie können den Zählwert über die Programmlogik direkt auslesen, und der an Ihr Programm ausgegebene Wert ist ein korrekter Zählwert für den Moment, in dem der Zähler gelesen wurde, doch der Zähler setzt die Zählung schneller Ereignisse fort. Der tatsächliche Zählwert kann sich ändern, bevor Ihr Programm einen Prozess mit einem alten Zählwert beendet.

7.2 Impulsdauermodulation (PWM)

Zur Steuerung schneller Impulsausgänge stehen zwei Impulsgeneratoren zur Verfügung: Impulsdauermodulation (PWM) und Impulsfolge (PTO). Weil der PWM-Ausgang zwischen 0 und Vollauschlag liegen kann, bietet er einen digitalen Ausgang, der in vielerlei Hinsicht einem Analogausgang gleicht. Der PWM-Ausgang kann z. B. zur Steuerung der Drehzahl eines Motors vom Stillstand bis zur vollen Drehzahl dienen oder er kann dafür eingesetzt werden, die Position eines Ventils von geschlossen bis vollständig geöffnet zu steuern. PTO wird von den Bewegungssteuerungsoperationen genutzt.



- 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 Vollauschlag (kein Impuls, immer ein) liegen.

Die Operation 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 je nach Bedarf verändert, um die Steuerung zu beeinflussen.

Sie können jeden Impulsgenerator entweder PWM oder PTO zuordnen, jedoch nicht beiden gleichzeitig.

Impulsgeneratoren konfigurieren

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.

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	--

Um die Impulsdauermodulation vorzubereiten, konfigurieren Sie zunächst einen Impulskanal in der Gerätekonfiguration durch Auswahl der CPU, von "Impulsgenerator (PTO/PWM)" und entweder "PWM1" oder "PWM2". Aktivieren Sie den Impulsgenerator (Optionskästchen). Wenn ein Impulsgenerator aktiviert wird, wird diesem bestimmten Impulsgenerator ein eindeutiger Standardname zugewiesen. Diesen Namen können Sie durch Bearbeiten im Eingabefeld "Name:" ändern, er muss jedoch eindeutig sein. Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablen-tabelle "Konstanten" und können als PWM-Parameter der Operation CTRL_PWM genutzt werden. Sie können den Impulsgenerator umbenennen, mit Kommentar versehen und parametrieren. Dazu gehen Sie wie folgt vor:

- Der Impulsgenerator wird wie folgt genutzt: PWM oder PTO (wählen Sie PWM)
- Ausgangsquelle: In CPU integriert oder Signalboard
- Zeitbasis: Millesekunden oder Mikrosekunden
- Impulsdauerformat:
 - Prozent (0 bis 100)
 - Tausendstel (0 bis 1000)
 - Zehntausendstel (0 bis 10000)
- Zykluszeit: Geben Sie Ihre Zykluszeit ein. Dieser Wert kann nur hier geändert werden.
- Anfangsimpulsdauer: Geben Sie die Anfangsimpulsdauer ein. Die Impulsdauer kann während der Laufzeit geändert werden.
- Anfangsadresse: Geben Sie die Adresse des Ausgangs (A), an der Sie den Wert für die Impulsdauer ablegen möchten, als Datenwort ein. Die Standardadresse ist AW1000 für PWM1, und AW1002 für PWM2. Der Wert an dieser Adresse bestimmt die Dauer des Impulses und wird jedes Mal auf den oben angegebenen Wert für die "Anfangsimpulsdauer" initialisiert, wenn das Zielsystem vom Betriebszustand STOP in den Betriebszustand RUN wechselt. Sie ändern diesen Ausgangswortwert während der Laufzeit, um die Impulsdauer zu verändern.

Verwendung der Operation CTRL_PWM



Wenn Sie die Operation CTRL_PWM in den Programmiereditor einfügen, wird ein DB zugewiesen. Die Operation CTRL_PWM speichert Parameterdaten in einem Datenbaustein (DB). Die Parameter des Datenbausteins werden von der Operation CTRL_PWM bestimmt.

Parameter	Datentyp	Beschreibung
PWM	WORD	PWM-Kennung: Die Namen aktivierter Impulsgeneratoren werden zu Variablen in der Variablen-tabelle "Konstanten" und können als PWM-Parameter genutzt werden.
ENABLE	BOOL	1 = Impulsgenerator starten 0 = Impulsgenerator stoppen
BUSY	BOOL	Funktion besetzt
STATUS	WORD	Ausführungsbedingung

Den aktivierten Impulsgenerator geben Sie mit dem Variablennamen für den PWM-Parameter an.

Wenn der Eingang EN gleich WAHR ist, startet oder stoppt die Operation PWM_CTRL die angegebene PWM anhand des Werts am Eingang ENABLE. Die Impulsdauer wird von dem Wert der zugewiesenen Adresse des Ausgangsworts angegeben. Weil die CPU die Anforderung verarbeitet, während die Operation CTRL_PWM ausgeführt wird, meldet der Parameter BUSY bei S7-1200-CPU immer FALSCH.

Die Impulsdauer wird beim ersten Wechsel des Zielsystems in RUN auf den in der Gerätekonfiguration eingegebenen Anfangswert gesetzt. Um die Impulsdauer zu ändern, schreiben Sie die gewünschten Werte in die Ausgangsadresse, die in der Gerätekonfiguration als Datenwort angegeben wurde. ("Ausgangsadressen"/"Anfangsadresse:"). Um die gewünschte Impulsdauer in das entsprechende Datenwort für den Ausgang zu schreiben, nutzen Sie eine Operation wie MOVE, CONVERT, PID oder eine Operation der Kategorie "Mathematische Operationen". Sie müssen dabei den gültigen Bereich des Ausgangswertes beachten (d. h. Prozent, Tausendstel, Zehntausendstel oder S7-Analogformat).

Einfaches Arbeiten mit den Online-Tools

8.1 Online-Verbindung mit einer CPU herstellen

Durch die Onlineverbindung stehen Ihnen zusätzliche Funktionen zur Verfügung.

- Wechseln des Betriebszustands der CPU (Seite 99) über das CPU-Bedienpanel
- Zurückladen, Vergleichen und Synchronisieren der Codebausteine des Anwenderprogramms (Seite 100)
- Mit einer Beobachtungstabelle (Seite 103) das Anwenderprogramm testen und Werte in der CPU forcen (Seite 105)
- Mit dem Diagnosepuffer (Seite 103) Ereignisse anzeigen



Um Ihr Projekt (einschließlich Anwenderprogramm, Gerätekonfiguration und IP-Adresse) zu laden, stellen Sie eine Onlineverbindung zu einer CPU her. Verwenden Sie zum Herstellen der Verbindung zu einer Online-CPU den Ordner "Online-Zugänge".

1. Öffnen Sie den Ordner "Online-Zugänge" und wählen Sie die Onlineverbindung für Ihre CPU.
2. Doppelklicken Sie auf "Erreichbare Teilnehmer aktualisieren", um die Online-CPU anzuzeigen.

Über die Taskcard "Online-Tools" greifen Sie auf die Daten der Online-CPU zu.

8.2 Laden einer IP-Adresse in eine Online-CPU



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

- IP-Adresse für die CPU (Seite 52) einrichten
- 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" (Seite 97) 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".

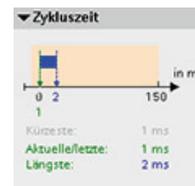
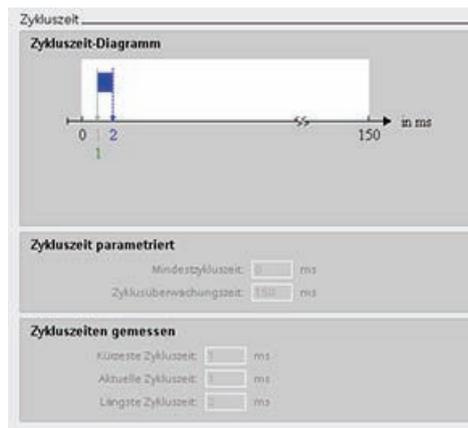
8.3 Interaktion mit der Online-CPU

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 Umräumen des Speichers.



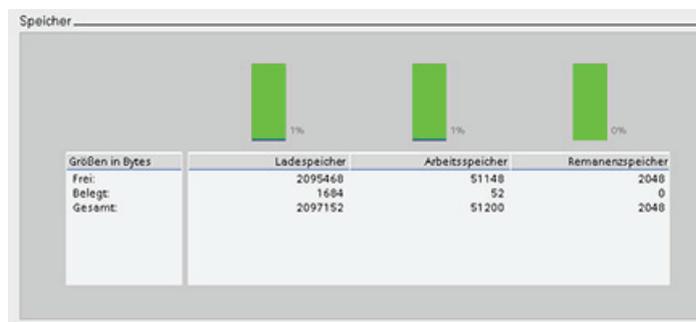
Der aktuelle Betriebszustand der CPU wird durch die Farbe der RUN/STOP-Anzeige angegeben: Gelb steht für den Betriebszustand STOP, Grün für RUN.

Um das Bedienpanel verwenden zu können, müssen Sie 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.



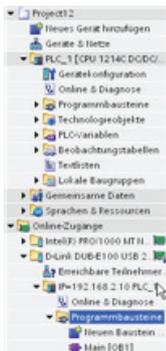
Sie können die Zykluszeit einer Online-CPU überwachen.

Außerdem können Sie die Speicherauslastung der CPU anzeigen.



8.4 Laden von der Online-CPU

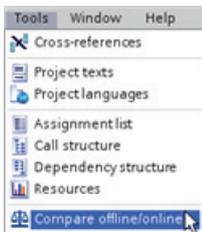
STEP 7 stellt zwei Verfahren bereit, um Codebausteine des Anwenderprogramms von einer Online-CPU zu laden.



In der Projektnavigation können Sie mit der Maus Codebausteine von der Online-CPU ziehen und in einer CPU in Ihrem Offline-Projekt ablegen.

1. Erweitern Sie anschließend bei geöffnetem Projekt den Behälter "Online-Zugänge", und wählen Sie eine Online-CPU aus.
2. Erweitern Sie die Online-CPU, um die Codebausteine des Anwenderprogramms anzuzeigen.
3. Ziehen Sie den Ordner "Programmbausteine" mit der Maus von der Online-CPU in den Behälter "Programmbausteine" der CPU im Offline-Projekt.

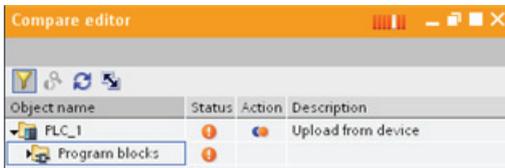
STEP 7 Basic kopiert die Codebausteine von der Online-CPU in Ihr Offline-Projekt.



Sie können auch mit der Funktion "Vergleichen" die Codebausteine der Online-CPU und der Offline-CPU synchronisieren:

1. Wählen Sie die Offline-CPU aus.
2. Wählen Sie im Menü "Werkzeuge" den Befehl "Offline/online vergleichen".

Wenn die Codebausteine der Offline-CPU nicht den Codebausteinen der Online-CPU entsprechen, haben Sie im Editor "Vergleichen" die Möglichkeit, die beiden CPUs zu synchronisieren.



Klicken Sie auf das Symbol für "Aktion", anzugeben, ob Sie Zurückladen, Laden oder keine Aktion wünschen.

Durch Klicken auf die Schaltfläche "Synchronisieren" laden Sie die Codebausteine in die oder aus der vorgesehenen CPU.

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.

Zunächst müssen Sie die CPU mit dem Programmiergerät verbinden und ein neues Projekt anlegen.



Zum Einfügen einer "nicht spezifizierten CPU" verwenden Sie eine der folgenden Optionen:

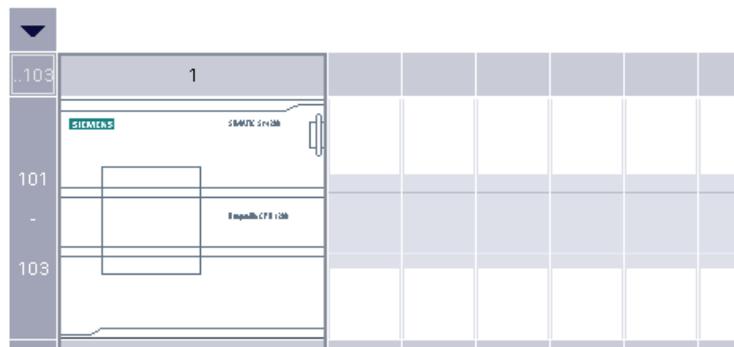
- Fügen Sie in der Gerätekonfiguration (Projektansicht oder Portalansicht) ein neues Gerät hinzu, wählen jedoch anstelle einer bestimmten CPU die "nicht spezifizierte CPU".
- Klicken Sie in der Portalansicht unter "Erste Schritte" auf "Ein PLC-Programm erstellen".

STEP 7 Basic 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.



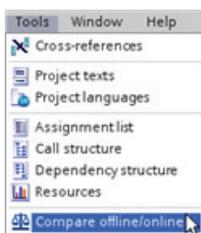
Das Gerät ist nicht spezifiziert.

- 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 Basic 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.



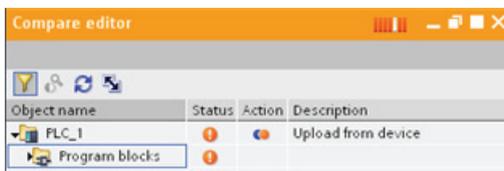
8.5 Vergleichen von Offline- und Online-CPU's



Sie können die Codebausteine in einer Online-CPU mit den Codebausteinen Ihres Offline-Projekts wie folgt vergleichen:

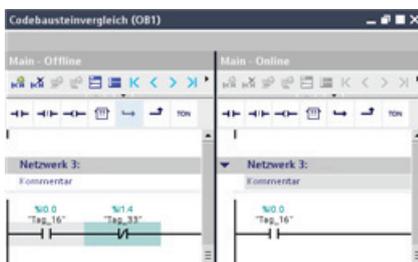
1. Wählen Sie die Offline-CPU aus.
2. Wählen Sie im Menü "Werkzeuge" den Befehl "Offline/online vergleichen".

Wenn die Codebausteine der Offline-CPU nicht den Codebausteinen der Online-CPU entsprechen, haben Sie im Editor "Vergleichen" die Möglichkeit, die beiden CPUs zu synchronisieren.



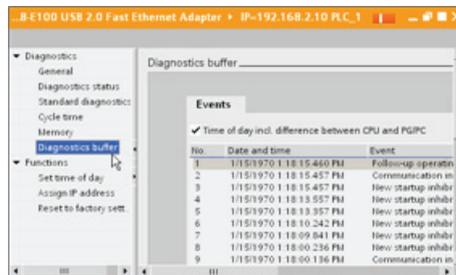
Klicken Sie auf das Symbol für "Aktion", anzugeben, ob Sie Zurückladen, Laden oder keine Aktion wünschen.

Durch Klicken auf die Schaltfläche "Synchronisieren" laden Sie die Codebausteine in die vorgesehene CPU.



Klicken Sie auf die Schaltfläche für "detaillierten Vergleich", um die Codebausteine nebeneinander anzuzeigen. Bei diesem Detailvergleich werden die gefundenen Unterschiede zwischen den Codebausteinen von Online- und Offline-CPU's hervorgehoben.

8.6 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.

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.

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.

8.7 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), Peripherieeingänge oder -ausgänge (z. B. "On:P", "E 3.4:P" oder "A 3.4:P"), Merker (M) oder einen DB handeln. Die Beobachtungsfunktion ändert nicht den Programmablauf. Sie liefert Ihnen Informationen zum Programmablauf und den Daten des Programms in der CPU. Sie können auch mit den Funktionen "Ändern" und "Forcen" die Ausführung Ihres Anwenderprogramms testen.

	Name	Adresse	Anzeigeformat	Beobachtun...	Beobachten mit T...	Steuern mit Trigg...
1	"Start"	%I0.0	Bool		Permanent	Permanent
2	"Stop"	%I0.1	Bool		Permanent	Permanent
3	"Running"	%I0.2	Bool		Permanent	Permanent
4	"Del"	%I0.3	Bool		Permanent	Permanent

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. Sie können auch Variablen "forcen", d. h. zwangsweise auf einen Wert setzen. Sie können festlegen, dass die Variable zu Beginn oder am Ende des Zyklus beobachtet oder geändert werden soll, beim Wechsel der CPU in den Betriebszustand STOP oder "dauerhaft", d. h. ohne dass der Wert nach einem Wechsel von STOP nach RUN zurückgesetzt wird.



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.

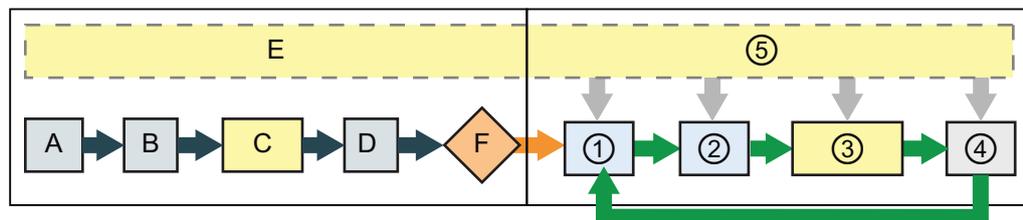
8.8 Variablen in der CPU forcen

Eine Beobachtungstabelle enthält eine 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.

- Die CPU überschreibt vor der Ausführung des Zyklus den Wert des Peripherieeingangs mit dem "geforcen" Wert. Das Anwenderprogramm nutzt den geforcen Wert während der Bearbeitung.
- Am Ende des Zyklus überschreibt die CPU die vom Anwenderprogramm erzeugten Ausgangswerte mit einem beliebigen geforcen Wert, der für die Peripherieausgänge angegeben wurde. Der geforcete Wert erscheint am physischen Ausgang und wird im Prozess verwendet.

Wenn ein Eingang oder Ausgang in einer Beobachtungstabelle geforcet wird, werden die Force-Aktionen Teil des Anwenderprogramms. Beim Schließen von STEP 7 Basic bleiben die geforceten Elemente für das momentan vom CPU-Programm ausgeführte Anwenderprogramm so lange aktiv, bis sie gelöscht werden. Um diese geforceten Elemente zu löschen, müssen Sie über STEP 7 Basic eine Verbindung zur Online-CPU herstellen und dann mit Hilfe der Beobachtungstabelle die Forcefunktion für diese Elemente deaktivieren oder stoppen.

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 geforcete Elemente mit an die CPU übertragen.



Anlauf

- A Das Löschen des Speicherbereichs E wird von der Forcefunktion nicht beeinflusst.
- B Die Initialisierung der Ausgangswerte wird von der Forcefunktion nicht beeinflusst.
- C Während der Ausführung der Anlauf-OBs schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift.
- D Nach dem Kopieren des Zustands der physischen Ausgänge in den Speicherbereich E schaltet die CPU die Forcewerte auf.
- E Das Speichern von Interruptereignissen in der Warteschlange wird nicht beeinflusst.
- F Die Freigabe des Schreibens in die Ausgänge wird nicht beeinflusst.

RUN

- ① Beim Schreiben von A-Speicher in die physischen Ausgänge schaltet die CPU den Forcewert bei der Aktualisierung der Ausgänge auf.
- ② Nach dem Kopieren des Zustands der physischen Ausgänge in den Speicherbereich E schaltet die CPU die Forcewerte auf.
- ③ Während der Ausführung der Anwenderprogramms (Programmzyklus-OBs) schaltet die CPU den Forcewert auf, wenn das Anwenderprogramm auf den physischen Eingang zugreift.
- ④ Die Selbstdiagnose wird von der Funktion "Forcen" nicht beeinflusst.
- ⑤ Die Bearbeitung von Kommunikatonsanforderungen und Verarbeitung von Interrupts während eines beliebigen Teils des Zyklus wird nicht beeinflusst.

Technische Daten

A.1 Allgemeine technische Daten

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



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
- EU-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



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.



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



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.



Das Automatisierungssystem S7-1200 erfüllt die Anforderungen der Normen nach AS/NZS 2064 (Klasse A)

Zulassung für das Seewesen: Die S71200 Produkte werden regelmäßig für die Zulassung 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)

Industrienumgebungen: Das Automatisierungssystem S7-1200 wurde für den Einsatz in Industrienumgebungen entwickelt.

Anwendungsgebiet	Anforderungen an die Störaussendung	Anforderungen an die Störfestigkeit
Industrie	EN 61000-6-4:2007	EN 61000-6-2:2005

Elektromagnetische Verträglichkeit: 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.

Elektromagnetische Verträglichkeit - Entstörung nach EN 61000-6-2	
EN 61000-4-2 Elektrostatische Entladung	8 kV Entladung durch die Luft an allen Oberflächen 6 kV Entladung durch Kontakt mit freiliegenden leitenden Oberflächen
EN 61000-4-3 Abgestrahltes elektromagnetisches Feld	80 bis 100 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 6100-4-5 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

Elektromagnetische Verträglichkeit - Leitungsgeführte und abgestrahlte Störungen nach EN 61000-6-4	
Leitungsgeführte Störaussendungen EN 55011, Klasse A, Gruppe 1 0,15 MHz bis 0,5 MHz 0,5 MHz bis 5 MHz 5 MHz bis 30 MHz	<79 dB (µV) Quasi-Spitze; <66 dB (µV) Mittelwert <73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert <73 dB (µV) Quasi-Spitze; <60 dB (µV) Mittelwert
Störaussendungen EN 55011, Klasse A, Gruppe 1 30 MHz bis 230 MHz 230 MHz bis 1 GHz	<40 dB (µV/m) Quasi-Spitze; gemessen bei 10m <47 dB (µV/m) Quasi-Spitze; gemessen in einer Entfernung von 10 m

Umgebungsbedingungen

Umgebungsbedingungen - Transport und Lagerung	
EN 6006822, Test Bb, trockene Wärme und EN 6006821 Test Ab, Kälte	-40 °C bis +70 °C
EN 60068230, Test Db, feuchte Wärme	25 °C bis 55 °C, 95 % Luftfeuchtigkeit
EN 60068-2-14, Test Na, Temperaturschock	-40 °C bis +70 °C, Haltezeit 3 Stunden, 2 Zyklen
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)

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 2000 m)
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: 7,00 mm von 5 bis 9 Hz; 2 G von 9 bis 150 Hz 10 Ablenkungen je Achse, 1 Oktave/Minute

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 115/230V-Stromkreis	1.500 V AC Routineprüfung/1950 V DC Typprüfung
115/230VStromkreis an 24/5VStromkreis	1.500 V AC Routineprüfung/3250 V DC Typprüfung

Schutzklasse: Schutzklasse II nach EN 61131-2 (Schutzleiter nicht erforderlich)

Schutzart

- 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.

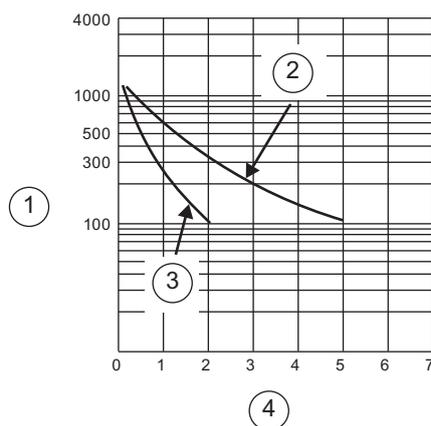
Bemessungsspannungen

Bemessungsspannung	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 müssen Sie berücksichtigen, vor allem, wenn Sie mit Geräten arbeiten, die auf kurze Impulse reagieren.

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 (x 10³ 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=7 ms)
- ④ Nennbetriebsstrom (A)

A.2 CPU-Module

Die vollständigen technischen Daten der S7-1200 CPU-Module entnehmen Sie bitte dem Systemhandbuch.

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 <ul style="list-style-type: none"> • AC/DC/Relais • DC/DC/Relais • DC/DC/DC 	<ul style="list-style-type: none"> • 420 Gramm • 380 Gramm • 370 Gramm 	<ul style="list-style-type: none"> • 425 Gramm • 385 Gramm • 370 Gramm 	<ul style="list-style-type: none"> • 475 Gramm • 435 Gramm • 415 Gramm
Leistungsverlust <ul style="list-style-type: none"> • AC/DC/Relais • DC/DC/Relais • DC/DC/DC 	<ul style="list-style-type: none"> • 10 W • 8 W • 8 W 	<ul style="list-style-type: none"> • 11 W • 9 W • 9 W 	<ul style="list-style-type: none"> • 14 W • 12 W • 12 W
Verfügbare Strom (5 VDC) für SM- und CM-Bus	max. 750 mA	max. 1000 mA	max. 1600 mA
Verfügbare 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

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C
Anwenderspeicher <ul style="list-style-type: none"> • Arbeitsspeicher • Ladespeicher • Remanenter Speicher 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB 	<ul style="list-style-type: none"> • 25 KB • 1 MB • 2 KB 	<ul style="list-style-type: none"> • 50 KB • 2 MB • 2 KB
Integrierte digitale E/A	6 Eingänge 4 Ausgänge	8 Eingänge 6 Ausgänge	14 Eingänge 10 Ausgänge
Integrierte analoge E/A	2 Eingänge	2 Eingänge	2 Eingänge
Größe des Prozessabbilds <ul style="list-style-type: none"> • Eingänge • Ausgänge 	<ul style="list-style-type: none"> • 1024 Byte • 1024 Byte 	<ul style="list-style-type: none"> • 1024 Byte • 1024 Byte 	<ul style="list-style-type: none"> • 1024 Byte • 1024 Byte
Merker (M)	4096 Byte	4096 Byte	8192 Byte
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	max. 3 CM	max. 3 CM
Schnelle Zähler <ul style="list-style-type: none"> • Einphasenzähler (Taktfrequenz) • A/B-Zähler (Taktfrequenz) 	3 gesamt <ul style="list-style-type: none"> • 3 bei 100 kHz • 3 bei 80 kHz 	4 gesamt <ul style="list-style-type: none"> • 3 bei 100 kHz und 1 bei 30 kHz • 3 bei 80 kHz und 1 bei 20 kHz 	6 gesamt <ul style="list-style-type: none"> • 3 bei 100 kHz und 3 bei 30 kHz • 3 bei 80 kHz und 3 bei 20 kHz
Impulsausgänge	2	2	2
Eingänge für Impulsabgriff	6	8	14

CPU-Merkmale	CPU 1211C	CPU 1212C	CPU 1214C
Verzögerungs-/Weckalarme	4 gesamt mit Auflösung von 1 ms	4 gesamt mit Auflösung von 1 ms	4 gesamt mit Auflösung von 1 ms
Flankeninterrupts Optional mit SB	6 steigend und 6 fallend 10 steigend und 10 fallend	8 steigend und 8 fallend 12 steigend und 12 fallend	12 steigend und 12 fallend 14 steigend und 14 fallend
Echtzeituhr • Genauigkeit • Pufferung (wartungsfreier Hochleistungskondensator)	• +/- 60 Sekunden/Monat • Typ. 10 Tage/min. 6 Tage bei 40 °C	• +/- 60 Sekunden/Monat • Typ. 10 Tage/min. 6 Tage bei 40 °C	• +/- 60 Sekunden/Monat • Typ. 10 Tage/min. 6 Tage bei 40 °C
Ausführungsgeschwindigkeit • Boolescher Wert • Wort übertragen • Realzahlenarithmetik	• 0,1 µs/Operation • 12 µs/Operation • 18 µs/Operation	• 0,1 µs/Operation • 12 µs/Operation • 18 µs/Operation	• 0,1 µs/Operation • 12 µs/Operation • 18 µs/Operation
Kommunikation • Datenraten • Elektrische Trennung (externes Signal zu PLC-Logik) • Kabelart	1 Ethernet-Port • 10/100 MBit/s • Wandler potentialgetrennt, 1500 V DC • CAT5e geschirmt	1 Ethernet-Port • 10/100 MBit/s • Wandler potentialgetrennt, 1500 V DC • CAT5e geschirmt	1 Ethernet-Port • 10/100 MBit/s • Wandler potentialgetrennt, 1500 V DC • CAT5e geschirmt
Anschlüsse • HMI • PG • Anwenderprogramm • CPU-CPU	• 3 • 1 • 8 • 3	• 3 • 1 • 8 • 3	• 3 • 1 • 8 • 3

Digitaleingänge	Beschreibung	
Eingänge • CPU 1211C • CPU 1212C • CPU 1214C	Summe • 6 • 8 • 14	Anzahl, die gleichzeitig eingeschaltet sein kann • 6 • 8 • 14
Typ	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)	
Nennspannung	24 V DC bei 4 mA, Nennwert	
Zulässige Dauerspannung	max. 30 V DC	
Stoßspannung	35 V DC für 0,5 s	
Signal logisch 1 (min.)	15 V DC bei 2,5 mA	
Signal logisch 0 (max.)	5 V DC bei 1 mA	
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute	
Potentialgetrennte Gruppen	1	
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)	

A.2 CPU-Module

Digitaleingänge	Beschreibung
HSC Eingangstaktfrequenzen (max.)	Pegel logisch 1 = 15 bis 26 VDC
<ul style="list-style-type: none"> • CPU 1211C • CPU 1212C • CPU 1214C 	<ul style="list-style-type: none"> • Einphasenzähler: 100 KHz A/B-Zähler: 80 kHz • Einphasenzähler: 100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Ea.7) A/B-Zähler: 80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Ea.7) • Einphasenzähler: 100 kHz (Ea.0 bis Ea.5) und 30 kHz (Ea.6 bis Eb.5) A/B-Zähler: 80 kHz (Ea.0 bis Ea.5) und 20 kHz (Ea.6 bis Eb.5)
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt, 50 geschirmt für HSC-Eingänge

Digitalausgänge	Relais	DC
Anzahl Ausgänge	AC/DC/Relais und DC/DC/Relais	DC/DC/DC
	<ul style="list-style-type: none"> • CPU 1211C: 4 • CPU 1212C: 6 • CPU 1214C: 10 	<ul style="list-style-type: none"> • CPU 1211C: 4 • CPU 1212C: 6 • CPU 1214C: 10
Anzahl Ausgänge, die gleichzeitig eingeschaltet sein kann	AC/DC/Relais und DC/DC/Relais	DC/DC/DC
	<ul style="list-style-type: none"> • CPU 1211C: 4 • CPU 1212C: 6 • CPU 1214C: 10 	<ul style="list-style-type: none"> • CPU 1211C: 4 • CPU 1212C: 6 • CPU 1214C: 10
Typ	Relais, Trockenkontakt	MOSFET, elektronisch (stromliefernd)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	-/-	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	-/-	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Widerstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	-/-	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein
Elektrische Trennung (Feld zu Logik)	<ul style="list-style-type: none"> • Spule zu Kontakt 1500 V AC für 1 Minute • Spule zu Logik: Keine 	500 V AC für 1 Minute
Isolationswiderstand	min. 100 MΩ, wenn neu	-/-
Elektrische Trennung zwischen offenen Kontakten	750 V AC für 1 Minute	-/-
Potentialgetrennte Gruppen	AC/DC/Relais und DC/DC/Relais	DC/DC/DC
	<ul style="list-style-type: none"> • CPU 1211C: 1 • CPU 1212C: 2 • CPU 1214C: 2 	<ul style="list-style-type: none"> • CPU 1211C: 1 • CPU 1212C: 1 • CPU 1214C: 1
Induktive Klemmspannung	-/-	L+ minus 48 V DC, 1 W Verlustleistung
Schaltverzögerung (Aa.0 bis Aa.3)	max. 10 ms	max. 1,0 µs von Aus nach Ein max. 3,0 µs von Ein nach Aus
Schaltverzögerung (Aa.4 bis Ab.1)	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus

Digitalausgänge	Relais	DC
Frequenz Impulsgenerator (Aa.0 und Aa.2)	Nicht empfohlen	max. 100 KHz, min. 2 Hz
Mechanische Lebensdauer (ohne Last)	10.000.000 Schaltspiele auf/zu	-/-
Lebensdauer der Kontakte bei Nennlast	100.000 Schaltspiele auf/zu	
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Leitungslänge (Meter)	500 m geschirmt, 150 m ungeschirmt	

Analogeingänge	Beschreibung
Anzahl und Art der Eingänge	2 Spannungseingänge (asymmetrisch) CPU 1211C, CPU 1212C und CPU 1214C
Bereich	0 bis 10 V
Vollausschlag (Datenwort) ¹	0 bis 27648
Überschwingbereich (Datenwort) ¹	27649 bis 32511
Überlauf (Datenwort) ¹	32512 bis 32767
Auflösung	10 Bit
Max. Stehspannung	35 V DC
Glättung ²	Keine, schwach, mittel oder stark
Rauschunterdrückung ³	10, 50 oder 60 Hz
Impedanz	≥100 kΩ
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / 0 bis 55 °C)	3,0 % / 3,5 % des Vollausschlags
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz
Betriebssignalebereich	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein
Leitungslänge	100 m, verdreht und geschirmt

¹ Darstellungen der Spannungs- und Stromdaten für die Analogeingänge enthält das Systemhandbuch der S7-1200.

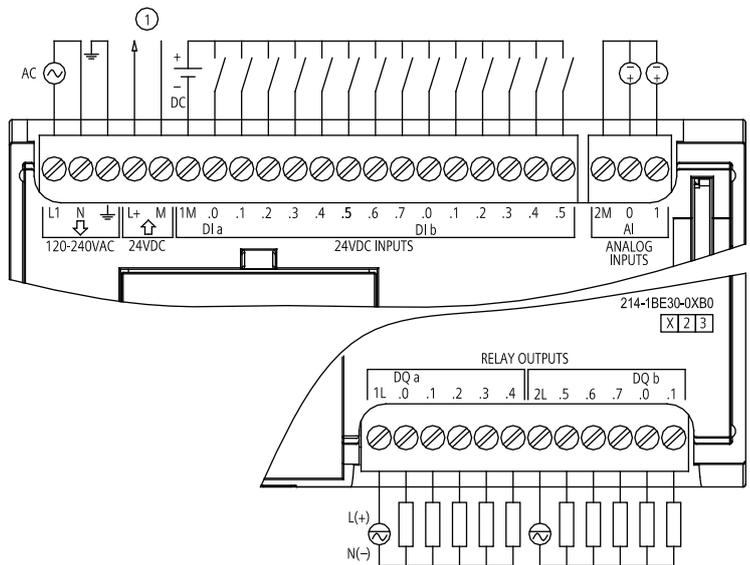
² Informationen zu den Sprungantworten der Analogeingänge enthält das Systemhandbuch der S7-1200.

³ Informationen zu den Abtastfrequenzen der Analogeingänge enthält das Systemhandbuch der S7-1200.

Beispielschaltpläne für die S7-1200-CPU

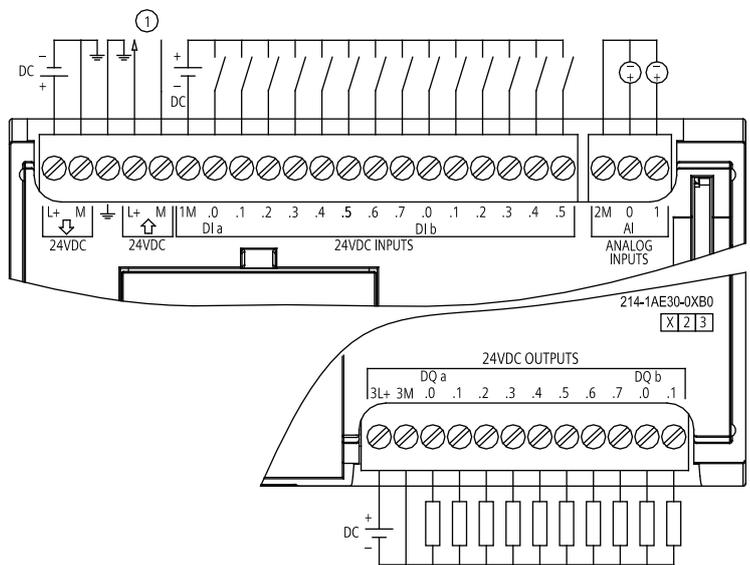
Beachten Sie für vollständige Informationen das S7-1200 Systemhandbuch.

CPU 1214C AC/DC/Relais



① 24-V-DC-Geberspannung

CPU 1214C DC/DC/DC



① 24-V-DC-Geberspannung

A.3 Signalboards

Allgemeines	SB 1223 DI 2 x 24 V DC, DO 2 x 24 V DC	SB 1223 AO 1x12 Bit
Bestellnummer	6ES7 223-0BD30-0XB0	6ES7 232-4HA30-0XB0
Abmessungen (B x H x T)	38 x 62 x 21 (mm)	38 x 62 x 21 (mm)
Gewicht	40 Gramm	40 Gramm
Leistungsverlust	1,0 W	1,5 W
Stromaufnahme (SM-Bus)	50 mA	15 mA
Stromaufnahme (24 V DC)	4 mA / Eingang	40 mA (ohne Last)
Eingänge/Ausgänge	2 Eingänge (IEC Typ 1, stromziehend) 2 Ausgänge (MOSFET, elektronisch)	1 Ausgang (Spannung oder Strom)

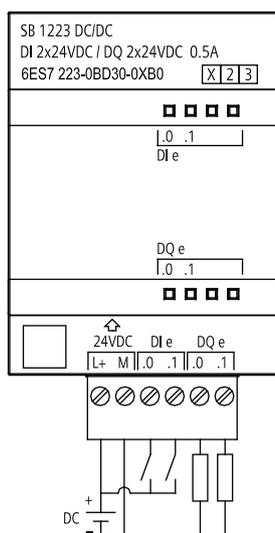
Digitaleingänge	SB 1223 DI 2 x 24 V DC, DO 2 x 24 V DC
Anzahl und Art der Eingänge (Anzahl gleichzeitig eingeschalteter Eingänge)	IEC Typ 1 stromziehend: 2 Eingänge (2)
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
HSC Eingangstaktfrequenzen (max.)	20 kHz (15 bis 30 V DC); 30 kHz (15 bis 26 V DC)
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1
Filterzeiten	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 2
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt

Digitalausgänge	SB 1223 DI 2 x 24 V DC, DO 2 x 24 V DC
Anzahl und Art der Ausgänge (Anzahl gleichzeitig eingeschalteter Ausgänge)	MOSFET, elektronisch: 2 Ausgänge (2)
Spannungsbereich	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	min. 20 V DC
Signal logisch 0 bei 10 k Ω Last	max. 0,1 V DC
Strom (max.)	0,5 A
Lampenlast	5 W
Kontaktwiderstand bei EIN	max. 0,6 Ω
Kriechstrom pro Ausgang	max. 10 μ A
Frequenz Impulsgenerator	max. 20 KHz, min. 2 Hz
Einschaltstrom	5 A für max. 100 ms
Überlastschutz	Nein
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	1

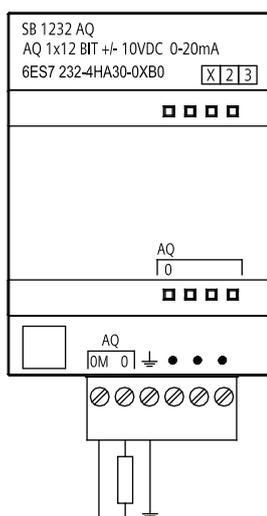
Digitalausgänge	SB 1223 DI 2 x 24 V DC, DO 2 x 24 V DC
Ströme je Leiter	1 A
Induktive Klemmspannung	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 2 µs von Aus nach Ein; max. 10 µs von Ein nach Aus
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Leitungslänge (Meter)	500 geschirmt, 150 ungeschirmt

Analogausgänge	SB 1223 AO 1x12 Bit	
Anzahl und Art der Ausgänge	1 (Spannung oder Strom)	
Bereich	±10 V oder 0 bis 20 mA	
Auflösung	Spannung: 12 Bit	Strom: 11 Bit
Vollausschlag (Datenwort)	Spannung: -27,648 bis 27,648	Strom: 0 bis 27,648
Genauigkeit (25 °C / 0 bis 55 °C)	±0,5 % / ±1 % des Vollausschlags	
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µS (R), 750 µS (1 uF)	Strom: 600 µS (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: ≥ 1000 Ω	Strom: ≤ 600 Ω
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Elektrische Trennung (Feld zu Logik)	Keine	
Leitungslänge (Meter)	100 Meter, verdreht und geschirmt	
Diagnose	Überlauf/Unterlauf	Ja
	Erdschluss (nur Spannungsmodus)	Ja
	Drahtbruch (nur Strommodus)	Ja

SB 1223 2 x 24 V DC 2 x 24 V DC



SB 1232 AO 1



A.4 Digitale Signalmodule

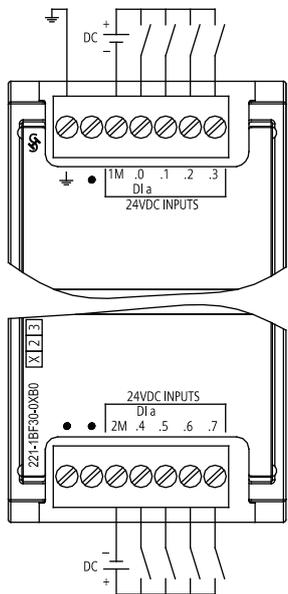
Die folgenden technischen Dateien zeigen beispielhaft eines der für die S7-1200 erhältlichen digitalen SM-Module. Beachten Sie für weitere Informationen das S7-1200 Systemhandbuch.

Beispiel für Signalmodule SM 1221 (nur Eingang)

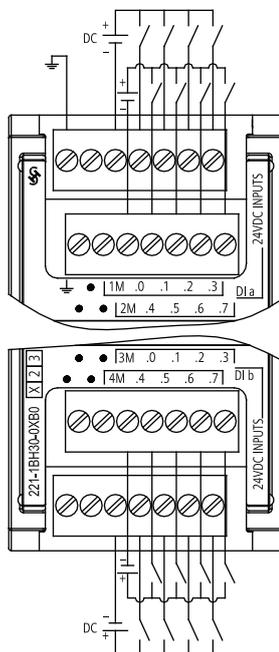
Allgemeines	SM 1221 DI 8x24VDC	SM 1221 DI 16x24VDC
Anzahl Eingänge (Anzahl gleichzeitig eingeschalteter Eingänge)	8 (8)	16 (16)
Abmessungen (B x H x T)	45 x 100 x 75 (mm)	45 x 100 x 75 (mm)
Gewicht	170 Gramm	210 Gramm
Leistungsverlust	1,5 W	2,5 W
Stromaufnahme (SM-Bus)	105 mA	130 mA
Stromaufnahme (24 V DC)	4 mA/Eingang	4 mA/Eingang

Digitaleingänge	Beschreibung
Eingangstyp	Stromziehend/stromliefernd (IEC Typ 1, wenn stromziehend)
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.)	15 V DC bei 2,5 mA
Signal logisch 0 (max.)	5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik)	500 V AC für 1 Minute
Potentialgetrennte Gruppen	DI 8 x 24 VDC: 2; DI 16 x 24 VDC: 4
Filterzeiten (ms)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms (wählbar in Gruppen zu je 4)
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt

SM 1221 DI 8 x 24 VDC



SM 1221 DI 16 x 24 VDC



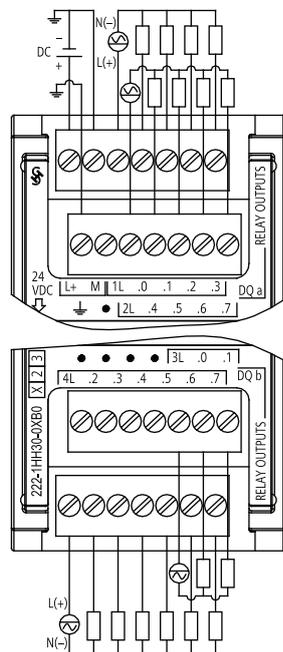
Beispiel für Signalmodule SM 1222 (nur Ausgang)

Allgemeines	SM1222 DO 16xRelais	SM1222 DO 16x24 VDC
Anzahl und Art der Ausgänge	16 Relais, Trockenkontakt	16 MOSFET, elektronisch
Abmessungen (B x H x T)	45 x 100 x 75 (mm)	45 x 100 x 75 (mm)
Gewicht	260 Gramm	220 Gramm
Leistungsverlust	8,5 W	2,5 W
Stromaufnahme (SM-Bus)	135 mA	140 mA
Stromaufnahme (24 V DC)	11 mA / Relaisspule	-/-

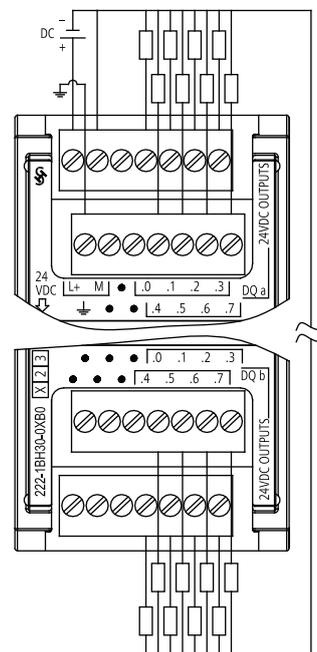
Digitalausgänge	SM1222 DO 16xRelais	SM1222 DO 16x24 VDC
Anzahl und Art der Ausgänge (Anzahl gleichzeitig eingeschalteter Ausgänge)	16 Relais, Trockenkontakt (16)	16 MOSFET, elektronisch (16)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom	-/-	min. 20 V DC
Signal logisch 0 bei 10 kΩ Last	-/-	max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	-/-	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms

Digitalausgänge	SM1222 DO 16xRelais	SM1222 DO 16x24 VDC
Überlastschutz	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
Isolationswiderstand Elektrische Trennung zwischen offenen Kontakten	min. 100 MΩ, wenn neu 750 V AC für 1 Minute	-/-
Potentialgetrennte Gruppen	4	1
Strom je Leiter (max.)	10 A	8 A
Induktive Klemmspannung	-/-	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 10 ms	max. 50 µs von Aus nach Ein max. 200 µs von Ein nach Aus
Mechanische Lebensdauer (ohne Last) Lebensdauer der Kontakte bei Nennlast	10.000.000 Schaltspiele auf/zu 100.000 Schaltspiele auf/zu	-/-
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Leitungslänge (Meter)	500 geschirmt, 150 ungeschirmt	500 geschirmt, 150 ungeschirmt

SM 1222 DO 16 x Relais



SM 1222 DO 16 x 24 V DC



Beispiele für Kombinationssignalmodule SM 1223 mit digitalen Eingängen/Ausgängen

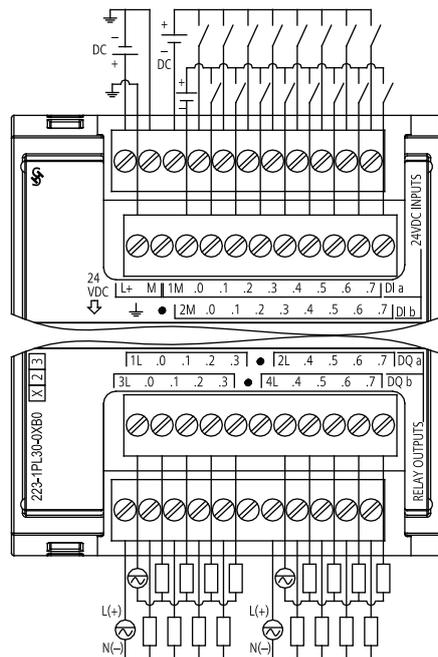
Allgemeines	SM 1223 DI 16x24 VDC, DO 16xRelais	SM 1223 DI 16x24 VDC, DO16x24 VDC
Anzahl und Art der Eingänge (Anzahl gleichzeitig eingeschalteter Eingänge)	16 stromziehend/stromliefernd (IEC Typ 1 stromziehend) (16)	16 stromziehend/stromliefernd (IEC Typ 1 stromziehend) (16)
Anzahl und Art der Ausgänge (Anzahl gleichzeitig eingeschalteter Ausgänge)	16 Relais, Trockenkontakt (16)	16 MOSFET, elektronisch (16)
Abmessungen (B x H x T)	70 x 100 x 75 (mm)	70 x 100 x 75 (mm)
Gewicht	350 Gramm	310 Gramm
Leistungsverlust	10 W	4,5 W
Stromaufnahme (SM-Bus)	180 mA	185 mA
Stromaufnahme (24 V DC)	4 mA / Eingang 11 mA / Relaispule	4 mA / Eingang

Digitaleingänge	Beschreibung
Anzahl und Art der Eingänge	16 stromziehend/stromliefernd (IEC Typ 1 stromziehend)
Nennspannung	24 V DC bei 4 mA, Nennwert
Zulässige Dauerspannung	max. 30 V DC
Stoßspannung	35 V DC für 0,5 s
Signal logisch 1 (min.) Signal logisch 0 (max.)	15 V DC bei 2,5 mA 5 V DC bei 1 mA
Elektrische Trennung (Feld zu Logik) Potentialgetrennte Gruppen	500 V AC für 1 Minute 2
Filterzeiten (ms)	0,2, 0,4, 0,8, 1,6, 3,2, 6,4 und 12,8 ms wählbar in Gruppen zu je 4
Leitungslänge (Meter)	500 geschirmt, 300 ungeschirmt

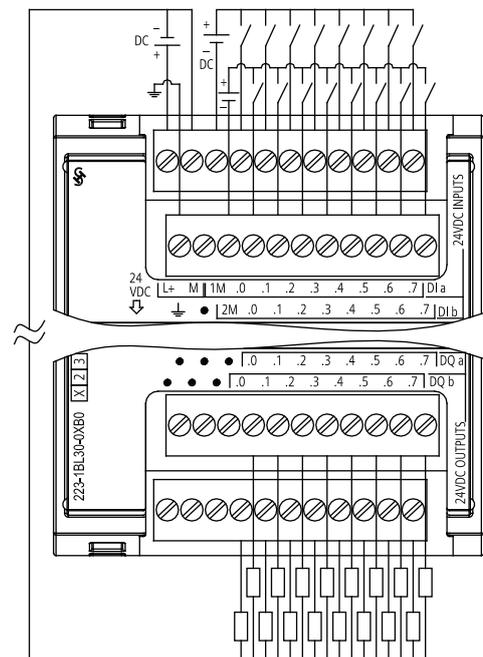
Digitalausgänge	SM 1223 DI 16x24 VDC, DO 16xRelais	SM 1223 DI 16x24 VDC, DO16x24 V DC
Anzahl und Art der Ausgänge (Anzahl gleichzeitig eingeschalteter Ausgänge)	16 Relais, Trockenkontakt (16)	16 MOSFET, elektronisch (16)
Spannungsbereich	5 bis 30 V DC oder 5 bis 250 V AC	20,4 bis 28,8 V DC
Signal logisch 1 bei max. Strom Signal logisch 0 bei 10 kΩ Last	-/-	min. 20 V DC max. 0,1 V DC
Strom (max.)	2,0 A	0,5 A
Lampenlast	30 W DC/200 W AC	5 W
Kontaktwiderstand bei EIN	max. 0,2 Ω wenn neu	max. 0,6 Ω
Kriechstrom pro Ausgang	-/-	max. 10 µA
Einschaltstrom	7 A bei geschlossenen Kontakten	8 A für max. 100 ms
Überlastschutz	Nein	Nein

Digitalausgänge	SM 1223 DI 16x24 VDC, DO 16xRelais	SM 1223 DI 16x24 VDC, DO16x24 V DC
Elektrische Trennung (Feld zu Logik)	Spule zu Kontakt: 1500 V AC für 1 Minute Spule zu Logik: Keine	500 V AC für 1 Minute
Isolationswiderstand Elektrische Trennung zwischen offenen Kontakten	min. 100 MΩ, wenn neu 750 V AC für 1 Minute	-/-
Potentialgetrennte Gruppen	4	1
Strom je Leiter (max.)	8 A	8 A
Induktive Klemmspannung	-/-	L+ minus 48 V, 1 W Verlustleistung
Schaltverzögerung	max. 10 ms	max. 50 μs von Aus nach Ein max. 200 μs von Ein nach Aus
Mechanische Lebensdauer (ohne Last) Lebensdauer der Kontakte bei Nennlast	10.000.000 Schaltspiele auf/zu 100.000 Schaltspiele auf/zu	-/-
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)	
Leitungslänge (Meter)	500 geschirmt, 150 ungeschirmt	

SM1223 DI 16 x 24 VDC, DO 16 x Relais



SM 1223 DI 16 x 24 VDC, DO 16 x 24 V DC



A.5 Analoge Signalmodule

Die folgenden technischen Dateien zeigen beispielhaft eines der für die S7-1200 erhältlichen analogen SM-Module. Beachten Sie für weitere Informationen das S7-1200 Systemhandbuch.

Allgemeines	SM 1231 AI 4x13 Bit	SM 1234 AI 4x13 Bit AO 2x14 Bit	SM 1232 AO 2x14 Bit
Anzahl und Art der Eingänge (wählbar in Gruppen zu je 2)	4 Spannung oder Strom (differential)	4 Spannung oder Strom (differential)	0
Anzahl und Art der Ausgänge	0	2 Spannung oder Strom	2 Spannung oder Strom
Abmessungen B x H x T	45 x 100 x 75 (mm)	45 x 100 x 75 (mm)	45 x 100 x 75 (mm)
Gewicht	180 Gramm	220 Gramm	180 Gramm
Leistungsverlust	1,5 W	2,0 W	1,5 W
Stromaufnahme (SM-Bus)	80 mA	80 mA	80 mA
Stromaufnahme (24 V DC)	45 mA	60 mA (ohne Last)	45 mA (ohne Last)

Analogeingänge	Beschreibung
Art der Eingänge	Spannung oder Strom (differential), wählbar in Gruppen zu je 2
Bereich	±10 V, ±5 V, ±2,5 V oder 0 bis 20 mA
Vollausschlag (Datenwort)	-27.648 bis 27.648
Überschwing-/Unterschwingbereich (Datenwort) ¹	Spannung: 32.511 bis 27.649 / -27.649 bis -32.512 Strom: 32.511 bis 27.649 / 0 bis -4864
Überlauf/Unterlauf (Datenwort) ¹	Spannung: 32.767 bis 32.512 / -32.513 bis -32.768 Strom: 32.767 bis 32.512 / -4865 bis -32.768
Auflösung	12 Bit + Vorzeichenbit
Maximale Stehspannung/-strom	±35 V / ±40 mA
Glättung ²	Keine, schwach, mittel oder stark
Rauschunterdrückung ³	400, 60, 50 oder 10 Hz
Impedanz	≥ 9 MΩ (Spannung) / 250 Ω (Strom)
Elektrische Trennung (Feld zu Logik)	Keine
Genauigkeit (25 °C / 0 bis 55 °C)	±0,1 % / ±0,2 % des Vollausschlags
Analog-Digital-Umsetzzeit	625 µs (400 Hz Unterdrückung)
Gleichtaktunterdrückung	40 dB, Nennwert bei 60 Hz
Betriebssignalebene	Signal- plus Gleichtaktspannung muss kleiner als +12 V und größer als -12 V sein
Leitungslänge (Meter)	100 Meter, verdreht und geschirmt

¹ Darstellungen der Spannungs- und Stromdaten für die Analogeingänge enthält das Systemhandbuch der S7-1200.

² Informationen zu den Sprungantworten der Analogeingänge enthält das Systemhandbuch der S7-1200.

³ Informationen zu den Abtastfrequenzen der Analogeingänge enthält das Systemhandbuch der S7-1200.

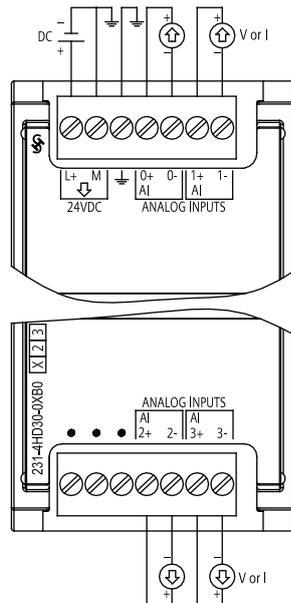
Analogausgänge	Beschreibung
Art der Ausgänge	Spannung oder Strom
Bereich	±10 V oder 0 bis 20 mA
Auflösung	Spannung: 14 Bit Strom: 13 Bit
Vollausschlag (Datenwort) ¹	Spannung: -27.648 bis 27.648 Strom: 0 bis 27.648
Genauigkeit (25 °C / 0 bis 55 °C)	±0,3 % / ±0,6 % des Vollausschlags
Ausregelzeit (95 % des neuen Werts)	Spannung: 300 µS (R), 750 µS (1 uF) Strom: 600 µS (1 mH), 2 ms (10 mH)
Lastimpedanz	Spannung: ≥ 1000 Ω Strom: ≤ 600 Ω
Verhalten bei Wechsel von RUN nach STOP	Letzter Wert oder Ersatzwert (Voreinstellung 0)
Elektrische Trennung (Feld zu Logik)	Keine
Leitungslänge (Meter)	100 Meter verdreht und geschirmt

¹ Darstellungen der Spannungs- und Stromdaten für die Analogausgänge enthält das Systemhandbuch der S7-1200.

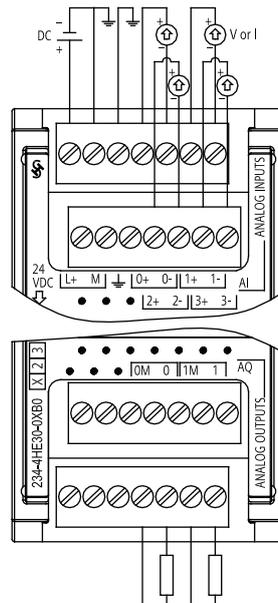
Diagnose	SM 1231 AI 4x13 Bit	SM 1234 AI 4x13 Bit AO 2x14 Bit	SM 1232 AO 2x14 Bit
Überlauf/Unterlauf	Ja ¹	Ja ¹	-/-
Erdschluss (nur Spannungsmodus)	Nein	Ja (Ausgänge)	Ja
Drahtbruch (nur Strommodus)	Nein	Ja (Ausgänge)	Ja
24 V DC Niederspannung	Ja	Ja	Ja

¹ 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.

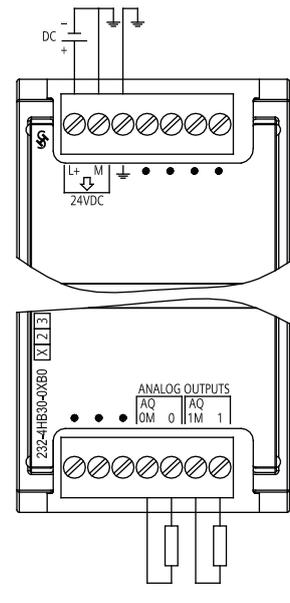
SM 1231 AI 4 x 13 Bit



SM 1234 AI 4x13bit AQ 2x14bit



SM 1232 AQ 2 x 14 bit

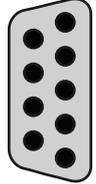


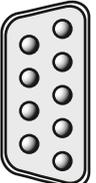
A.6 Kommunikationsmodule

Die folgenden technischen Dateien zeigen beispielhaft eines der für die S7-1200 erhältlichen CM-Module. Beachten Sie für Pinbelegungen und weitere Informationen das S7-1200 Systemhandbuch.

Allgemeines	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

Sende- und Empfangseinheit		Beschreibung
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 Signal zu CPU-Logik	500 V AC für 1 Minute
Kabellänge, geschirmt (max.)		<ul style="list-style-type: none"> RS485: 1000 m RS232: 10 m

CM 1241 RS485				
Pol	Beschreibung	Steckverbinder (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

CM 1241 RS232				
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

Index

"

"Box"-Operation
Erste Schritte, 27

A

Adressen im Speicher, 38
Alarme
Organisationsbaustein (OB), 56
Allgemeine technische Daten, 101

Ä

Ändern von Einstellungen bei STEP 7 Basic, 20

A

Anlaufparameter, 50
Anlegen einer HMI-Verbindung, 29
Anschlüsse
HMI-Verbindung, 29
Netzwerkverbindung, 29
Anwenderprogramm
Drag & Drop zwischen Editoren, 18
Einfügen von Anweisungen, 17
Favoriten, 17
Anzeigen von Inhalt und Index (Online-Hilfe), 14
Arbeitsspeicher, 35
ATEX-Zulassung, 101, 102
Aufrufstruktur, 74
Einleitung, 74
Ausgangsparameter, 57

B

Baustein
Arten, 40
Aufrufen eines anderen Codebausteins, 59
Erste Schritte, 59
Konsistenzprüfung, 75
Bausteinaufruf
Grundwissen, 40
Bausteine
Datenbausteine (DBs), 40

Funktionen (FCs), 40
Funktionsbausteine (FBs), 40
Organisationsbausteine (OBs), 40
Bearbeitung von Interruptereignissen
Organisationsbaustein (OB), 56
Bedienoberfläche
Portalansicht, 13
Projektansicht, 14
Bedienpanel, 18, 35, 94
Bemessungsspannungen, 105
Beobachtungstabellen, 72, 98
Betriebszustand, 18, 35, 94
Betriebszustand RUN, 34
Bedienpanel, 35, 94
Betriebszustand STOP, 34
Bedienpanel, 35, 94
Bibliothek für das USS-Protokoll, 82
Bitverknüpfung, 61

C

CE-Zulassung, 101
CM 1241 RS232, Technische Daten, 120
CM 1241 RS485, Technische Daten, 120
Codebaustein
Baustein aufrufen, 59
DB (Datenbaustein), 58
FB (Funktionsbaustein), 57
FC (Funktion), 57
Kopieren von einer Online-CPU, 95
Codebausteine, 54
CPU
1211C, technische Daten, 106
1212C, Technische Daten, 106
1214C, Technische Daten, 106
Anlauf, Aufgaben beim, 56
Anlaufparameter, 50
Anlaufverarbeitung, 50
Baustein aufrufen, 59
Bedienpanel, 18, 35, 94
Beobachtungstabellen, 98
Betriebszustände, 34
Diagnosepuffer, 98
Erste Schritte, 21
Ethernet-Port, 52
Forcen, 99
Gerätekonfiguration, 45
IP-Adresse, 52

- Kommunikation mit HMI konfigurieren, 77
- Konfigurieren von Parametern, 50, 51
- Module hinzufügen, 48
- Netzwerkverbindung, 49
- Neues Gerät hinzufügen, 47
- Nicht spezifizierte CPU, 46, 96
- Online, 93
- Organisationsbaustein (OB), 56
- Passwortschutz, 43
- PROFINET, 52
- Programmausführung, 33
- Schutzstufen, 43
- Synchronisieren, 95
- Thermischer Bereich, 11
- Überblick, 7
- Upload, 95
- Vergleichstabelle, 8
- CSA-Zulassung, 101
- C-Tick-Zulassung, 102
- cULus-Zulassung, 101

D

- Datenbaustein
 - Globaler Datenbaustein, 38, 58
 - Instanzen-Datenbaustein, 38
- Datenbaustein (DB), 58
- Datenhantierungsbausteine (DHBs), 59
- Datentyp DTL, 37
- Datentyp DTL (Date and Time Long), 37
- Datentypen, 36
 - DTL, 37
- DB (Datenbaustein), 58
- Diagnosepuffer, 98
- Direkthilfe, 14
- Dokumentation, 14
- Drag & Drop zwischen Editoren, 18
- Drucken von Hilfethemen, 16

E

- E/A
 - Adressierung, 40
- E/A-Module
 - Beobachtungstabellen, 98
- Editoren teilen
 - Erste Schritte, 22, 25
- Einbau
 - Einbaumaße, 11
 - Thermischer Bereich, 11
- Einfügen eines Geräts

- Nicht spezifizierte CPU, 46, 96
- Einfügen von Anweisungen
 - Drag & Drop zwischen Editoren, 18
 - Favoriten, 17
- Elektromagnetische Verträglichkeit (EMV), 103
- Entwerfen einer Automatisierungslösung mit einem PLC-Gerät, 40, 53
- Ereignisausführung, 41
- Ereignisse, 98
 - Organisationsbaustein (OB), 56
- Erkennen, 46, 96
- Erreichbare Teilnehmer, 93
- Erste Schritte
 - "Box"-Operation, 27
 - Adressierung, 25
 - Arithmetische Operation, 27
 - Baustein, 59
 - CPU, 21
 - Direkthilfe, 14
 - Dokumentation, 14
 - Editoren teilen, 22, 25
 - Hilfe beim Eingeben, 14
 - HMI, 28, 30
 - HMI-Verbindung, 29
 - Informationssystem, 14
 - Kaskadierende Tooltips, 14
 - Kontakte, 24
 - KOP-Programm, 24, 27
 - Netzwerk, 24
 - Netzwerkverbindung, 29
 - neue SPS, 21
 - Online-Hilfe, 14
 - Operationen, 25
 - PLC-Variablen, 22, 25
 - Programmbaustein, 59
 - Projekt, 21
 - Tooltips, 14
 - Variablen, 22, 25
- Erweiterung der Leistungsmerkmale der S7-1200, 9
- Ethernet
 - IP-Adresse, 52
 - Netzwerkverbindung, 49
- Ethernet-Kommunikation, 77
- Ethernet-Operationen
 - TRCV_C, 79
 - TSEND_C, 78

F

- Favoriten, Funktionsleiste, 17
- FB (Funktionsbaustein), 57
- FC (Funktion), 57

Fenster der Online-Hilfe erweitern, 14
 FM-Zertifizierung, 102
 Forcen, 99
 Funktion (FC), 57
 Funktionsbaustein (FB)
 Anfangswert, 57
 Ausgangsparameter, 57
 Instanz-Datenbaustein, 57
 FUP (Funktionsplan), 60

G

Gerätekonfiguration, 45
 CPU konfigurieren, 50
 Erkennen, 46, 96
 Ethernet-Port, 52
 Konfigurieren der CPU, 51
 Konfigurieren der Module, 50, 51
 Module hinzufügen, 48
 Netzwerkverbindung, 49
 Neues Gerät hinzufügen, 47
 PROFINET, 52
 Globale Bibliothek
 USS, 82
 Globaler Datenbaustein, 38, 58

H

Hardwarekonfiguration, 45
 Erkennen, 46, 96
 Ethernet-Port, 52
 Konfigurieren der CPU, 50, 51
 Konfigurieren der Module, 50, 51
 Module hinzufügen, 48
 Netzwerkverbindung, 49
 Neues Gerät hinzufügen, 47
 PROFINET, 52
 Hilfe, 14
 Abkoppeln, 14
 Anzeigen von Inhalt und Index, 14
 Drucken, 16
 Erweitern, 14
 Hilfe beim Eingeben, 14
 Hinzufügen eines Geräts
 Nicht spezifizierte CPU, 46, 96
 HMI
 Bild, 30
 Erste Schritte, 28, 30
 HMI-Verbindung, 29
 Netzwerkverbindung, 29
 PROFINET-Kommunikation konfigurieren, 77

HMI-Geräte
 Netzwerkverbindung, 49
 Überblick, 10
 HMI-Verbindung, 29
 HSC (schneller Zähler), 86
 Konfiguration, 88

I

Impulsfolge (PTO), 90
 Informationssystem, 14
 Abkoppeln, 14
 Anzeigen von Inhalt und Index, 14
 Drucken, 16
 Erweitern, 14
 Instanz-Datenbaustein, 38
 IP-Adresse, 52
 IP-Adresse des Routers, 52
 IP-Router, 52

K

Kommunikation
 Bibliotheken, 80
 IP-Adresse, 52
 Netzwerk, 77
 Kommunikationsmodul
 Gerätekonfiguration, 45
 Module hinzufügen, 48
 Neues Gerät hinzufügen, 47
 Kommunikationsmodul (CM)
 Technische Daten, 120
 Vergleichstabelle, 9
 Kommunikationsmodul (CM), USS-Bibliothek, 82
 Konfiguration
 Anlaufparameter, 50
 Erkennen, 46, 96
 HSC (schneller Zähler), 88
 Industrial Ethernet-Port, 52
 IP-Adresse, 52
 PROFINET, 52
 Konfigurieren von Parametern
 CPU, 50, 51
 Module, 51
 Konsistenzprüfung, 75
 Einleitung, 75
 Kontakte
 Erste Schritte, 24
 KOP (Kontaktplan), 60
 Kopieren von Codebausteinen von der Online-CPU, 95

- L**
- Ladespeicher, 35
 - Lebensdauer eines Relais, 105
 - Lineare Programmierung, 54
- M**
- MAC-Adresse, 52
 - Module
 - Konfigurieren von Parametern, 51
 - Parameter konfigurieren, 50
 - Thermischer Bereich, 11
 - Vergleichstabelle, 9
 - Montage
 - Abmessungen, 11
 - Thermischer Bereich, 11
 - MRES
 - Bedienpanel, 18, 35, 94
- N**
- Netzwerk
 - Erste Schritte, 24, 27
 - Netzwerkverbindung, 29
 - Netzwerkkommunikation, 77
 - Netzwerkverbindung, 49
 - Netzwerkverbindung erstellen, 49
 - Netzwerkverbindung erstellen, 29
 - Neues Projekt
 - Erste Schritte, 21
 - Hinzufügen eines HMI-Geräts, 28
 - HMI-Bild, 30
 - HMI-Verbindung, 29
 - Netzwerkverbindung, 29
 - Nicht spezifizierte CPU, 46, 96
- O**
- Online
 - Bedienpanel, 18, 35, 94
 - Codebausteine, 95
 - Erkennen, 96
 - erreichbare Teilnehmer, 93
 - Forcen, 99
 - Synchronisieren, 95
 - Überwachung der Speicherauslastung, 95
 - Verbinden mit einer CPU, 93
 - Zykluszeitüberwachung, 94
 - Online-Hilfe, 14
 - Abkoppeln, 14
 - Anzeigen von Inhalt und Index, 14
 - Drucken, 16
 - Hilfenfenster erweitern, 14
 - Online-Hilfe abkoppeln, 14
 - Online-Hilfe aufrufen, 14
 - Operation Ausschaltverzögerung (TOF), 66
 - Operation Bereich ununterbrechbar kopieren (UMOVE_BLK), 64
 - Operation CTRL_PWM, 90
 - Operation Einschaltverzögerung (TON), 66
 - Operation Impuls (TP), 66
 - Operation PID_Compact, 68
 - Operation RT (Zeit zurücksetzen), 66
 - Operation Schnellen Zähler aktivieren (HSC), 88
 - Operation TRCV_C, 79
 - Operation TSEND_C, 78
 - Operation Verschieben, 64
 - Operation Zeit akkumulieren (TONR), 66
 - Operation Zeit zurücksetzen (RT), 66
 - Operationen
 - Bereich kopieren (MOVE_BLK), 64
 - Bereich ununterbrechbar kopieren (UMOVE_BLK), 64
 - Bitverknüpfung, 61
 - CTRL_PWM, 90
 - Drag & Drop, 17
 - Drag & Drop zwischen Editoren, 18
 - Einfügen, 17
 - Erste Schritte, 25, 27
 - Favoriten, 17
 - Hinzufügen eines Parameters, 27
 - PID_Compact, 68
 - Schneller Zähler (HSC), 88
 - TRCV_C, 79
 - TSEND_C, 78
 - Vergleichen, 63
 - Verschieben, 64
 - Zähler, 64
 - Zeit, 66
 - Zeit: RT (Zeit zurücksetzen), 66
 - Zeit: TOF (Ausschaltverzögerung), 66
 - Zeit: TON (Einschaltverzögerung), 66
 - Zeit: TONR (Zeit akkumulieren), 66
 - Zeit: TP (Impuls), 66
 - Operationen Bereich kopieren (MOVE_BLK), 64
 - Operationen mit Drag & Drop einfügen, 17
 - Organisationsbaustein
 - Anlegen, 56
 - Bearbeiten, 56
 - Funktionsweise konfigurieren, 57
 - mehrere Programmzyklus-OBs, 56

P

- Parameter konfigurieren
 - Ethernet-Port, 52
 - Module, 50
 - PROFINET, 52
- Parametrieren, 57
- Passwortschutz
 - CPU, 43
- PLC
 - Baustein aufrufen, 59
 - Bausteine verwenden, 40, 53
 - Erste Schritte, 21
 - Forcen, 99
 - Online, 93
 - Operationen, 25
 - Synchronisieren, 95
 - Überblick, 7
 - Upload, 95
 - Variablen, 22, 25
- PLC-Variablen
 - Erste Schritte, 22, 25
- Portalansicht, 13
 - Ethernet-Port konfigurieren, 52
 - Konfigurieren der CPU, 50, 51
 - Konfigurieren der Module, 50, 51
 - Module hinzufügen, 48
 - Neues Gerät hinzufügen, 47
 - PROFINET, 52
- Prioritäten bei der Bearbeitung, 41
- PROFINET, 77
 - IP-Adresse, 52
 - Netzwerkverbindung, 49
 - Testen eines Netzwerks, 53
- PROFINET-Schnittstelle
 - Eigenschaften der Ethernet-Adresse, 52
- Programm
 - Arithmetische Operation, 27
 - Baustein aufrufen, 59
 - Beispielnetzwerk, 24, 27
 - Erste Schritte, 24, 27
 - Kopieren von einer Online-CPU, 95
 - Synchronisieren, 95
 - Upload, 95
- Programm beobachten, 72
- Programm testen, 72
- Programmausführung, 33, 40
- Programmbaustein
 - Erste Schritte, 21, 59
- Programmierung
 - Drag & Drop zwischen Editoren, 18
 - Einfügen von Operationen, 17
 - Erste Schritte, 25
 - Favoriten, 17
 - FUP (Funktionsplan), 60
 - KOP (Kontaktplan), 60
 - Linear, 54
 - Nicht spezifizierte CPU, 46, 96
 - Strukturiert, 54
- Programminformationen
 - In der Aufrufstruktur, 74
- Programmkarte, 35
- Programmstruktur, 54
- Projekt
 - Erste Schritte, 21
 - Hinzufügen eines HMI-Geräts, 28
 - HMI-Bild, 30
 - HMI-Verbindung, 29
 - Netzwerkverbindung, 29
 - Programm, 25
 - Variablen, 22, 25
 - Zugang zur CPU einschränken, 43
- Projektansicht, 13, 14
 - Ethernet-Port konfigurieren, 52
 - Gerätekonfiguration, 45
 - Konfigurieren der CPU-Parameter, 50, 51
 - Konfigurieren der Module, 50, 51
 - Module hinzufügen, 48
 - Netzwerkverbindung, 49
 - Neues Gerät hinzufügen, 47
 - PROFINET, 52
- Protokoll
 - Kommunikation, 79
- PTO (Impulsfolge), 90
- PtP-Kommunikation, 79
- Punkt-zu-Punkt-Kommunikation, 79
- PWM
 - Operation CTRL_PWM, 90

Q

- Querverweise
 - Einleitung, 74
 - Verwendung, 74

R

- Remanenter Speicher, 35
- RUN, Betriebszustand
 - Bedienpanel, 18
 - Forcen, 99
 - Programmausführung, 33

- S**
- S7-1200
 - Anlaufparameter, 50
 - Baustein aufrufen, 59
 - Bedienpanel, 18, 35, 94
 - CPU, 7
 - CPU-Parameter konfigurieren, 50
 - Diagnosepuffer, 98
 - Einbaumaße, 11
 - Erweiterung der Leistungsmerkmale, 9
 - Ethernet-Port, 52
 - Forcen, 99
 - Gerätekonfiguration, 45
 - HMI-Geräte, 10
 - IP-Adresse, 52
 - Konfigurieren der CPU-Parameter, 51
 - Konfigurieren der Module, 51
 - Module hinzufügen, 48
 - Module konfigurieren, 50
 - Netzwerkverbindung, 49
 - Neues Gerät hinzufügen, 47
 - Online, 93
 - Organisationsbaustein (OB), 56
 - Passwortschutz, 43
 - PROFINET, 52
 - Programmausführung, 33
 - Synchronisieren, 95
 - Thermischer Bereich, 11
 - Upload, 95
 - Vergleichstabelle CPU-Varianten, 8
 - SB 1223, Technische Daten, 111
 - Schaltpläne
 - Analoge Signalmodule, 119
 - SM 1221 Signalmodul, 114
 - SM 1222 Signalmodul, 115
 - SM 1223 Signalmodul, 117
 - Schneller Zähler, 86
 - Schutzart, 104
 - Schutzstufe
 - CPU, 43
 - Serielle Kommunikation, 79
 - Sicherheit
 - CPU, 43
 - Signalboard (SB)
 - Gerätekonfiguration, 45
 - Module hinzufügen, 48
 - Vergleichstabelle, 9
 - Signalboard (SM)
 - Neues Gerät hinzufügen, 47
 - Signalmodul (SM)
 - Gerätekonfiguration, 45
 - Module hinzufügen, 48
 - Neues Gerät hinzufügen, 47
 - Vergleichstabelle, 9
 - Signalmodule
 - Technische Daten SM 1221, 113
 - Technische Daten SM 1222, 114
 - Technische Daten SM 1223, 116
 - Speicher
 - Arbeitsspeicher, 35
 - Ladespeicher, 35
 - Remanenter Speicher, 35
 - Systemmerker, 70
 - Taktmerker, 70
 - Temporärer Speicher (L), 38
 - Speicherkarte
 - Ladespeicher, 35
 - STARTUP, Betriebszustand
 - Forcen, 99
 - Programmausführung, 33
 - STEP 7
 - Ethernet-Port, 52
 - Gerätekonfiguration, 45
 - Konfigurieren der CPU, 50, 51
 - Konfigurieren der Module, 50, 51
 - Module hinzufügen, 48
 - Netzwerkverbindung, 49
 - Neues Gerät hinzufügen, 47
 - Portalansicht, 13
 - PROFINET, 52
 - Projektansicht, 13
 - STEP 7 Basic
 - Ändern von Einstellungen, 20
 - Bedienpanel, 18, 35, 94
 - Diagnosepuffer, 98
 - Drag & Drop zwischen Editoren, 18
 - Einfügen von Operationen, 17
 - Favoriten, 17
 - Forcen, 99
 - STOP, Betriebszustand
 - Bedienpanel, 18
 - Forcen, 99
 - Strukturierte Programmierung, 54
 - Subnetzmaske, 52
 - Synchronisieren, 95
 - Synchronisieren von Online- und Offline-CPU, 95
- T**
- TCP/IP-Kommunikation, 77
 - Technische Daten, 101
 - Allgemeine technische, 101
 - Analoge Signalmodule, 118
 - ATEX-Zulassung, 101, 102

Bemessungsspannungen, 105
 CE-Zulassung, 101
 CPU 1212C, 106
 CPU 1214C, 106
 CSA-Zulassung, 101
 C-Tick-Zulassung, 102
 cULus-Zulassung, 101
 Digitale Signalboards (SBs), 111
 Elektromagnetische Verträglichkeit (EMV), 103
 FM-Zertifizierung, 102
 Kommunikationsmodul CM 1241 RS232, 120
 Kommunikationsmodul CM 1241 RS485, 120
 Lebensdauer eines Relais, 105
 SB 1223, 111
 Schaltpläne analoge Signalmodule (SM), 119
 Schutz, 104
 SM 1221 Signalmodul, 113
 SM 1221, Schaltplan, 114
 SM 1222 Signalmodul, 114
 SM 1222, Schaltplan, 115
 SM 1223 Signalmodul, 116
 SM 1223, Schaltplan, 117
 UL-Zulassung, 101
 Umgebungen, 102
 Umgebungsbedingungen, 104
 Zulassung für das Seewesen, 102
 Technische Daten für analoge Signalmodule, 118
 Technische Daten für das digitale Signalboard (SB), 111
 Temporärer Speicher (L), 38
 Thermischer Bereich, 11
 TIA-Portal
 Ethernet-Port, 52
 Gerätekonfiguration, 45
 Konfigurieren der CPU, 50, 51
 Konfigurieren der Module, 51
 Module hinzufügen, 48
 Netzwerkverbindung, 49
 Neues Gerät hinzufügen, 47
 Portalansicht, 13
 PROFINET, 52
 Projektansicht, 13
 Tooltips, 14

U

Überwachung der Speicherauslastung, online, 95
 UL-Zulassung, 101
 Umgebungen
 Industrie, 102
 Umgebungsbedingungen, 104
 Upload

Codebausteine, 95
 Erkennen, 96
 Synchronisieren, 95

V

Variablen
 Erste Schritte, 22, 25
 Vergleichsoperationen, 63
 Vergleichstabelle
 CPU-Varianten, 8
 HMI-Geräte, 10
 Vergleichstabelle der Module, 9

W

Warteschlangen, 41

Z

Zähler
 Schneller Zähler (HSC), 86
 Schneller Zähler (HSC): Konfigurieren, 88
 Zähloperationen, 64
 Zeitoperation TOF (Ausschaltverzögerung), 66
 Zeitoperation TON (Einschaltverzögerung), 66
 Zeitoperation TONR (Zeit akkumulieren), 66
 Zeitoperation TP (Impuls), 66
 Zeitoperationen, 66
 Zulassung für das Seewesen, 102
 Zykluszeitüberwachung, 94

